Adversarial Perturbation Defense on Deep Neural Networks

XINGWEI ZHANG, XIAOLONG ZHENG, and WENJI MAO, School of Artificial Intelligence, University of Chinese Academy of Sciences; The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China

Deep neural networks (DNNs) have been verified to be easily attacked by well-designed adversarial perturbations. Image objects with small perturbations that are imperceptible to human eyes can induce DNN-based image class classifiers towards making erroneous predictions with high probability. Adversarial perturbations can also fool real-world machine learning systems and transfer between different architectures and datasets. Recently, defense methods against adversarial perturbations have become a hot topic and attracted much attention. A large number of works have been put forward to defend against adversarial perturbations, enhancing DNN robustness against potential attacks, or interpreting the origin of adversarial perturbations. In this article, we provide a comprehensive survey on classical and state-of-the-art defense methods by illuminating their main concepts, in-depth algorithms, and fundamental hypotheses regarding the origin of adversarial perturbations. In addition, we further discuss potential directions of this domain for future researchers.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence; Neural networks**; • **Security and privacy** → *Malware and its mitigation*;

Additional Key Words and Phrases: Adversarial perturbation defense, deep neural networks, security, origin

ACM Reference format:

Xingwei Zhang, Xiaolong Zheng, and Wenji Mao. 2021. Adversarial Perturbation Defense on Deep Neural Networks. *ACM Comput. Surv.* 54, 8, Article 159 (October 2021), 36 pages.
<https://doi.org/10.1145/3465397>

1 INTRODUCTION

Deep neural networks (DNNs) have made significant progress in numerous machine learning applications, such as computer vision, object recognition, natural language processing, information security, and artificial intelligence [1]. Although DNNs can significantly enhance the performance of machine learning models, they have been demonstrated to be vulnerable to well-designed adversarial perturbations [2]. The vulnerability of DNNs has triggered considerable concern about their applications to security-required issues, including face recognition, medical image analysis, and

Note that Xingwei Zhang and Xiaolong Zheng have made equal contributions.

This work is supported by the Ministry of Health of China under grant nos. 2017ZX10303401-002 and 2017YFC1200302, the Ministry of Science and Technology of China under grant nos. 2020AAA0108401 and 2019QY(Y)0101, and the Natural Science Foundation of China under grant nos. 71602184 and 71621002.

Authors' addresses: X. Zhang, X. Zheng (corresponding author), and W. Mao, The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China; emails: {zhangxingwei2019, xiaolong.zheng, wenji.mao}@ia.ac.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

0360-0300/2021/10-ART159 \$15.00

<https://doi.org/10.1145/3465397>

automatic driving. Adversarial perturbations can misguide targeting objects by poisoning clean inputs to change their intrinsic structures learned by DNNs for recognition or classification. Well-generated adversarial perturbations have been verified to attack strong models designed for real-world issues. For example, malware that integrates well-designed adversarial perturbations could effectively evade the detection of security entrance systems. Adversarial images could disguise specific persons to evade face recognition systems or fool person recognition systems. Moreover, by embedding adversarial perturbations into cameras, one could even attack systems requiring high security, such as automatic driving [3].

The vulnerability of DNNs was first presented in [2], in which the authors proposed a box-constrained formulation to generate attacks and solved the problem with the limited Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS). Subsequent researchers further propose a large number of convenient and powerful methods to generate adversarial perturbations. The effectiveness of these generation mechanisms is verified by fooling DNN-based models that are well trained and have high performance [4, 5]. Moreover, according to another line of studies, elaborately generated attacks could be defended by specific defense methods. For instance, adversarial examples are supposed to have different architectures with original samples, which can be detected by implementing detection mechanisms [6, 7]. DNN models could also be trained to defend perturbations by updating the objective functions that are redesigned to recognize the real class of perturbed samples [8, 9]. In addition, researchers further attempt to explore the origins of adversarial perturbations both in theory and practice. The origins of adversarial perturbations are verified to give a reasonable interpretation of the vulnerability of DNN models and shed light on the development of more powerful defense methods [10, 11]. Several recent studies have found that the robustness of DNNs against potential attacks could be enhanced by adding specific structures or constraints [12–14]; this phenomenon has gained significant attention [15, 16].

In the past few years, we have witnessed the rapid growth of works regarding adversarial perturbation defense on DNNs [2, 11, 17, 18]. Meanwhile, several survey papers have been proposed for introducing adversarial perturbations [19] and corresponding defense methods [20]. Miller et al. [21] also reviewed the latest studies on defense methods against several attacks, including test-time evasion (also known as adversarial perturbations mentioned in this article), data poisoning (DP), backdoor DP, and reverse engineering (RE) attacks. However, the content of the existing works on defense methods is limited, and few studies are related to interpreting the origin of adversarial perturbations. To the best of our knowledge, there is no existing work that has comprehensively reviewed the classic and latest studies on defense methods, their performance evaluation, and the origins of adversarial perturbations.

The remainder of this article is organized as follows. In Section 2, we briefly present the basic concept of DNN-based models, commonly used adversarial perturbations, and benchmarks for evaluating defense methods. Section 3 reviews the existing hypotheses on the origins of adversarial perturbations. In Section 4, we introduce the mechanism of classic and state-of-the-art defense methods. Section 5 discusses the limitations and future directions in this field. Section 6 concludes this article.

2 PRELIMINARY

In this section, we introduce the fundamental concepts and notations of DNN-based models and adversarial perturbations.

2.1 DNN-Based Models and Basic Notations

Driven by the rapid development of advanced hardware computing and the effectiveness of extracting valuable information from the intricacy of high-dimensional data, DNNs have

significantly outperformed traditional machine learning on complex issues. By functionally mimicking the human neocortex, the DNN is constructed with numerous fully connected neurons combined with nonlinear activations [1]. This architecture extends the space of inputs to a larger and more expressive space to extract valuable information [22]. Features are delivered in a layer-to-layer way and then make precise predictions. Since feature extraction is performed automatically without expert knowledge, DNNs are thought to be more effective than traditional feature extraction methods [23].

In this article, we briefly assign a DNN-based classifier (threat model for attack and defense evaluation) $f(x|\theta)$ with d length vector inputs (the dimension of images is flattened for simplification) to interpret the mechanism of attacks and corresponding defense methods. $f(x|\theta)$ is assumed to predict the exact class of inputs from m classes, and θ is the trainable parameter of $f(\cdot)$. The performance of $f(x|\theta)$ is normally evaluated by its accuracy on a standard dataset and its robustness against potential attacks (i.e., manipulated samples) [23, 24]. Generally, DNN classifiers should be designed to satisfy both accuracy and robustness, while practical experiments have demonstrated that it is impossible to meet both demands at the same time, and subsequent inferences prove that robustness may sometimes even be odd with accuracy [25]. Specifically, considering a simple binary classification model $f(x) = \text{sign}(x)$, $y \in \{-1, 1\}$, the accuracy of the model could be calculated through the expected operation on normal inputs (with no perturbation) and ground-truth classes $\text{acc} = \frac{1}{|\mathcal{D}|} |\mathbb{E}_{(x,y) \sim \mathcal{D}} [y \cdot f(x)]|$, where $\mathcal{D}$ is the standard dataset. Similarly, the robustness criterion can be formulated as $\text{acc}_{\mathcal{R}} = \frac{1}{|\mathcal{D}_{\mathcal{R}}|} |\mathbb{E}_{(x',y) \sim \mathcal{D}_{\mathcal{R}}} [y \cdot f(x')]|$, where $\mathcal{D}_{\mathcal{R}}$ is the dataset for robustness verification, which contains perturbed samples x' and ground-truth classes y . Adversarial perturbations η are assumed to have the same dimension as normal inputs, and adversarial examples are generated by directly adding perturbations on clean samples $x' = x + \eta$. The classifier is trained to obtain optimal parameter θ^* that could minimize the expected loss between predictions and ground-truth classes $\theta^* = \min_{\theta} \mathbb{E}_{(x,y)} (\mathcal{L}(y, f(x|\theta)))$, where $\mathbb{E}$ is the expectation operation and $\mathcal{L}(\cdot, \cdot)$ is the loss function. To make it easier to track notations, we present the basic notations mentioned in this article in Table 1.

2.2 Adversarial Perturbations

Well-designed adversarial perturbations have been verified to effectively misguide several computer vision tasks, such as face attribute classifiers, face recognition models, image and video object detectors, semantic segmentation, and object detection. Adversarial perturbations are also valid for attacking real-world computer vision models, including road sign recognition, 3D object recognition, and person detection. They can also attack well-trained speech and audio recognizers, natural language processing, malware detection systems, and medical machine learning models [26].

While adversarial perturbations have already been verified on different modalities of data, the implementation of adversarial perturbations on nonimage datasets is still indirect and complex. The mechanism of restricting the magnitude of adversarial perturbations is closer to images, since perturbations on images were originally thought to be small enough to make them imperceptible by humans. Therefore, state-of-the-art attacks and defense methods are evaluated on image datasets. The most commonly used image datasets for defense method evaluation in existing works are MNIST, CIFAR10, CIFAR100, SVHN, and ImageNet [3]. In contrast to other datasets, ImageNet is an overly large image dataset with many samples and categories, which makes the training procedure on ImageNet more difficult [27]. In image-based tasks, convolutional neural network (CNN) architectures are often employed as threat models to evaluate the performance of attack and defense methods. Commonly used CNN models include fully connected convolutional networks (FCNs) and some complex CNN architectures, including ResNet, VGG, LeNet, AlexNet, Inception-Resnet, and Wide-Resnet [3].

Table 1. Basic Notations and Their Corresponding Descriptions

Notation	Description	Notation	Description
x_i	The i -th value of x .	$\nabla_x(\cdot)$	Gradient on x .
$\mathbb{R}^d$	Real coordinate space of d dimension.	$p(\cdot)$	Distribution function.
$\ x\ _p = (\sum_{i=0}^{d-1} x_i ^p)^{\frac{1}{p}}$	ℓ_p norm.	$\mathcal{L}(\cdot)$	Loss function.
$(\eta \mid \ \eta\ _p \leq \varepsilon)$	ε -norm bounded attack.	$\mathbb{E}_{\mathcal{D}}$	The expectation based on data distribution $\mathcal{D}$.
$x' = x + \eta$	Adversarial example.	$\phi_i^{(j)}(\cdot)$	The i -th value of layer j on DNN.
$f(x \theta)$	Decision function with trainable parameter θ /threat model.	$\operatorname{argmax}(x)$	Operation that finds the index of maximum value of x .
ξ	Step size of PGD iteration.	λ, c	Hyperparameters.
$\mathcal{B}_p(x_0, \varepsilon) = \{x \mid \ x - x_0\ _p \leq \varepsilon\}$	ℓ_p bounded sphere around x_0 .	$\frac{\partial y}{\partial x}$	The derivative of y on x .
$\operatorname{Clip}_{\varepsilon}\{x\}$	Clip the value x to be less than ε .	$ \cdot $	Absolute value or the total number of samples.
$\Pi_{S_x}(\cdot)$	The projecting operator for optimal value in region S_x .	$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2 .

The magnitude of adversarial perturbations is often constrained by ℓ_p norm magnitudes $\|\eta\|_p \leq \varepsilon$. Generally, the strength of adversarial perturbations is quantized by their effectiveness on threat models with restricted magnitude. The size of magnitude is also an important factor for measuring the strength of attacks on defenses [21]. Attacks could be further divided into two categories according to the knowledge of threat models. Those attacks with full knowledge are called *white-box attacks*; *black-box attacks* are assumed to have no access to any threat model information. Therefore, black-box attacks are widely considered to be closer to realistic perturbations and white-box attacks are more powerful since they could lead DNN models towards the worst-case wrong predictions [5].

Intuitively, a ℓ_p norm bounded attack η_l that leads the model prediction towards class l could be calculated by solving a normed box-constrained problem:

$$\eta_l = \min_{\eta} \|\eta\|_p, \text{ s.t. } \operatorname{argmax} f(x) = k, \operatorname{argmax} f(x + \eta) = l, l \neq k. \quad (1)$$

Szegedy et al. [2] propose a box-constrained L-BFGS method to solve the problem above by transforming Equation (1) into $\min_{\eta} \|\eta\| + \mathcal{L}_f(x + \eta, l)$, $\text{ s.t. } x + \eta \in [0, 1]^d$, where $\mathcal{L}_f(\cdot)$ measures the distance between $x + \eta$ and class l under $f(\cdot)$, perturbed samples $x + \eta$ normalized to be less than 1 to maintain semantic meaning. Whereas the computational process of L-BFGS is expensive and time-consuming, Goodfellow et al. [8] later proposed a simpler nontargeted attack called the *fast gradient sign method* (FGSM). The FGSM directly utilizes the gradient of loss functions on clean inputs to approximate optimal perturbations $\eta = \varepsilon \cdot \operatorname{sign}(\nabla_x \mathcal{L}(x, l))$. The FGSM is a one-step attack that performs one step towards the worst-case direction. Kurakin et al. [28] further propose the basic iterative method (BIM) and extend the FGSM by iteratively employing FGSM steps to generate a more powerful iterative adversarial sample $x' = \operatorname{Clip}_{\varepsilon}\{x' + \xi \cdot \operatorname{sign}(\nabla_x \mathcal{L}(x', y))\}$. Similarly, Madry et al. [16] argue that the strongest adversarial sample could be attained by employing the projected gradient descent (PGD) method $x' = \Pi_{S_x}(x' + \xi \cdot \operatorname{sign}(\nabla_x \mathcal{L}(x', y)))$. In addition to

Table 2. Overview of Attacks Presented in This Article

Category	Subcategory	Paper
White-box attack	Targeted	[2, 31] [30]
	Untargeted	[8] [16] [29, 35]
Adaptive white-box attacks	—	[32]
Black-box attack	Gradient Estimation	[36, 37, 41–44]
	Substitute Model	[5, 38, 45]

performing gradient-based steps, Moosavi-Dezfooli et al. [29] suppose that effective adversarial perturbations could also be produced by misguiding normal samples towards decision boundaries and propose the DeepFool method. Papernot et al. [30] verified that modifying the features of inputs according to the forward derivative of the threat model could significantly misguide the predictions of models. Then, they proposed the Jacobian-based saliency map attack (JSMA) method, which utilizes an adversarial saliency map to select features, thus crafting effective perturbations.

With the rapid progress of defense methods such as network distillation [14], many commonly used attacks are later defeated by strong defense methods such as perturbation detection methods. Therefore, Carlini and Wagner [31] propose a more powerful perturbation called CW Attack. CW reformulates a highly nonlinear L-BFGS problem with a more suitable objective function $g(\cdot)$ that satisfies $\arg\max f(x + \eta) = l$ if and only if $g(x + \eta) \leq 0$. By introducing a novel objective function, the optimization problem becomes

$$\min_{\eta} \|\eta\|_p + \lambda \cdot g(x + \eta), \text{ s.t. } x + \eta \in [0, 1]^d. \quad (2)$$

Similar to L-BFGS, the realization mechanism of CW is also complex since they both need to search for an optimal λ by iterative training. Later, several novel methods effectively extended CW to a more convenient method. For instance, Athalye et al. [32] argue that CW could also be constructed through the PGD method. Zhao et al. [33] propose a versatile framework using the alternating direction method of multipliers (ADMM) to conveniently generate any ℓ_p bounded attacks. Instead of only constraining the distance between normal and adversarial examples using the ℓ_p norm, Xu et al. [34] developed the structured attack (StrAttack), which introduces group sparsity into the objective function based on the ADMM framework. Then, they show that the StrAttack achieves better interpretation. Recently, some more convenient and powerful white-box attack generation methods were further proposed [35].

In contrast to white-box attacks, black-box attacks cannot be realized directly with no requisite information. Instead, researchers find that estimating the gradient, loss function, or outputs of agnostic threat models could effectively produce valid attacks [36, 37]. Moreover, since adversarial perturbations are demonstrated to transfer between different architectures and datasets [4], attacking independent pretrained surrogate networks could also produce effective black-box attacks [5, 38].

Except for attacks that are designed based on existing DNN models, Athalye et al. [32] argue that most of the existing defense methods rely on a phenomenon called *obfuscated gradients*, which is a special kind of gradient mask phenomenon and could be circumvented by employing specific mechanisms. The authors propose several attack methods to circumvent obfuscated gradients, including the backward pass differentiable approximation (BPDA) method for evading nondifferentiable functions, the expectation over transformation (EOT) for circumventing models with random noise and the reparameterization method for preventing vanishing or exploding gradients.

We summarize these attacks in Table 2 and further divide existing white-box attacks based on whether they are targeted towards specific classes. The targeted attacks need to iteratively query

Table 3. Overview of 7 Categories on the Origins of Adversarial Perturbations

Origin	Description	Paper
Overlinearity	The DNN architecture is overlinear; hence, small perturbations may be amplified through transmission.	[8, 47, 48]
High Dimension	The vulnerability of any decision-making models is caused by the high dimension of data manifold.	[49–52]
Test Error on Noise	The vulnerability of DNNs against adversarial perturbations could be a consequence of the vulnerability of DNN models under noise.	[53–55]
Decision Boundaries	The boundary of standard training used for decision-making does not determine the real data manifold; hence, models are easily attacked.	[11, 25, 46, 56, 57]
Insufficient Data	The total number of samples used for training is insufficient for acquiring robustness.	[10, 58–61]
Nonrobust Features	Nonrobust and robust features all exist in any classifier, and nonrobust features cause vulnerability.	[18]
Training Mechanism	The mechanism used for simplifying the training procedure or standard test error cause the vulnerability of DNN models to be inevitable.	[17, 25]

models to verify their effectiveness, and the loss function of the targeted class and model prediction is impractical to measure. Hence, targeted attacks are more difficult to generate, while they are more practical in real-world circumstances [39]. Moreover, some existing defense methods that have been shown to be valid in defending existing attacks were later verified to offer limited robustness since these methods were designed to defend existing attacks instead of increasing the robustness of models. Therefore, the latest attack generation methods try to generate adaptive attacks that could adapt to specific defense methods to circumvent these attacks [32]. During various categories of adversarial perturbation generation methods, PGD, CW, and their derivative methods have been the most commonly used attacks and act as the benchmarks for evaluating novel defense methods in recently proposed work [16, 28, 40]. Finally, two lines of work have been proposed for adversarial perturbation defense—exploring the origins of adversarial perturbations and designing strong defense methods against existing attacks.

3 THE ORIGIN OF ADVERSARIAL PERTURBATIONS

Adversarial perturbations are verified to transfer between different datasets and architectures [4]. Most of the well-trained classification models that have high accuracy on regular datasets are found to be vulnerable to adversarial perturbations [4, 5, 46]. A large number of studies have tried to interpret this phenomenon from various viewpoints. Szegedy et al. [2] first argue that adversarial examples are caused by the discontinuous mapping between inputs and outputs. Subsequently, some other hypotheses also interpret the transferability phenomenon in terms of other aspects. Researchers have also developed diverse frameworks and structures to analyze the vulnerability of DNN models. In this section, we show the basic concepts of the state-of-the-art hypotheses on the origins of adversarial perturbations, as listed in Table 3 in detail. The origins of adversarial perturbations are thought to be key elements for further understanding the vulnerability of DNN models and enhancing their robustness.

Fig. 1. The concentric sphere datasets mentioned in [49]. $\circ$ and $\times$ are two sets of samples labeled 0 and 1, respectively. The green region indicates the decision sphere that recognizes samples as 0. (a) is the decision boundary of the models after standard training. (b) denotes the decision boundary that is perturbed towards z .

3.1 Overlinearity

The vulnerability of DNNs may be induced by the local linearity of neural networks. Eykholt et al. [24] argue that the input of image-based DNN classifiers always has large dimensions and that small perturbations on each dimension could be amplified through a dot product. On a simple linear neural layer, given a ℓ_∞ bounded perturbation that satisfies $\|\eta\|_\infty \leq \epsilon$, by assigning a perturbation $\eta = \text{sign}(w)$, $w \in \mathbb{R}^n$, $\frac{1}{n}w_i = m$, the output of the layer grows $\epsilon \cdot mn$ to misguide the predictions of the models. Based on this description, the authors propose the FGSM, which is derived from a linear classifier and is demonstrated to show effectiveness on nonlinear classifiers. This phenomenon further verifies the existence of local linearity on nonlinear structures. Luo et al. [47] further demonstrate that overlinearity in CNNs could lead CNNs to be vulnerable to adversarial examples, and they reduce the influence of perturbations by adding a foveation mechanism into CNN architectures. Similarly, Fawzi et al. [48] found that linear classifiers are more sensitive to adversarial perturbations than quadratic classifiers.

3.2 High Dimension

The vulnerability of DNNs to small perturbations could also be a natural consequence of the high dimensions of the data manifold. Gilmer et al. [49] analyze the phenomenon of adversarial examples by introducing two concentric sphere datasets that are demonstrated to perform similarly with image datasets. The concentric sphere datasets can be denoted as $x \in \mathbb{R}^d$ and the corresponding layer y : $\|x\|_2 = 1.0, y = 0$ and $\|x\|_2 = R > 1, y = 1$. For any classifier, let p be the accuracy of the model in the inner sphere, S_0 represents the accurately classified points in the inner sphere and E to be/present the points that are misclassified by the model in the inner sphere, as shown in Figure 1. The authors verify that $d(E) = O(\Phi^{-1}(p)/\sqrt{d})$, where $\Phi^{-1}(\cdot)$ is the inverse normal *cdf* function and $d(E) = \mathbb{E}_{x \sim S_0} d(x, E)$ is the average distance between points in S_0 and their nearest points in E . $d(E)$ could effectively quantize the robustness of classifiers. Obviously, $d(E)$ is inversely proportional to d . Therefore, for the high-dimension dataset, any classifier that wants to enhance robustness needs to significantly increase the standard accuracy on test samples. This assumption effectively interprets the phenomenon that attaining good robustness on MNIST is much easier than that on CIFAR10 and ImageNet.

Similarly, Shafahi et al. [50] also analyze the vulnerability of models to multiclass classification conditions and conclude that the vulnerability of DNN models lies in the high dimensionality

of datasets. Fawzi et al. [51] prove that any classifier will be vulnerable to small perturbations under high-dimensional data generated from smooth generative models. The authors present the in-distribution robustness $r_{in}(x)$ by measuring the minimum vector to disturb the model:

$$r_{in}(x) = \min_r \|g(z+r) - x\|, \text{ s.t. } \operatorname{argmax} f(g(z+r)) \neq \operatorname{argmax} f(x), \quad (3)$$

is in reverse proportion to the data dimension of d when the generator g is smooth. Mahloujifar et al. [52] further verified that if the metric probability space of test samples is concentrated, any classifier designed for a high-dimensional dataset could not avoid potential perturbations.

3.3 Test Error on Noise

The robustness against noise on input data could be viewed as a prior of robustness against adversarial perturbations. Fawzi et al. [53] argue that adversarial perturbations could be connected with random noise and propose two noise regimes called the *random noise regime* and the *semirandom noise regime* to verify this assumption. In the random noise regime, the input data are perturbed towards random directions. In the semirandom noise regime, the data are perturbed towards random subspaces. For a data point $x \in \mathbb{R}^d$, the decision function $\hat{k}(x) = \max_k f_k(x)$ and a subspace $S \in \mathbb{R}^L$, the worst-case perturbation under S can be denoted as:

$$r_S^*(x) = \min_{r \in S} \|r\|_2, \text{ s.t. } \exists k \neq \hat{k}(x) : f_k(x+r) \geq f_{\hat{k}}(x+r). \quad (4)$$

For the random regime, S is a one-dimensional subspace $S = t \cdot v$, where $v \in \mathbb{R}^{d-1}$ is a random subspace. For the semirandom regime, S could be an arbitrary dimension vector. The authors also state that $\|r_S^*(x)\|_2$ could be viewed as the minimal distance between x and the decision boundary on S . Through theoretical analysis, the authors show that for any linear model, the distance of the worst-case perturbation and subspace distance satisfy $\|r^*(x)\| = O(\sqrt{\frac{m}{d}} \|r_S^*(x)\|)$, where $r^*(x)$ is the worst-case perturbation:

$$r^*(x) = \min_{r \in \mathbb{R}^d} \|r\|_2, \text{ s.t. } \exists k \neq \hat{k}(x) : f_k(x+r) \geq f_{\hat{k}}(x+r). \quad (5)$$

Hence, for high-dimensional data, any linear classifier that is vulnerable to random noise could be vulnerable to adversarial perturbations with high probability.

Franceschi et al. [54] further verify that the relationship between adversarial perturbations and random noise still holds on commonly used nonlinear classifiers. Ford et al. [55] argue that the robustness of classifiers against adversarial perturbations could be a consequence of robustness against random noise. Li et al. [62] also assert that there exists a connection between the robustness of DNN models against adversarial perturbations and additive random noise on clean samples, and this connection could be scaled to larger networks.

3.4 Decision Boundaries

Some experiments suggest that the local linear property alone cannot cause the vulnerability of DNN models to adversarial perturbations [16, 32]. Alternatively, Tanay and Griffin [11] find that the existence of adversarial examples is the consequence of the decision boundary being overly close to the submanifold of normal data. Small perturbations on that submanifold could lead normal data to cross decision boundaries to misguide the classifier [46]. The authors further present a novel concept, called *boundary tilting*, that could effectively interpret the existence of adversarial examples if sampled data have different variances in different directions. Boundary tilting denotes a phenomenon in which the decision boundary of a well-trained classifier is tilted beyond the submanifold of normal sample data, as shown in Figure 2.

Fig. 2 (a) shows the phenomenon of boundary tilting, where the black space is the submanifold of sampled data and the red space is the decision boundary, which is tilted towards the submanifold of sampled data. Therefore, samples that are disturbed towards the decision boundary may break away from the manifold of normal data. $\times$ and $\circ$ represent samples of different classes. (b) is the interpretation of why boundary tilting may suffer from adversarial examples. Consider two classes of samples I and J that are symmetric from the decision boundary B . Samples i and j are the centers of I and J , respectively. Mirror samples $m(i, B)$ of i towards B will not suffer from adversarial examples, while if the boundary tilts to C , mirror samples $m(i, C)$ will derive from J and cause adversarial examples.

Moreover, Dong et al. [56] suggest that the decision boundaries of some DNN models are actually incorrect and that they cannot detect genuine semantic objects. Kim et al. [57] argue that boundary tilting could be employed to explain the phenomenon that the loss gradients of models trained with the adversarial training method [16] align better with human perception [25]. Experimental results verify that the adversarial examples of models trained with the adversarial training method are closer to the manifold of normal data than those of models trained with standard training.

3.5 Insufficient Data

Some researchers also argue that the vulnerability of DNN models is caused by incomplete training, which is attributed to insufficient data. Schmidt et al. [10] analyze the properties of the binary classifier theoretically under Gaussian models. Gaussian models draw data from a Gaussian distribution $y \in \{\pm 1\}$ randomly and x from $\mathcal{N}(y \cdot \theta^*, \sigma^2 I)$, where θ^* is the mean of input data, and σ is the uniform variance of each dimension on input data. The authors verify that under the restriction $\|\theta^*\|_2 = \sqrt{d}$ and $\sigma \leq c \cdot d^{1/4}$, the classifying accuracy under any ℓ_∞ attack $\|\eta\|_\infty \leq \epsilon$ is at most 1% if the number of samples satisfies

$$n \geq \begin{cases} 1, & \epsilon \leq \frac{1}{4}d^{-\frac{1}{4}}, \\ c \cdot \epsilon^2 \sqrt{d}, & \frac{1}{4}d^{-\frac{1}{4}} \leq \epsilon \leq \frac{1}{4}. \end{cases} \quad (6)$$

Therefore, more samples are required to obtain satisfying robustness for image-based models with very large dimensions. Hendrycks et al. [58] also demonstrate that using well-trained models pre-trained on larger datasets could effectively enhance the robustness of current models. Carmon et al. [59] and Stanforth et al. [60] show that unlabeled data could also effectively enhance the robustness of DNN models. Mao et al. [61] argue that DNN models trained on multiple tasks show more robustness than those trained on single tasks.

3.6 Nonrobust Features

Nonrobust features may exist in any DNN model and could inevitably cause the vulnerability of DNN models [18]. Considering a binary classification function $f(\cdot)$, a feature used for standard

classification and robust classification is evaluated as ρ useful and γ robustly useful features if it satisfies

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} \left[y \cdot \left(w_i \cdot \phi_i^{(l-2)}(x) \right) \right] \geq \rho, \quad \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\min_{\eta} y \cdot \left(w_i \cdot \phi_i^{(l-2)}(x + \eta) \right) \right] \geq \gamma, \quad (7)$$

where $\phi_i^{(l-2)}$ is the i -th value of activations at the penultimate layer and considered as the learned features of the input, and w_i is the parameter for final decision-making. Features with strong robustness against attacks are extracted as robust features, and other features are regarded as non-robust features. The authors verified that the classifier with robust features could defend against perturbations effectively through only standard training. Moreover, robust and nonrobust features can solely achieve good accuracy on a standard dataset through standard training. The phenomenon that adversarial training methods cannot avoid regular accuracy declination indicates that nonrobust features are also valuable information for prediction, and nonrobust features are shrunk during training [63].

3.7 Training Mechanism

The training mechanism is also likely to induce the vulnerability of DNN models. Bubeck et al. [17] state that adversarial examples may be the inevitable consequence of computational limitations and constraints that are utilized for simplifying the procedure of training DNN models. Nakkiran et al. [25] argue that accuracy and robustness could not be attained at the same time. The authors propose a set of data $\{(x, y) : y \in \{-1, 1\}, x_1 = \{-y, \frac{p}{1-p}\}, x_2, \dots, x_{d+1} \sim \mathcal{N}(\eta y, 1)\}$, and a natural classifier could be denoted as $f_{avg}(x) = \text{sign}(w_{unif}^\top \cdot x)$, $w_{unif} = [0, \frac{1}{d}, \dots, \frac{1}{d}]$. The prediction probability Pr is calculated as

$$Pr(f_{avg}(x) = y) = Pr\left(\frac{y}{d} \sum_{i=1}^d \mathcal{N}(\eta y, 1) > 0\right) = Pr\left(\mathcal{N}\left(\eta, \frac{1}{d}\right) > 0\right). \quad (8)$$

The authors prove that adding ε -bounded perturbation on features could lead the classifier towards $-y$:

$$\min_{\|\delta\|_\infty \leq \varepsilon} Pr(\text{sign}(x + \delta) = y) \leq Pr\left(\mathcal{N}\left(\eta, \frac{1}{d}\right) - \varepsilon > 0\right) = Pr\left(\mathcal{N}\left(-\eta, \frac{1}{d}\right) > 0\right). \quad (9)$$

Furthermore, the authors demonstrate that a classifier with $1 - \delta$ standard accuracy has at most $\frac{p}{1-p} \delta$ robust accuracy under any ℓ_∞ -bounded perturbation if $\varepsilon \geq 2\eta$.

3.8 Summary

Earlier in this article, we comprehensively introduced existing hypotheses on the origins of adversarial perturbations. While several hypotheses on the origin of adversarial perturbations have been proposed, the exact explanation of the adversarial perturbations is still a controversial issue. Specifically, insufficient data and high-dimensional opinions argue that the vulnerability of DNNs is caused by insufficient training. Since the dimensions of image datasets are always very large, more data, including unlabeled data, are needed to train models for robustness. Moreover, the test error on noise opinion tries to bridge a connection between the robustness of DNN models and potential noise on datasets. It argues that the training mechanism combined with random noise on raw data could act as the data augmentation for robustness.

In addition, in contrast to the viewpoint of insufficient training, other hypotheses regard the vulnerability of DNN models as the inner structure problems of DNN architectures. The overlinearity

perspective attributes the significant influence of small adversarial perturbations to the inner linearity character of the architecture of DNN. Small attacks on original datasets are amplified by information transformation on linear neural networks. The nonrobust feature perspective demonstrates that features derived from raw data show inherent robust and nonrobust attributes, and nonrobust features may inevitably cause vulnerability. The training mechanism perspective also argues that the robustness of DNN could not be attained combined with a standard performance at the same time, according to the training mechanisms of DNN models. Finally, the decision boundaries perspective considers the vulnerability phenomenon of DNNs as the existence of imperfect decision boundaries.

We found that while there were seven hypotheses, most were not in conflict, and the vulnerability of DNNs may be the consequence of the joint effect of different factors. For instance, the FGSM attack derived from the linear attribute of DNN could effectively attack nonlinear models, while the overlinear attribute alone could not interpret the existence of adversarial perturbations [16, 32]. Data augmentation methods could also improve the performance of decision boundaries to alleviate the boundary tilting phenomenon [25]. Works that try to discover more concrete inner links between different hypotheses are still limited.

In addition to interpreting the existence and transferability of adversarial perturbations, the origin of adversarial perturbations could also provide references for designing effective defense methods. For instance, high-dimensionality could interpret the phenomenon that image-based models are easy to attack and dimension reduction methods could effectively mitigate the influence of adversarial perturbations. Moreover, since most image-based methods are trained incompletely, adding more data for data augmentation, generative models, and samples from other datasets could all enhance the robustness of DNN models. Specifically, some data augmentation methods, such as adversarial training that utilizes worst-case adversarial examples and stochastic defense methods that add random Gaussian noise to normal samples, can significantly improve the effectiveness of DNN models. In addition, the test error on the noise category suggests defense methods to add noise to the input for training towards robustness against potential attacks.

4 DEFENSE METHODS

Based on the realization mechanism, existing defense methods can be divided into five categories, including perturbation detection, input transformation, stochastic defense, adversarial training, and certified robustness.

(1) First, while overly small perturbations are visually unrecognized by humans, adversarial examples are assumed to have different architectures compared with normal samples. These perturbations could be detected by extracting features from samples or intrinsic network structures using **perturbation detection** methods. Perturbation detection methods are easy to implement on well-trained models and could avoid affecting their regular performance.

(2) **Input transformation** methods denote some specific processes on samples to mitigate the influence of adversarial perturbations. Specifically, some mechanisms performed on DNN structures that could effectively reform normal samples from adversarial examples or mitigate perturbations through denoising architectures are verified to effectively reduce the influence of adversarial perturbations.

(3) **Stochastic defense** methods that make DNN models more difficult to attack by adding noise or randomness processes could mitigate the influence of adversarial perturbations. Compared with deterministic models, attacks on stochastic models are more difficult to generate since invaders always have no inner information of random noise.

(4) **Adversarial training** methods attempt to correctly recognize the real class of adversarial examples by updating the objective functions of models and training with mixed samples.

Fig. 3. An overall illustration of defense methods proposed in this article.

Researchers argue that DNN-based classifiers that are trained to recognize the real class of worst-case attacks [16] show strong robustness against potential attacks. Therefore, adversarial training methods try to promote DNN models to perform a min-max manner training procedure. The max process is performed to find the worst-case attacks, and the min process updates the model parameters to minimize the loss functions under previously generated attacks.

(5) Except for leveraging defense methods for robustness against existing attacks, **certified robustness** methods try to verify the vulnerability of DNNs precisely or provide provable robustness regions against any manipulations. Generally, DNN architectures consist of a large number of parameters; these parameters are connected with nonlinear activation functions. This makes the decision functions of DNNs highly nonconvex. Therefore, verifying the properties of DNN models is thought to be impractical [64]. Instead, existing certified robustness methods mainly focus on offering approximately guaranteed robustness regions against bounded attacks on specific neural network-based models through inference. The inferred robustness regions could be further approximated with differential bounds, and those bounds are optimized towards better robustness.

Figure 3 summarizes the realization of defense methods discussed earlier intuitively. Different categories of defense methods are surrounded by specific boxes with different colors. Perturbation detection methods are marked in green boxes, input transformation methods are in yellow boxes and stochastic defense methods are in red boxes. The realization mechanisms of these three categories are uniform, and they can all be implemented on raw samples or network architectures to mitigate adversarial perturbations. Adversarial training methods are in the blue box; their realization mechanisms always need to obtain normal samples and perturbed samples for mixed training. Additional samples are required for unsupervised and semisupervised adversarial training. Certified robustness methods are in the black box; they are implemented to approximate provable robustness regions of DNN models under restricted perturbations or to implement specific regularizations on DNN architectures.

Notably, the mechanism of adversarial training and certified robustness is to enhance the robustness of DNN models through retraining while other categories of defense methods try to mitigate the influence of adversarial perturbations. Generally, methods that need retraining are thought to be complex and time-consuming because the training procedure of DNN models is difficult, especially on large image datasets such as CIFAR10 and ImageNet. Therefore, methods that do not need to retrain initial models are easier to realize on well-trained models. In addition, the defense

Table 4. Overview of Classic and State-of-the-Art Work on Defense Methods

Subcategory		Advantage	Disadvantage	Need Retraining	Model In-dependent	Paper
Perturbation Detection	Independent Detectors	No retraining or Model independent	Lack generalization		✓	[6, 7, 65–70] [71]
	Model-Based Detectors					[72–76]
Input Transformation	Formal Process	Attack independent	High cost		✓	[71, 77–81]
	Normal Sample Reconstruction				✓	[6, 15, 82, 83]
Stochastic Defense		Easy to realize	Difficult to find a balance	✓	✓	[86–90]
						[14, 91–95]
Adversarial Training	Supervised Adversarial Training	Enhance the intrinsic robustness	Longer training time	✓		[8, 16, 27, 28, 95–101]
	Semi/unsupervised Adversarial Training			✓		[9, 59, 60, 102–104]
Certified Robustness		Provide provable robustness	Poor scalability	✓		[105–113] [114, 115] [12, 116–118]

methods that need to extract features from the inner architectures of DNNs or calculate the gradients of DNN models are thought to be model-dependent, and their complexity is in direct proportion to the model size. Hence, model-dependent methods are even complex to perform on in-depth DNN models. A brief overview of their characteristics is given in Table 4.

Most perturbation detection and input transformation methods do not enhance the robustness of models; they exclude adversarial samples instead. The mechanism of detecting or transforming adversarial examples directly through analyzing samples is model independent; thus, the realization procedure is light and could transfer between independent models shared with different architectures. Most adversarial training and certified robustness methods are model dependent and time-consuming.

The performance of state-of-the-art defense methods is usually evaluated on existing strong bounded adversarial perturbations such as PGD, CW, and their derivatives. However, the evaluation criteria of different categories are diverse. For some perturbation detection methods and stochastic processes, such as SAP [92] and dropout [93], robustness and standard accuracy could not be attained at the same time. Generally, improving the efficiency of resisting attacks is contrary to standard performance in some situations. Therefore, the performance of perturbation detection methods is evaluated through the false-positive rate (FPR) and true-positive rate (TPR). FPR denotes the ratio of recognizing normal samples as adversarial examples, and TPR is the ratio that figures out adversarial examples accurately. While oversize TPR inevitably implies excess false recognition, TPR is always reported at the premise of maximizing FPR. For some robust trained models, the ratio of recognizing the real class of perturbed samples is also relatively low compared with that with standard training. Therefore, for scenarios in which security is highly demanded, detectors with outstanding TPRs are preferred [21]. Similarly, some stochastic defense methods could also cause standard accuracy declination. Therefore, the performance of these methods is always measured as the robustness improvement under a given standard accuracy decline.

The performance evaluation of adversarial training, robustness certification, and some input transformation methods are more direct since robust and standard accuracy enhancement could be attained at the same time through training. The standard accuracy under regular samples has also been the key evaluation index for evaluating the performance of adversarial training. The complexity and performance evaluation of existing defense methods are further illustrated in Section 4.6. The basic advantages and disadvantages of different categories of defense methods are given in Table 4. Specifically, most of the perturbation detection methods need no retraining and are model independent—hence, are easy to implement—while they are designed for specific attacks, thus lack generalization on agnostic attacks. In most cases, input transformation methods are attack independent—they show more generalization. While the manifold of data for reconstructing normal samples are difficult to attain, the reconstruction models may be costly to discover the intrinsic structure of normal samples. Similar to the perturbation detection methods, most of the stochastic defense methods are model independent, hence are easy to realize. While excessive training could heavily decrease the regular performance of models, it is always difficult to find a balance between robustness and standard performance. Adversarial training methods could enhance the intrinsic robustness of DNN models instead of mitigating the influence of existing attacks, while the process of adversarial training is several times longer than that of standard training. Finally, certified robustness methods could provide provable robustness against specific norm bounded attacks, but they are difficult to be scaled to large DNN models. Next, we will introduce the classic and state-of-the-art defense methods of different categories in detail.

4.1 Perturbation Detection

Adversarial examples could be detected by training detectors to analyze specific features. Features used for detection could be the inner features of raw samples or the representations of models. Therefore, perturbation detection methods could be implemented using either model-based or independent detectors. Independent detection methods are based on some independent architectures. Model-based detection methods are always constructed by performing specific statistical tests on mixed samples.

4.1.1 Independent Detectors. Adversarial examples perturbed by the FGSM could be distinguished from normal samples using a binary classifier [65]. Afterwards, Grosse et al. [66] introduced a kernel-based test for a perturbation detector that measures the distance between two samples x_1 and x_2 through the biased estimator of the true maximum mean discrepancy (MMD):

$$MMD[F, X_1, X_2] = \max_{h \in H} \left(\frac{1}{n} \sum_{i=1}^n h(x_{1i}) - \frac{1}{m} \sum_{j=1}^m h(x_{2j}) \right), \quad (10)$$

where H is a gather of kernel functions for measuring the distance between two probabilities, h is the chosen kernel function that maximizes the distance between different samples, and n and m are the dimensions of x_1 and x_2 . The authors verify that adversarial examples have a larger MMD compared with normal samples than that between disparate normal samples. Metzen et al. [141] built a subnetwork that branches off the classifier network at specific layers and validated its effectiveness in detecting adversarial examples. Subnets are assumed to have the same network architectures as the original models, which could help extract valuable information from raw inputs for better estimation. Subnetworks could be inserted at the arbitrary layer on the original models. The authors also further verify the performance of subnetworks at different insertion locations. The network parameters of the classifier and detector can be updated simultaneously by replacing the objective function from the standard loss function $\mathcal{L}(x, y)$ to $\mathcal{L}(x, y) = (1 - \lambda) \cdot \mathcal{L}(x, y) + \lambda \cdot \mathcal{L}_{det}(x, 1)$, where $\mathcal{L}_{det}(x, 1)$ is the cost function of the detector for recognizing normal samples as adversarial

examples. Meng et al. [6] proposed an autoencoder-based detector called MagNet, which consists of an encoder and a decoder to explore the intrinsic features of normal samples. Adversarial examples could be effectively detected by measuring the distance between features extracted from the encoder. Moreover, a simpler cascade classifier on convolutional networks to detect adversarial examples is proposed in [67]. In this work, the cascade classifier is trained on the statistics of outputs from each convolutional layer for perturbation detection. The cascade classifier is nonsubdifferentiable and, thus, has a strong defense ability in detecting gradient-based adversarial perturbations.

Perturbation detection methods could also be implemented by exploring features learned by DNN models. Feinman et al. [7] demonstrate that adversarial examples could be detected by Bayesian uncertainty (BU) and kernel density estimation between input x and class k :

$$K(x, X_k) = \sum_{x_i \in X_k} k_\sigma(\phi(x), \phi(x_i)), \quad k_\sigma(x, y) = \exp\left(-\frac{\|x - y\|^2}{\sigma^2}\right), \quad (11)$$

where X_k represents samples with category k . The Bayesian uncertainty of adversarial examples is verified to be higher than that of normal samples. Lu et al. [68] constructed a more powerful perturbation detector called SafetyNet. SafetyNet utilizes an RBF-SVM structure that takes the discrete codes from ReLU activation functions to detect adversarial examples $g(c) = \sum_{i=1}^N \alpha_i \cdot y_i \cdot \exp(-\frac{\|c - c_i\|^2}{2\sigma^2}) + b$, where c_i and y_i are support vectors (clean samples) and corresponding classes, respectively, and α_i denotes the parameters calculated by a support vector machine (SVM). σ^2 is a hyperparameter for measuring the threshold value of the detection mechanism.

In addition, some image data processing methods have been verified to effectively enhance the performance of perturbation detectors. Hendrycks and Gimpel [69] and Bhagoji et al. [70] argue that several data transformation methods, such as principal component analysis (PCA), could be applied to detect adversarial examples. Adversarial images processed by PCA and other transformation methods show greater differences compared with normal images. Xu et al. [71] demonstrated that the accuracy of perturbation detectors could be increased by feature squeezing (FS), and they argued that three input space reduction methods, including bit-depth reduction, median smoothing, and nonlocal means methods, could effectively make adversarial examples more distinguishable on classifiers [119].

4.1.2 Model-Based Detector. Some of the independent detectors were later verified to be easily bypassed if attackers knew the inner parameters of detectors [72]. Moreover, although previous methods could effectively detect adversarial examples on MNIST with high probability, most were later found to show little performance on larger datasets, such as CIFAR10. Therefore, some model-based detectors were proposed for more efficient attack detection.

Statistical tests on DNN models are found to recognize adversarial examples effectively. Lee et al. [73] demonstrate that adversarial examples could be detected through the confidence score calculated using Mahalanobis distance (MD) between the sample x and the closest distribution $M(x) = \max_c -(f(x) - \mu_c)^\top \Sigma^{-1}(f(x) - \mu_c)$. The authors assume that the class-conditional softmax-based distribution of the pretrained model $P(f(x)|y = c)$ follows the multivariate Gaussian distribution $\mathcal{N}(f(x)|\mu_c, \Sigma)$, where the mean and covariance of the class c are estimated through $\mu_c = \frac{1}{N_c} \sum_{y_i=c} f(x_i)$ and $\Sigma = \frac{1}{N} \sum_c (\sum_{y_i=c} (f(x_i) - \mu_c)(f(x_i) - \mu_c)^\top)$. Ma et al. [74] find that adversarial examples have uncommon distance distributions compared with their neighbors, which is estimated by the local intrinsic dimensionality (LID). LID could increase the distinction between adversarial examples and normal samples through $LID(x) = (\frac{1}{k} \sum_{i=1}^k \log \frac{r_i(x)}{r_k(x)})^{-1}$, where $r_i(x)$ denotes the i -th nearest sample around x , and $r_k(x)$ is the maximum distance between x and its nearest k samples. With sampling batch samples randomly from training data and

Table 5. TPRs and FPRs of State-of-the-Art Detection Methods Under Several Commonly Used Attacks

Detector\Attack	FGSM	PGD	CW		JSMA		FPR
			Next	LL	Next	LL	
FS [71]	0.21	0.55	0.98	1.0	0.84	0.89	0.049
MagNet [6]	1.0	1.0	0.96	0.97	0.72	0.74	0.064
LID [74]	0.94	0.96	0.91	0.91	0.92	0.92	0.056
NIC [76]	1.0	1.0	1.0	1.0	0.94	1.0	0.037

Note: DenseNet and CIFAR10 are used as the threat model and dataset for evaluation. For targeted attacks, next attacks that choose the neighbor class of a specific class and least likely (LL) attacks that choose the far most class are also reported in [76].

adversarial examples, the authors construct a detector successfully for perturbation detection and verify that LID is a highly reliable characteristic for perturbation detection. The label change rate of samples could also act as the measurement of perturbation detection. Wang et al. [75] found that adversarial examples are more sensitive than clean samples under random DNN mutations. They formulate the label change rate, which could be obtained through the mutation test (MMT): $\zeta(x) = \frac{\sum_{i=1}^n 1_{f_i \in F \text{ and } \arg\max f_i(x) \neq \arg\max f(x)}}{|F|}$, where F is a gather of mutated models that could induce mutation. Ma et al. [76] proposed a novel method called *network invariant checking (NIC)* and verified that DNN characteristics could also be employed for perturbation detection. They then proposed a perturbation detection method by extracting DNN invariants. The authors find that adversarial examples exploit two common DNN channels called the *provenance channel* and the *activation value distribution channel*. This finding could help to detect adversarial examples. We report the performance of the state-of-the-art detectors under several attacks in Table 5. These detection methods are verified to effectively detect adversarial examples generated by strong attacks with limited standard accuracy deduction. Moreover, since attaining an effective detection ratio may inevitably suffer from accuracy deduction on regular performance [72], sometimes the FPR and the TPR are utilized to construct receiver operating characteristic (ROC) curves to measure the performance of perturbation detectors. While oversize TPR also implies excess false recognition, TPR is always reported at the premise of maximum FPR.

4.2 Input Transformation

Adversarial perturbations could be excluded or mitigated through some transformation methods. Some formal processes on image samples can effectively mitigate attacks added to normal samples. Some reforming mechanisms and denoising processes are also effective transformation methods for defending against adversarial perturbations. Reforming mechanisms are assumed to explore the intrinsic architecture of normal samples and try to reconstruct normal samples from adversarial examples.

4.2.1 Formal Process. Several image processing methods have been verified to show effectiveness for mitigating the influence of small attacks. Dziugaite et al. [77] found that JPG compression could reverse small FGSM perturbations. Xu et al. [71] demonstrated that the feature-squeezing method could also mitigate the influence of attacks. Guo et al. [78] validated the effectiveness of several commonly used image transformation processes and argued that total variation minimization and image quilting are the most effective image processes for mitigating the influence of adversarial perturbations. Random and nondifferential functions could make DNN models more difficult to attack by white-box attacks. The total variation minimization method selects a set

of pixels randomly and utilizes these pixels to reconstruct a simpler image. The image quilting method is implemented by synthesizing images from a set of pixels that are not perturbed by adversarial attacks. Alternative sets of pixels are chosen to contain no adversarial perturbations; hence, synthesized images are assumed to effectively exclude perturbations.

Adding random resizing and random padding to image samples could also mitigate the influence of adversarial perturbations, and they are easy to implement because they are model independent [79]. Random sizing and padding layers resize image data to a random size and pad zero pixels around images. For image data with a size of $W \times H \times 3$, a random resizing layer resizes the input images to $W' \times H' \times 3$, while $|W' - W|$ and $|H' - H|$ are restricted to be small enough to avoid inducing an accuracy drop. A random padding layer randomly pads zero pixels around images by inserting images into a $W'' \times H'' \times 3$ frame and padding zero pixels around images. Similarly, Prakash et al. [80] verified that adding corruptions to images and softening corruptions through a wavelet-based method could effectively denoise corruptions. They proposed a corruption method called pixel deflection for adding noise to mixed normal and perturbed images. The pixel deflection method selects a specific pixel on images randomly and replaces it with another randomly chosen pixel around the neighborhood of the selected pixel.

Stochastic input transformation methods can effectively provide robustness, yet they cannot avoid accuracy deterioration on normal samples. Kou et al. [81] observed that the distribution of softmax outputs under several transformations could effectively express the real class of attacked samples. They present a novel structure called the *distribution classifier* to recognize the features shared by both clean and adversarial samples. Through empirical analysis, the authors found that the distribution classifier could effectively enhance the performance of stochastic input transformation defense methods on both normal and attacked data.

4.2.2 Normal Sample Reconstruction. Instead of processing mixed samples to mitigate perturbations, normal sample reconstruction methods try to reconstruct normal samples from perturbed samples. Reconstruction methods try to explore the intrinsic distribution of normal samples and reform perturbed samples towards the learned distribution. Gu and Rigazio [82] argue that the well-trained denoising autoencoder (DAE), which is constructed to recover normal samples from adversarial examples, could effectively defend against agnostic adversarial perturbations. The DAE is trained to minimize the distance between the reconstructed sample and the normal sample: $\mathcal{L} = \|x - \tilde{x}\|$, $\tilde{x} = DAE(x')$, where $\tilde{x}$ is the reconstructed sample by the denoising autoencoder. Liao et al. [84] argue that after the denoising process, there are also small residual adversarial noises left on adversarial samples after reconstruction, which could be amplified through transmission on a DNN. To extend the DAE to larger datasets such as ImageNet, the authors introduce the high-level representation guided denoiser (HGD) for image classification, which minimizes the distance of activations at each layer on DNN $\mathcal{L} = \|\phi^{(l)}(x) - \phi^{(l)}(\tilde{x})\|$. Xie et al. [85] further proposed that DAEs could be implemented in an end-to-end manner by adding a novel structure called the *denoising block*. The denoising block is implemented using the same structure as deep residual networks, and the 1×1 convolutional network is utilized to adjust trade-offs between eliminating adversarial perturbations and maintaining normal samples. The denoising operation denoises perturbed features with specific filters. For instance, the nonlocal mean computes a denoised feature map y that maps feature x by calculating a weighted mean of x : $y_i = \frac{1}{C(x)} \sum_{j \in \Omega(i)} g(x_i, x_j) \cdot x_j$. The nonlocal mean is verified to be the most effective filter among several commonly used filters, where $g(x_i, x_j)$ is a weight function between features x_i and x_j , which could be Gaussian sets or dot product sets; $C(x)$ represents the normalization function and $\Omega(i)$ denotes the features in a local region around x_i .

The reconstruction procedure could be implemented by analyzing the joint distribution of pixels on image samples [83]; the authors propose a novel method for transforming called PixelDefend. Through empirically analyzing adversarial examples with the PixelCNN method, which quantizes the joint distribution of all pixels on an image, the authors find that the distribution of adversarial examples has a large deviation from that of normal samples. The distribution of normal samples p is trained on normal samples, and the adversarial example x' is transformed by solving the optimization problem $\max_{x^*} p(x^*), s.t. \|x^* - x'\| \leq \varepsilon$. Moreover, autoencoders could also be employed to reform adversarial examples. Meng and Chen [6] proposed MagNet, which could effectively detect adversarial examples and reform normal samples from perturbed samples. The detector and reformer are constructed with an encoder e and decoder d . e and d are trained under clean samples to achieve the intrinsic structure of normal samples by minimizing the reconstruction loss $\mathbb{E}_{x \in \mathcal{D}} \|x - d \circ e(x)\|$. Similarly, Samangouei et al. [15] propose a generative adversarial network (GAN)-based transformation method called the Defense-GAN. Defense-GAN contains a GAN that consists of a generator G for generating approximated samples and a discriminator for distinguishing real data from generated samples. Defense-GAN leverages the generator of a well-trained GAN to reform an adversarial example x' through $x^* = G(z^*)$, $s.t. z^* = \min_z \|G(z) - x'\|_2^2$. By finding the optimal vector z^* by minimizing the distance between the generated sample and x' , Defense-GAN could generate normal samples from attacked samples to purify agnostic perturbations effectively.

4.3 Stochastic Defense Methods

Stochastic defense methods try to replace deterministic models with stochastic structures by adding randomized sets or smoothing the distribution of predictions to increase the difficulty for invaders to produce effective white-box attacks.

Adding ensemble information in the region around specific test samples and voting for final predictions on ensemble information could effectively enhance the robustness of DNN models against small attacks [86]. The authors argue that region-based information acts as adding random noise to test data, and they demonstrate that random-based defense methods can effectively resist any small perturbations. Zantedeschi et al. [87] verified that DNN models could be strengthened by adding Gaussian noise to normal data for data augmentation. They proposed the Gaussian data augmentation (GDA) method $\min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \mathbb{E}_{\Delta x \sim \mathcal{N}(0, \sigma^2)} \mathcal{L}(\theta, x + \Delta x, y)$, where σ is the variance of noise data, which could also represent the magnitude of perturbations. Although random noises are not effective attacks, they are found to augment data samples to enhance the robustness of models. GDA is more convenient to implement than adversarial perturbations since GDA could be added to pretrained models directly. Liu et al. [91] further found that adding noise to network layers during training and prediction procedures could effectively enhance the robustness of DNN models. The authors argue that adding noise before each convolutional layer has the same effect as training different ensemble models. They also demonstrate that adding noise during the training procedure is equivalent to adding Lipschitz regularization on model parameters, which could enhance the generalization of DNN models against potential attacks.

In addition, smoothing the distribution of predictions is demonstrated to be an effective mechanism for mitigating the influence of small gradient-based perturbations. Network distillation, which is designed for network compression, is found to enhance the robustness of neural networks [14]. Neural network distillation is implemented by designing a smaller neural network $f^d(x)$ to mimic the initial model $f(x)$: $\min - \frac{1}{|\mathcal{D}|} \sum_{x \sim \mathcal{D}} (\sum_{i=1,2,\dots,m} f_i(x) \log f_i^d(x))$. The authors demonstrate that smoothing the distribution of output through network distillation could efficiently reduce the sensitivity of DNN models against small modifications on inputs. Adversarial

examples could also be distinguished from clean samples by minimizing the reverse cross-entropy instead of the common cross-entropy [88]:

$$non_ME(x) = - \sum_{i \neq \tilde{y}} \tilde{f}_i(x) \log(\tilde{f}_i(x)), \quad \tilde{f}_i(x) = f_i(x) / \sum_{j \neq \tilde{y}} f_j(x). \quad (12)$$

Minimizing reverse cross-entropy gives high confidence to the true class of normal samples while facilitating uniform distribution of other classes. Moreover, Buckman et al. [89] found that replacing a one-hot encoder with thermometer encoding could make neural networks more robust.

Stochastic processes that are implemented on DNN models could also effectively reduce vulnerability against attacks. Dhillon et al. [92] proposed the stochastic activation pruning (SAP) method, which is implemented by dropping out specific neurons on DNN layers with a given probability. Then, SAP scales up the remaining neurons to maintain activations on each layer. Similarly, Wang et al. [93] argue that the dropout mechanism, which is always applied to prevent overfitting, could also enhance the robustness of DNN models. The authors verify that increasing the dropout rate could effectively enhance the robustness of DNN models against attacks. However, an oversize dropout rate inevitably decreases the test accuracy on the original test data. Hence, they propose a defensive dropout algorithm to search for the optimal dropout rate that could enhance the robustness of DNN models and maintain the original accuracy simultaneously. Wang et al. [94] propose a more effective defense method called *hierarchical random switching* (HRS). HRS replaces basic blocks of neural networks with many channels. Each channel has the same architecture but different weights from the initial block. All channels are trained during the training procedure, and a channel is chosen by switchers for prediction. HRS is verified to enhance the robustness of DNN models effectively accompanied by little accuracy decline on normal samples. Mustafa et al. [90] further demonstrate that overly close proximity of model features belonging to different classes will lead to the vulnerability of DNN models. Promoting the features of different classes will be more disparate; therefore, the authors propose the prototype conformity loss (PCL) function:

$$\mathcal{L}_{PC}(x, y) = \sum_i \left\{ \|g_i - w_{y_i}^c\| - \frac{1}{m-1} \sum_{j \neq y_i} (\|g_i - w_j^c\| + \|w_{y_i}^c - w_j^c\|) \right\}, \quad (13)$$

where g_i is the i -th value of the activation functions at the penultimate layer, $w_{y_i}^c$ is the trainable variable of class y_i connected with g_i that is used for decision-making, and m is the total number of classes. The authors verify that the robustness of DNN models is enhanced by restricting features of each class to be more similar. Bui et al. [95] also present the local compactness and global compactness method to reduce the divergence of the intermediate representations under adversarial samples and samples of different classes for more robustness. Local compactness is implemented by reducing the divergence of the intermediate representations under normal and adversarial samples $\mathcal{L}_{com}^{lc} = \mathbb{E}_{x \sim \mathcal{D}} [\|\phi(x) - \phi(x')\|]$, where $\phi(\cdot)$ is the learned activations. Global compactness is utilized to separate the divergence of representations belonging to different classes while reducing those belonging to the same classes:

$$\mathcal{L}_{com}^{gb} = \mathbb{E}_{x_i, x_j \sim \mathcal{D}} [w_{ij} \|\phi(x_i) - \phi(x_j)\|], \quad w_{ij} = \frac{\alpha - I_{y_i \neq y_j}}{\alpha}, \quad (14)$$

where $\alpha \in (0, 1]$ is a constant for controlling the penalization on small feature variations between samples that belong to different classes.

We review the performance of the state-of-the-art stochastic methods under different attacks in Table 6. HRS could attain excellent DES under different attacks and a compatible adversarial robustness rate of 54.5, while other stochastic methods are broken under PGD attack with

Table 6. Mean Defense Efficiency Score (DES) of State-of-the-Art Stochastic Methods Under CIFAR10 [94]

Stochastic\Attack	FGSM	PGD	CW-PGD	PGD/mean
Dropout [93]	1.09	0.81	0.76	32.62
Gaussian [120]	2.73	6.17	5.47	39.92
Adv Training [16]	5.05	7.37	6.57	-
HRS [94]	32.23	35.43	35.55	69.7

Note: The mean DES quantizes the ratio of the gain on robust accuracy Δd and regular accuracy declination Δt : $\Delta d / \Delta t$.

$\varepsilon = 8/255$. Furthermore, adversarial training inevitably suffers from a larger accuracy decline of 7% compared with 0.48% of HRS.

4.4 Adversarial Training

Adversarial training is regarded as a variant of standard training. The training procedure is implemented on mixed normal samples and adversarial examples, and commonly used adversarial training methods are always realized by adding worst-case attacks directly on objective functions. Training techniques that are designed for standard training could be employed directly on adversarial training. Moreover, based on the training data, adversarial training could also be realized through semi-supervised and unsupervised methods.

4.4.1 Supervised Adversarial Training. The concept of adversarial training was first proposed in [8]. The authors verified that the robustness of DNN models against FGSM could be enhanced through training on mixed normal and adversarial samples by introducing a novel objective function $\tilde{\mathcal{L}}(x, y) = \lambda \cdot \mathcal{L}(x, y) + (1 - \lambda) \cdot \mathcal{L}(x + \varepsilon \cdot \text{sign}(\nabla_x \mathcal{L}(x, y)), y)$. Later, Kurakin et al. [28] extended adversarial training to large-scale datasets by introducing batch normalization:

$$\mathcal{L} = \frac{1}{(m - k) + \lambda k} \left(\sum_{i \in \text{CLEAN}} \mathcal{L}(x_i, y_i) + \lambda \cdot \sum_{j \in \text{ADV}} \mathcal{L}(x'_j, y_j) \right), \quad (15)$$

where m and k are the total number of normal and adversarial examples.

Subsequent adversarial training methods try to solve the robustness methods through min-max training. Huang et al. [96] propose a mechanism of adversarial training by solving the optimization problem $\min_f \sum_i (\max_{\|r^{(i)}\| \leq c} \mathcal{L}(f(x_i + r^{(i)}), y_i))$, where $r^{(i)}$ denotes the bounded worst-case perturbation. The authors also verify that, compared with previous training methods that minimize the loss function of standard and adversarial samples simultaneously, the min-max optimization method could enhance the robustness of DNN models while maintaining high accuracy on standard datasets. Similarly, Shaham et al. [97] proposed robust optimization (RO) $\min_{\theta} \tilde{\mathcal{L}}(\theta, x, y) = \min_{\theta} \sum_{i=1}^m (\max_{\tilde{x}_i \in U_i} \mathcal{L}(\theta, \tilde{x}_i, y_i))$, where m is the total number of samples, U_i denotes the uncertainty sets around $\tilde{x}_i$, and $\tilde{x}_i$ is chosen according to x_i , which could induce the worst-case perturbation. The uncertain sets U_i are generally a p -norm bounded ball around x_i . $\tilde{x}_i$ can be found by calculating an additive adversarial perturbation Δx_i , $x_i + \Delta x_i \in U_i$ that satisfies $\Delta x_i = \max_{\Delta: x_i + \Delta \in U_i} \mathcal{L}_{\theta, y_i}(x_i + \Delta)$. The first-order Taylor expansion of the loss function is employed to estimate the optimal perturbation $\hat{\Delta}_{x_i} = \max_{\Delta: x_i + \Delta \in U_i} \mathcal{L}_{\theta, y_i}(x_i) + \langle \nabla \mathcal{L}_{\theta, y_i}(x), \Delta \rangle$. Although previous min-max optimization methods have enhanced the robustness of DNN models against one-step attacks, they show limited performance against iteratively generated adversarial examples. Therefore, Madry et al. [16] propose a more powerful method through minimizing a saddle point problem $\min_{\theta} \mathbb{E}_{(x, y) \sim \mathcal{D}} [\max_{\eta} \mathcal{L}(x + \eta, y | \theta)]$, s.t. $\|\eta\|_p \leq \varepsilon$. The max

optimization function is solved using the iterative projected gradient descent (PGD) approach $x^{t+1} = \Pi_{S_X}(x^t + \varepsilon \cdot \text{sign}(\nabla_x \mathcal{L}(\theta, x, y)))$. Ye et al. [121] argue that commonly used min-max-based adversarial training methods require larger model capacity for acquiring adversarial robustness compared with standard training. They propose a novel framework for performing concurrent adversarial training and weight pruning; then, they demonstrate that models trained using the novel framework are compressed while maintaining adversarial robustness.

The PGD-based adversarial training method has been validated to be the most effective defense method since it can handle worst-case perturbations in a restricted region. Therefore, PGD-based adversarial training methods are regarded as the universal benchmark to assess the performance of novel defense methods. Recently, several novel works have been presented for more effective adversarial training. Wang and Zhang [122] argue that a smaller gradient magnitude could offer more robustness and propose the bilateral adversarial training method that perturbs both normal images and real labels:

$$\min_{\theta} \left[\max_{x' \in S_x, y' \in S_y} \mathcal{L}(x', y', \theta) \right], S_y = \left\{ z | z \in \mathcal{B}(y, \varepsilon_y), z \geq 0, \sum_i z_i = 1 \right\}, \quad (16)$$

where S_y is the distribution region around y that is utilized to produce adversarial labels. Cheng et al. [99] find that a uniform large perturbation step ε may be harmful for training a robust model since some samples are already around decision boundaries. Therefore, the authors propose the customized adversarial training (CAT) method, which offers a disparate perturbation step for every sample:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^n \max_{x'_i \in \mathcal{B}(x_i, \varepsilon_i)} \mathcal{L}(f(x'_i), y_i), \text{ s.t. } \varepsilon_i = \underset{\varepsilon}{\operatorname{argmin}} \left\{ \max_{x'_i \in \mathcal{B}(x_i, \varepsilon)} f(x'_i) \neq y_i \right\}. \quad (17)$$

Misclassified samples are also verified to have a significant influence on the robustness of DNN models. Wang et al. [100] proposed the misclassification-aware adversarial training (MART) method, which could differentiate correctly classified and misclassified samples during training. The authors define the adversarial risk for misclassified samples, which adds a regularization on misclassified samples and updates the objective function as

$$\mathcal{L}(x_i, y_i | \theta) = \text{BCE}(p(x'_i | \theta)) + \lambda \cdot \text{KL}(p(x_i | \theta) || p(x'_i | \theta)) \cdot (1 - p_{y_i}(x_i | \theta)), \quad (18)$$

where $p_{y_i}(x_i | \theta)$ denotes the value of probability belonging to class y_i , BCE is the boosted cross entropy loss, and KL is the KL divergence. $(1 - p_{y_i}(x_i | \theta))$ is a coefficient that penalizes oversize KL divergence of normal and attacked samples that are misclassified by the model.

In contrast to commonly used defense methods against attacks on specific samples, Shafahi et al. [101] propose the universal adversarial training method for defending against universal attacks. Universal adversarial training could also be implemented in the min-max manner:

$$\min_w \max_{\|\eta\|_p \leq \varepsilon} \mathcal{L}(w, \delta) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}(w, x_i + \eta), \quad (19)$$

where η is the universal perturbation of the given n samples, which could be generated through the gradient ascent method:

$$\max_{\eta} \mathcal{L}(w, \eta) = \frac{1}{N} \sum_{i=1}^N \mathcal{L}(w, x_i + \eta), \text{ s.t. } \|\eta\|_p \leq \varepsilon. \quad (20)$$

Most adversarial training methods mentioned earlier are trained against existing attacks instead of model-agnostic attacks. Tramèr et al. [98] extend adversarial training methods for resisting

Table 7. Performance of the Top 5 and Top 1 Accuracies of Several Adversarial Training Methods [27]

Adversarial Training\Attack	White-box Top 1	White-box Top 5	Black-box Top 1	Black-box Top5
AT [16]	3.9	10.3	36.5	62.3
Ensemble [98]	1.3	6.5	47.1	74.3
ATA [28]	1.5	5.5	-	-
ALP [27]	27.4	55.4	46.7	74

strong black-box attacks by introducing ensemble adversarial training. The authors verify that classifiers trained with standard adversarial training to defend against specific attacks are still vulnerable to some agnostic and black-box attacks. Adversarial training methods should take several attacks generated on other well-trained models. Through empirical analysis, the authors conclude that ensemble adversarial training shows strong generative robustness against powerful agnostic attacks.

To enhance the performance of adversarial training on large datasets such as ImageNet, Kannan et al. [27] further propose a novel method called *logit pairing*. Logit pairing encourages logits of DNN models from a normal sample and its corresponding adversarial samples to be more similar by minimizing the loss function $\mathcal{L}(f(x), f(x'))$. The authors present the adversarial logit pairing (ALP) method by adding logit pairing into the batch objective function:

$$\min_{\theta} \mathcal{L}(M, \theta) + \lambda \cdot \frac{1}{m} \sum_{i=1}^m \mathcal{L}(f(x_i \theta), f(x'_i \theta)). \quad (21)$$

Concretely, Table 7 summarizes the performance of different adversarial training methods under ImageNet and black-box attacks.

4.4.2 Semi-supervised/Unsupervised Adversarial Training. Semi-supervised and unsupervised learning methods are proposed to increase the generalization of DNN models. Several works have verified that models trained through semi-supervised/unsupervised adversarial training methods could also attain identical robustness effectively with fewer samples or enhance the robustness on the same datasets compared with supervised adversarial training.

Miyato et al. [9] first asserted that the worst-case perturbation could be estimated without labels by calculating the KL divergence of output distributions under normal and adversarial examples $\Delta_{KL}(\eta, x, \theta) = KL[p(y|x, \theta) || p(y|x + \eta, \theta)]$. The norm bounded perturbation η^* that induces the greatest variation is defined as the virtual adversarial perturbation: $\eta^* = \max_{\eta} \{\Delta_{KL}(\eta, x, \theta); ||\eta|| \leq \xi\}$. Based on η^* , Miyato et al. [102] proposed the virtual adversarial training (VAT) method $\frac{1}{N} \sum_{n=1}^N \log p(y|x, \theta) + \lambda \cdot \frac{1}{N} \sum_{n=1}^N LDS(x, \theta)$.

In addition, semi-supervised/unsupervised adversarial training could be implemented by introducing novel samples from generative models. Ilyas et al. [103] propose a GAN-based adversarial training method called *invert and classification (INC)*. INC attempts to train a generator to generate more adversarial examples for training, which increases the ability of DNN models to defend against potential attacks. Moreover, Zhang and Wang [104] argue that supervised adversarial training methods that generate adversarial examples based on single samples and corresponding labels inevitably derive the data manifold from the original manifold used for standard training. Labeled-based perturbations that are related only to objective samples thus lack the collective distributional property of overall samples, which could hinder the performance of original models. Hence, they state that novel robust training methods should focus on the inner structure

of samples instead of modifying decision boundaries that could effectively classify samples in the subspace of adversarial examples. Then, they propose the feature scattering method. Feature scattering generates adversarial perturbations in an unsupervised way on the distribution of worst-case adversarial perturbations by maximizing the distance between the features of clean and adversarial samples $v^* = \sum_{i=1}^m v_i \delta_{x'_i} = \max_{v \in S_\mu} D(\mu, v)$, where $\mu = \sum_{i=1}^m u_i \delta_{x_i}$ and $v = \sum_{i=1}^m v_i \delta_{x'_i}$ denote the distribution of clean and adversarial samples, D is the distance between two distributions, and δ_{x_i} and $\delta_{x'_i}$ are the Dirac functions at x_i and x'_i . u_i and v_i are the weights of x_i and x'_i for formulating the data distribution. S_μ is a bounded ball around x_i . Therefore, the training procedure could be updated as $\min \mathbb{E}_{x \in \mathcal{D}} \mathcal{L}(x', y)$, s.t. $v^* = \sum_{i=1}^m v_i \delta_{x'_i} = \max_{v \in S_\mu} D(\mu, v)$.

Semi-supervised adversarial training has gained much attention due to its outstanding performance, and several studies have proposed effective semi-supervised training methods practically and theoretically. The latest studies have verified that larger unlabeled datasets could be employed for semi-supervised training directly and increase the performance of adversarial training. Carmon et al. [59] propose robust self-training (RST) by leveraging unlabeled data. The authors present the self-training method by constructing a cluster of labeled $(x_1, y_1, \dots, x_n, y_n)$ and unlabeled data $(\tilde{x}_1, \dots, \tilde{x}_n)$. An intermediate model $\tilde{\theta}_{intermediate}$ is trained on labeled data by minimizing the standard loss on labeled data $\theta_{intermediate} = \frac{1}{n} \sum_{i=1}^n \mathcal{L}(x_i, y_i, \theta)$, and pseudolabels are generated on the intermediate model $\tilde{y}_i = f_{\theta_{intermediate}}(\tilde{x}_i)$. The final parameter is obtained by minimizing the adversarial loss function on two sets of samples:

$$\frac{1}{n} \sum_{i=1}^n \mathcal{L}_{robust}(x_i, y_i, \theta) + \lambda \cdot \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \mathcal{L}_{robust}(\theta, \tilde{x}_i, \tilde{y}_i), \quad (22)$$

where $\mathcal{L}_{robust}()$ is calculated as:

$$\mathcal{L}_{robust}(\theta, x, y) = \mathcal{L}_{standard}(\theta, x, y) + \lambda \cdot \max_{x' \in \mathcal{B}(x)} D_{KL}(p_\theta(\cdot|x) || p_\theta(\cdot|x')), \quad (23)$$

$D_{KL}()$ quantizes the distance between predictions of normal and adversarial samples using KL divergence. Similarly, Stanforth et al. [60] state that the adversarial error could be approximated through:

$$\mathcal{L}_{adv} = \mathcal{L}_{nat} + (\mathcal{L}_{adv} - \mathcal{L}_{nat}) \leq \mathcal{L}_{nat} + \mathbb{E}_{x \in \mathcal{D}} \sup_{x' \in \mathcal{B}(x)} D(f_\theta(x'), f_\theta(x)), \quad (24)$$

where $\mathcal{L}_{adv}$ and $\mathcal{L}_{nat}$ are prediction losses on adversarial examples and natural examples, and $D(f_\theta(x'), f_\theta(x))$ is the distance between predictions of x and x' under θ . The authors argue that minimizing the standard loss and smoothness loss could effectively enhance the robustness of classifiers against adversaries. Based on the approximation, the authors propose the unsupervised adversarial training (UAT) method, which minimizes the overall loss function of smoothness loss and standard loss for quantizing the adversarial generalization and standard accuracy $\mathcal{L}(\theta) = \mathcal{L}_{sup}(\theta) + \lambda \cdot \mathcal{L}_{unsup}(\theta)$. UAT is verified to attain equivalent robustness with 10 times fewer data compared with supervised adversarial training methods. The performance of adversarial training could be improved greatly by semi-supervised and unsupervised mechanisms. Unsupervised adversarial training could also further maintain regular accuracy better on standard samples.

4.5 Certified Robustness

Since exactly analyzing the properties of neural networks is difficult and the analysis methods are difficult to scale to in-depth DNN architectures [123], existing certified robustness methods mainly attempt to estimate provable robustness regions of DNN models. However, most of the estimated robust regions are nondifferential and are always impractical to reach through training. Hence,

state-of-the-art certified robustness works try to further approximate the tractable robust regions that could be optimized through training or regularizing specific constants of DNN architectures.

Adding specific regularizations to DNN structures is demonstrated to provide robustness. Ross and Doshi-Velez [116] argue that gradient regularization on input $\min_{\theta} H(y) = \sum_{i=1}^d (\frac{\partial}{\partial x_i} \sum_{j=1}^m -y_j \log f_j(x, \theta))^2$ could avoid oversized changes in predictions induced by small perturbations in the input vectors. Sinha et al. [105] propose an adversarial training procedure that could provide provable guarantees through the distributional robust optimization method. The authors propose a min-max procedure for enhancing the robustness of DNN against a region around data distribution $\mathcal{D}$ through $\min_{\theta} \sup_P \mathbb{E}_P[\mathcal{L}(\theta; \mathcal{D})]$, where P is a bounded region around the distribution of normal samples, and $\sup_P \mathbb{E}_P[\mathcal{L}(\theta; Z)]$ is the worst-case perturbation in the set of regions around the data distribution. For a smooth loss function $\mathcal{L}(\theta; X)$ and the ρ -bounded robust region $P = \{P : W_e(P, X) \leq \rho\}$ that is measured using the Wasserstein metric W_e . Under small perturbations, the authors demonstrate that the formal optimization problem can be solved by minimizing its Lagrangian relaxation, and the objective function is strongly concave and can be solved through the stochastic gradient method.

The cross-Lipschitz regularization of DNNs has also been proven to provide provable robustness [117]. The authors provide a lower bound on ℓ_p bounded perturbations to change the classifier prediction:

$$\|\eta\|_p \leq \max_{R>0} \min \left\{ \min_{x' \in \mathcal{B}_p(x, R)} \frac{f_c(x) - f_j(x)}{\|\nabla_k f(x') - \nabla f_j(x')\|_q}, R \right\} := \alpha, \quad (25)$$

where q should satisfy $\frac{1}{p} + \frac{1}{q} = 1$ and k is the chosen class under perturbation $\eta: k = \arg \max_j f_j(x + \eta)$. Obviously, the bound R increases as $f_c(x) - f_j(x)$ increases or $\max_{j \neq c} \max_{x' \in \mathcal{B}_p(x, R)} \|\nabla_k f(x') - \nabla f_j(x')\|_q$ decreases. Therefore, the robustness of DNN could be enhanced by minimizing the cross-Lipschitz regularization function $\Omega(f) = \frac{1}{nm^2} \sum_{i=1}^n (\sum_{l,k=1}^m \|\nabla f_l(x_i) - \nabla f_k(x_i)\|_2^2)$. This facilitates different decision functions at arbitrary samples to be more constant. Similarly, Cisse et al. [12] verify the effectiveness of defending perturbations by constraining the Lipschitz constant of network layers. The authors prove that under ℓ_p bounded adversarial example x' , the loss function under normal examples $\mathcal{L} = \mathbb{E}_{x \sim \mathcal{D}} \mathcal{L}(x + \eta, y)$ and adversarial examples $\mathcal{L}' = \mathbb{E}_{x \sim \mathcal{D}} \mathcal{L}(L(x', y))$ satisfy

$$\mathcal{L}' \leq \mathcal{L} + \mathbb{E}_{x \sim \mathcal{D}} [\mathcal{L}(f(x'), y) - \mathcal{L}(f(x), y)] \leq \mathcal{L}(W) + \lambda_p \Lambda_p \varepsilon, \quad (26)$$

where λ_p and Λ_p are the Lipschitz constants of the loss function and the classifier. Since the Lipschitz constant of the network layer could be amplified through propagation, they should be constrained to be smaller than 1. The Lipschitz constant constrained networks are called the *Parseval networks*, and they are proven to be intrinsically robust to any bounded perturbations. Tsuzuku et al. [118] argue that the calculation procedure of Lipschitz constraint methods is overcomplicated on large networks and further propose a more convenient constraint method called *Lipschitz-Margin training*, which is based on the relation between the Lipschitz constant and prediction margins.

Although Lipschitz regularization methods are theoretically verified to provide formal robustness, the calculation procedure of those methods is still complex, and the bound of the provable robustness region is loose [113]. Instead, gradient-based certification methods that provide tractable bounds of provable robustness against bounded perturbations are more efficient to implement. Wong and Kolter [106] propose a mechanism for minimizing the convex out bound of the adversarial polytope, which denotes the allowed perturbations that could not change the prediction of classifiers after multi ReLU-based layers using the dual method. Raghunathan et al. [107] further present the upper bound of losses under the worst-case perturbations for neural networks using

the dual method. The upper bound is differential and, thus, could be trained for enhanced robustness. The trainable lower bounds of the provable robustness region are also demonstrated to show effective performance on simple networks. Several feasible certification methods have been proposed for a more efficient mechanism to extend the implementation of certification methods on larger nonlinear network architectures [109–111]. Moreover, Zhai et al. [114] propose an attack-free certified robust training method called *maximizing the CERTified radius (MACER)* that could provide robustness against any ℓ_2 normed attacks accompanied by faster convex time. Croce and Hein [115] further present a novel regularization method that could provide provable robustness for any ℓ_p bounded attacks on ReLU-based networks. Commonly used adversarial training methods trained with ℓ_∞ bounded attacks were later demonstrated to show little performance on other ℓ_1 or ℓ_2 bounded attacks [124, 125]. State-of-the-art certified robustness methods are trying to provide provable robustness against any bounded attacks.

Another line of work approximates robustness by resorting to a smoothing mechanism. The smoothing mechanism argues that approximated certified robustness against ℓ_2 or ℓ_1 bounded attacks could be attained through training against random noise [112, 126]. Cohen et al. [113] propose a randomized smoothing method that could provide a tight robustness guarantee against ℓ_2 bounded attacks. By constructing a novel classifier $g(x) = \arg \max_k \mathbb{P}(f(x + \epsilon) = k)$, $\epsilon \in \mathcal{N}(0, \sigma^2 I)$, where $\mathbb{P}(\cdot)$ is the prediction probability of each class, the authors verify that under any classifier $f(\cdot)$, $g(\cdot)$ could be guaranteed to be robust to $\|\eta\|_2 \leq R$ attacks. Since R is in direct proportion to the variance of Gaussian noise, the robustness of $g(\cdot)$ could be attained by training $f(\cdot)$ with Gaussian noise added to normal data. Compared with approximating the provable robustness under bounded attacks, smoothing performances are much easier to implement and could be employed on ImageNet robustness training. However, existing smoothing works only show effectiveness against ℓ_2 or ℓ_1 bounded attacks.

In addition, the certified robustness method could be implemented by analyzing the trade-off between standard and adversarial accuracy. Zhang et al. [40] theoretically analyze the trade-off between accuracy and robustness by introducing the boundary error for the binary classifier f : $R_{bdy}(f) = \mathbb{E}_{x \sim \mathcal{D}} 1\{x \in \mathcal{B}(DB(f), \epsilon), f(x)y > 0\}$, where $DB(f)$ is the decision boundary of f . $\mathcal{B}(DB(f), \epsilon)$ represents the neighbor of $DB(f)$: $\{x, \exists x' \in \mathcal{B}(x, \epsilon), s.t. f(x)f(x') \leq 0\}$. The standard error $R_{nat}(f) = \mathbb{E}_{(x,y) \sim \mathcal{D}} 1\{f(x)y \leq 0\}$ and robust error $R_{rob}(f) = \mathbb{E}_{(x,y) \sim \mathcal{D}} 1\{\exists x' \in \mathcal{B}(x, \epsilon), s.t. f(x')y \leq 0\}$ can be denoted as $R_{bdy}(f) = R_{rob}(f) - R_{nat}(f)$. Through theoretical approximation, the authors infer that the upper and lower bounds of $R_{rob}(f) - R_{nat}^*(f)$ highly rely on the distance between predictions of normal samples and boundary examples, where $R_{nat}^*(f)$ denotes the optimal natural error. Based on the aforementioned analysis, the authors propose trade-off-inspired adversarial defense via the surrogate-loss minimization (TRADES) method:

$$\min_f \mathbb{E}_{(x,y) \sim \mathcal{D}} \left\{ \lambda \cdot \mathcal{L}(f(x), y) + \max_{x' \in \mathcal{B}(x, \epsilon)} \mathcal{L}(f(x), f(x')) \right\}, \quad (27)$$

where $\mathcal{L}(f(x), f(x'))$ is the distance between predictions of models under normal and adversarial samples.

4.6 Performance Evaluation and Complex Issue

In the previous sections, the performance of state-of-the-art perturbation detection and stochastic defense methods are presented with TPR, FRP, and DES. In this section, we illustrate the performance of other categories of defense methods that could be evaluated on the same platform. A more detailed analysis of existing defense methods is also presented. The complexity issue of different categories of defense methods is presented in this section as well.

Table 8. Performance of the State-of-the-Art Input Transformation, Adversarial Training and Certified Robustness Methods

Defense Method	Category	Dataset	Threat Model	PGD	CW	Natural Accuracy
Natural Training	-	CIFAR10	WRN	0	0	95.9
FS [71]	IT	CIFAR10	DenseNet	56	86	89.3
PD [83]	IT	CIFAR10	ResNet	46	78	85
AT +PD [83]	IT	CIFAR10	ResNet	69	84	88
AT [16]	AT	CIFAR10	WRN	52.7	50.7	87.3
MART [100]	AT	CIFAR10	WRN	55.5	54.8	83.8
BAT [122]	AT	CIFAR10	WRN	57.5	56.2	91.0
Cat [99]	AT	CIFAR10	WRN	73.2	71.7	89.6
UAT++ [60]	AT	CIFAR10@Tiny Image 80 m	WRN	62.2	-	85.9
RST [59]	AT	CIFAR10@Tiny Image 80 m	WRN	63.1	64.9	89.7
TRADES [40]	CR	CIFAR10	WRN	56.4	81.2	84.9
SP [111]	CR	CIFAR10	WRN	23.5	-	27.1

4.6.1 Performance Evaluation. Table 8 summarizes the performance of the state-of-the-art defense methods under PGD and CW attacks. PGD and CW are all ℓ_∞ bounded attacks of 8/255 magnitude since ℓ_∞ is the most commonly used norm to restrict attacks, which indicates the amplitude peak of manipulation on each pixel of images. We observed that input transformation methods alone could offer considerable robustness and assist in boosting the performance of adversarial training. In addition, most of the influential works against ℓ_∞ bounded attacks proposed recently are adversarial training methods. Adversarial training methods have been verified to be the most effective defense methods since they can provide effective robustness enhancement instead of increasing the difficulty of generating attacks. Hence, they are difficult to defeat by other later proposed stronger attacks. The scalable certified robustness method SP [111], which provides a provable robust region, shows limited performance on strong PGD attacks since the approximated provable region is still loose, especially on in-depth DNN models. TRADES [40] provides efficient robustness, yet it is still trained through adversarial training.

While the progress of state-of-the-art attacks and defenses has been very rapid during the past several years, there is still no universal criterion for fairly evaluating all categories of defense methods. State-of-the-art works mainly focus on presenting more powerful methods belonging to the same category. Some exclusive measurements are still requested for comparing disparate categories of defense methods. For instance, Wang et al. [94] present a stochastic method that acquires equivalent robustness with less standard accuracy declination compared with adversarial training [16]. As analyzed earlier, an effective evaluation framework may be the robust performance against attacks under the minimum required regular accuracy, which is a potential theory to be solved in the future.

4.6.2 Complexity Issue. The complexity issue of defense methods was introduced at the beginning of Section 4. Specifically, for a model-independent defense method without retraining, the computation complexity could be denoted as $O(N * d_1)$, where N and d_1 are the total number of input samples and the dimension of defense architecture. For a model-dependent method that needs retraining, the computational complexity changes to $O(N * d)$, where d is the dimension of the DNN. Since the dimension of the DNN is often much larger than that of independent detectors, these defense methods may take much more time and computing resources for robustness.

Adversarial training methods need to calculate the worst-case perturbation at each iteration. In most cases, finding the worst-case perturbations requires iteratively querying models for directions, and the computational complexity of adversarial training becomes $O(n * N * d)$, where n is the total number of steps of each iteration for the required perturbations. Therefore, the training procedure is usually several times longer than ordinary training [25]. Researchers have recently proposed several novel training mechanisms for higher training efficiency [104, 127], yet the training procedure is still complex, especially on large datasets. State-of-the-art adversarial training methods cannot avoid standard performance declination [40]. Therefore, adversarial training is usually thought to be a mechanism that sacrifices training efficiency and standard accuracy for robustness.

Certified robustness methods try to exactly verify the vulnerability of fully connected networks against any manipulation or check whether a sample could change the final decision of neural network-based models. Considering a nonlinear fully connected network model with k hidden layers $f(x) = \mathcal{A}_{k-1}(\phi^{(k-1)}(\mathcal{A}_{k-2}(\phi^{(k-2)}(\dots(\mathcal{A}_0(\phi^{(0)}(x))))))$ and nonlinear activation $\mathcal{A}_i(\cdot)$, a provable robust DNN model against any manipulation Δ should satisfy

$$\nabla x \in \mathbb{R}^d, \text{argmax } f(x) = \text{argmax } f(x + \Delta). \quad (28)$$

The most intuitive certified robustness method is to precisely check whether a sample is in a provable region that could not change the ultimate decisions. Obtaining a region of input samples requires backpropagation of misclassification on output layers and other hidden layers to the input layer. For the l -th hidden layer $\phi^{(l)}$ and a manipulation r_l added on the hidden layer, the manipulated activations on the hidden layers $\mathcal{A}_l(\cdot)$ should first propagate through feed-forward layers to verify whether the predictions $\mathcal{A}_{k-1}(\cdot)$ are changed and backpropagate the information to the input layer for approximating input regions. Generally, the dimension of hidden layers is large, and refinements on specific layers may have an unpredictable influence on previous layers. Moreover, the exact feedback of manipulated layers on activations to the previous layer $\mathcal{A}_l(\cdot) \rightarrow \phi^{(l-1)}(\cdot)$ is impractical to acquire under nonlinear activation functions. Huang et al. [128] first presented several theorems for simplifying the whole procedure to reduce the feature dimension of each layer. The independence between adjacent layers, which promotes the robustness of the DNN architecture, could be checked layer by layer. A satisfactory modulus theory (SMT)-based solver is also proposed to encode the activation function from a layer to the previous layer. Several effective solver-based methods have also been proposed for verifying the properties of fully connected networks using SMT [123] or mixed integer linear programming (LP) [129]. Compared with gradient-based adversarial training methods, although verifying the robustness of fully connected neural networks could effectively provide provable robustness, analyzing the commonly used ReLU-based networks exactly is thought to be NP-hard and time-consuming even on small networks with limited parameters. Hence, the analysis mechanisms are even difficult to scale to large DNNs [123]. Furthermore, for a fully connected network architecture that has completely independent layers and linear activations, the computational complexity of acquiring a provable region against PGD attacks added on hidden layers could be $O(k * n * N * d)$, which is even more complex than min-max method-based adversarial training methods.

While the constructions of adversarial training and certified robustness methods are thought to be overly complex and time-consuming, existing perturbation detection, input transformation, and stochastic defense methods are later defeated by adaptive attacks, as shown in Section 5.1. Adaptive attacks are some perturbations that have knowledge about defense methods and could adapt to changed architecture. Therefore, while those defense mechanisms could effectively offer robustness against existing attacks, they still suffer from the risks of more powerful bounded

attacks. This phenomenon has promoted adversarial training and certified robustness methods to gain the most attention in the past few years.

4.7 Applications of Defense Methods

Combined with the application of adversarial perturbations, recently corresponding defense methods have also performed well in building robust models on security-required scenarios. First, several works have proposed robust models on vision-based issues to avoid potential attacks added to raw images. Zhou et al. [130] apply the variational autoencoder-based defense method to the face recognition model and validate the effectiveness of robust face recognition models against existing adversarial perturbations. Zhou et al. [131] present an adversarial defense method to resist 3D point cloud adversarial examples in 3D computer vision tasks. Jia et al. [132] propose a novel method to resist adversarial videos. Xu et al. [133] designed and demonstrated the performance of a novel adversarial training method on semantic segmentation. Li et al. [134] present a novel defense framework to enhance the robustness of object detection models. Bai et al. [135] propose a defense network to defend against adversarial attacks designed on metric-based applications and validate its effectiveness on person reidentification.

Adversarial defense methods are also applied to real-world applications, including malware detection [136], audio recognition [137], recommendation systems [138], and medical artificial intelligence systems [139]. Finally, while commonly used adversarial training methods are verified to enhance the robustness of image-based DNN models with inevitable standard performance decline, recent works find a surprising phenomenon in which they could enhance both the performance of generalization and robustness on NLP tasks [140].

5 DISCUSSIONS AND RESEARCH DIRECTIONS

In previous sections, we presented a comprehensive review of existing defense methods and basic hypotheses on the origins of adversarial perturbations. With the rapid progress of adversarial perturbations and counter defense methods, researchers have gained a deeper understanding of the vulnerability of DNN structures and the mechanism of adversarial perturbations. This has shed light on designing more powerful and effective robust models. In contrast, many existing defense methods were later verified to show limited performance on more powerful attacks. In this section, we discuss some key limitations, challenges, and characteristics of potential research directions on adversarial defense methods, including the challenge caused by the gradient mask character, the controversial topic on the origin of adversarial perturbations, the limitations of adversarial training and certified robustness methods, and the unexpected yet attractive attribute of the adversarial training methods.

5.1 The Challenge of Gradient Mask-Based Defense Methods

The gradient mask denotes a phenomenon in which models trained for robustness essentially converge to a degenerate global minimum of the loss function. Since a curvature of decision surfaces near data points obfuscates the loss function, generated adversarial perturbations are not the worst-case attacks, and models trained based on those perturbations show only limited robust performance [98]. Athalye et al. [32] verify that some existing defense methods that have shown powerful defense performance against existing adversarial perturbations are actually puzzled by the obfuscated gradient phenomenon. An obfuscated gradient is a specific type of gradient mask including three cases called *shattered gradients*, *stochastic gradients*, and *exploding and vanishing gradients*. Shattered gradients are caused by nondifferential defense functions, such as some input transformation methods that could hinder the calculation of the worst-case perturbation. The authors thus present the BPDA method for replacing nondifferential intermediate layer $\phi^{(i)}(\cdot)$ with

differential substitute $g(x) \approx \phi^{(i)}(x)$ to perform backpropagation. Stochastic gradients are caused by random settings added to the inputs or inner structures of DNN models. Gradients generated on randomized samples will not provide an accurate gradient on the single sample. The authors argue that expectation over transformation (EOT), which optimizes the expectation of transformation T from samples $t(x)$ and the gradients over expectation, could be approximately estimated by $\nabla \mathbb{E}_{t \sim T} f(t(x)) = \mathbb{E}_{t \sim T} \nabla f(t(x))$. Transformations $g(\cdot)$ that could induce gradient explosion or vanishing could be replaced with differential function $h(z) = x$ that satisfies $g(h(z)) = h(z)$. By calculating the gradients of models suffering from shattered gradients, the authors find that most of the defense methods published on ICLR2018 that belong to input transformation [15, 78, 83], stochastic [79, 89, 92], and perturbation detection [74] methods could be circumvented effectively. Models trained using the adversarial training method is the exclusive method that is verified to show authentic robustness [16].

Although many existing studies have declared the ability to defend strong attacks effectively, some are actually obfuscated by shattered gradients, which could lead to incorrect calculation of worst-case perturbations. Hence, the test accuracy under perturbations may not provide a reliable evaluation of defense performance. Athalye et al. [32] argue that adaptive attacks should be considered for evaluating the performance of defense methods in future works. Adaptive attacks are specific attacks that have full access to defense mechanisms and could adapt themselves to circumvent defenses. Some more accurate and direct evaluation criteria are also requested. For instance, the similarity between manifolds of normal data and gradients of the loss function on adversarial examples may also act as an auxiliary evaluation of defense methods [57].

5.2 The Exact Origin of Adversarial Perturbations

Interpretations of the existence of adversarial perturbations have shown great value in designing more powerful defense methods and explaining the mechanism of the effectiveness of adversarial training. For example, based on the analysis that adversarial perturbations are caused by insufficient data [10], Carmon et al. [59] and Stanforth et al. [60] present semi-supervised adversarial training methods that could take unlabeled data from other datasets to increase the accuracy of both adversarial examples and normal samples. Based on the phenomenon of nonrobust features [18] and boundary tilting [11], Zhang and Wang [104] present an unsupervised adversarial example-generating method for adversarial training to avoid relying too heavily on nonrobust features and deviating decision boundaries from the manifold of normal samples. Thermometer encoding is designed to overcome overlinearity [89]. Dimension reduction methods such as feature squeezing [71] have also been demonstrated to reduce the influence of adversarial perturbations. Although we have introduced some basic hypotheses on the origin of adversarial perturbations in this article, the existence of adversarial perturbations, compared with valid interpretations of the standard accuracy declination on adversarial training, is still a controversial problem to be solved.

5.3 Limitations of Adversarial Training

Adversarial training methods have been verified to show effective performance against strong attacks. However, there are still some controversies regarding adversarial training. First, the robustness accuracy against attacks is relatively low compared with standard accuracy. For example, the recognition accuracy of adversarial training [16] trained under PGD¹⁰ has only a 47% accuracy on perturbed test samples on CIFAR10. Second, well-trained adversarial training models inevitably suffer from a decline in standard accuracy on normal samples [25, 28], which can cause the standard accuracy of well-trained Wide-ResNet34 on CIFAR10 to drop from 95 to 87.

Fig. 4. The gradients of the loss function on input features in models trained on different datasets with diverse attacks or no attacks [25].

5.4 Unexpected Prospect of Adversarial Training

The original intention of adversarial training [16] is to force DNN models to recognize the segmentation of samples effectively under worst-case perturbations to defend against other potential attacks. Unexpectedly, Nakkiran et al. [25] find that after adversarial training, the gradient of loss functions on input features aligns better with human perception features compared with the gradients of standard training, as shown in Figure 4. This gives a reasonable interpretation that small adversarial perturbations that are imperceptible to human eyes could disturb DNN models. This phenomenon may also offer an interpretable design methodology for DNN-based computer vision issues such as object recognition or reinforcement learning to urge agents to recognize the “real” object for better generalization. In addition, alignment effectiveness could also be referred to as a verification of whether a defense method shows authentic robustness.

5.5 Poor Scalable Character of Certified Robustness

The certified robustness defense method is the most direct method for analyzing the robustness of neural networks or providing provable robustness. Certified robustness methods could also offer ℓ_p bounded robustness that could not be attained by adversarial training. Since the structure of a DNN is difficult to interpret, the exact provable bound of a DNN under specific attacks is thought to be unreachable. Existing studies have provided several works on estimating approximated bounds that could be optimized. Most of those studies on certified robustness are restricted to small network structures and datasets. While there are already some studies focusing on scaling certified robustness methods to DNNs [111], as shown in Table 8, the performances of these methods are still unable to reach estimated bounds. Bubeck et al. [17] argue that computing constraints used for DNN simplicity may inevitably produce adversarial perturbations. Therefore, analyzing the robustness by interpreting DNN models and developing robust mechanisms for training DNN architectures are highly promising.

6 CONCLUSIONS

In this article, we have comprehensively reviewed classic and state-of-the-art defense methods against adversarial perturbations and elaborated the basic hypotheses on the origin of adversarial perturbations. We also give performances of the state-of-the-art defense works that belong to different categories. Defending against strong attacks and exploring the vulnerability of DNNs

can help significantly improve the security of DNN models, thus extending the application of DNN models to real-world high-security issues. This field is highly promising and active, and a large number of significant algorithms and methods have been proposed. This article focuses on reviewing the most influential or recently proposed corresponding studies.

ACKNOWLEDGEMENTS

We appreciate four anonymous reviewers for their valuable and insightful comments. We thank the Editor-in-Chief and Associate Editor for their support of this paper.

REFERENCES

- [1] L. Deng and D. Yu. 2014. Deep learning: Methods and applications. *Foundations and Trends in Signal Processing* 7, 3–4 (2014), 197–387.
- [2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. 2013. Intriguing properties of neural networks. Retrieved June 25, 2021 from <https://arXiv:1312.6199>.
- [3] X. Yuan, P. He, Q. Zhu, and X. Li. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems* 30, 9 (2019), 2805–2824.
- [4] N. Papernot, P. McDaniel, and I. Goodfellow. 2016. Transferability in machine learning: From phenomena to black-box attacks using adversarial samples. Retrieved June 25, 2021 from <https://arXiv:1605.07277.s>.
- [5] Y. Liu, X. Chen, C. Liu, and D. Song. 2016. Delving into transferable adversarial examples and black-box attacks. Retrieved June 25, 2021 from <https://arXiv:1611.02770>.
- [6] D. Meng and H. Chen. 2017. Magnet: A two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Dallas, TX USA, 135–147.
- [7] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner. 2017. Detecting adversarial samples from artifacts. Retrieved June 25, 2021 from <https://arXiv:1703.00410>.
- [8] I. Goodfellow, J. Shlens, and C. Szegedy. 2014. Explaining and harnessing adversarial examples. Retrieved June 25, 2021 from <https://arXiv:1412.6572>.
- [9] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii. 2018. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 8 (2018) 1979–1993.
- [10] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry. 2018. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems*. MIT Press, Palais des Congrès de Montréal, Montréal Canada, 5019–5031.
- [11] T. Tanay and L. Griffin. 2016. A boundary tilting perspective on the phenomenon of adversarial examples. Retrieved June 25, 2021 from <https://arXiv:1608.07690>.
- [12] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier. 2017. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning*. ACM, Sydney, NSW Australia, 854–863.
- [13] D. Gopinath, G. Katz, C. S. Pasareanu, and C. Barrett. 2017. DeepSAFE: A data-driven approach for checking adversarial robustness in neural networks. Retrieved June 25, 2021 from <https://arXiv:1710.00486>.
- [14] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP'16)*. IEEE, San Jose, CA USA, 582–597.
- [15] P. Samangouei, M. Kabkab, and R. Chellappa. 2018. Defense-gan: Protecting classifiers against adversarial attacks using generative models. Retrieved June 25, 2021 from <https://arXiv:1805.06605>.
- [16] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. 2017. Towards deep learning models resistant to adversarial attacks. Retrieved June 25, 2021 from <https://arXiv:1706.06083>.
- [17] S. Bubeck, E. Price and I. Razenshteyn. 2018. Adversarial examples from computational constraints. Retrieved from <https://arXiv:1805.10204>.
- [18] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. 2019. Adversarial examples are not bugs, they are features. Retrieved June 25, 2021 from <https://arXiv:1905.02175>.
- [19] N. Akhtar and A. Mian. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* 6, (2018), 14410–14430.
- [20] M. Ozdag. 2018. Adversarial attacks and defenses against deep neural networks: A survey. *Procedia Computer Science* 140 (2018), 152–161.
- [21] D. J. Miller, Z. Xiang, and G. Kesidis. 2019. Adversarial learning in statistical classification: A comprehensive review of defenses against attacks. Retrieved June 25, 2021 from <https://arXiv:1904.06292>.

- [22] G. E. Hinton, S. Osindero, and Y.-W. The. 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18, 7 (2006), 1527–1554.
- [23] Y. LeCun, Y. Bengio, and G. Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.
- [24] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. 2017. Robust physical-world attacks on deep learning models. Retrieved June 25, 2021 from <https://arXiv:1707.08945>.
- [25] P. Nakkiran. 2019. Adversarial robustness may be at odds with simplicity. Retrieved June 25, 2021 from <https://arXiv:1901.00532>.
- [26] K. Chen, H. Zhu, L. Yan, and J. Wang. 2020. A survey on adversarial examples in deep learning. *Journal on Big Data* 2, 2 (2020), 71–84.
- [27] H. Kannan, A. Kurakin, and I. Goodfellow. 2018. Adversarial logit pairing. Retrieved June 25, 2021 from <https://arXiv:1803.06373>.
- [28] A. Kurakin, I. Goodfellow, and S. Bengio. 2016. Adversarial machine learning at scale. Retrieved June 25, 2021 from <https://arXiv:1611.01236>.
- [29] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. 2016. Deepfool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, New York, NY USA, 2574–2582.
- [30] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. 2016. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P'16)*. IEEE, Saarbrücken, Saarland Germany, 372–387.
- [31] N. Carlini and D. Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP'17)*. IEEE, SAN Jose, CA USA, 39–57.
- [32] A. Athalye, N. Carlini, and D. Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*. ACM, Stockholm, Sweden, 274–283.
- [33] P. Zhao, S. Liu, Y. Wang, and X. Lin. 2018. An ADMM-based universal framework for adversarial attacks on deep neural networks. In *Proceedings of the 26th ACM International Conference on Multimedia*. ACM, San Servolo Island, Venice Italy, 1065–1073.
- [34] K. Xu, S. Liu, P. Zhao, P.-Y. Chen, H. Zhang, Q. Fan, D. Erdogmus, Y. Wang, and X. Lin. 2018. Structured adversarial attack: Towards general implementation and better interpretability. OpenReview.net, Millennium Hall, Addis Ababa Ethiopia. Retrieved June 25, 2021 from <https://arXiv:1808.01664>.
- [35] B. Wang, F. Zou, and X. Liu. 2020. New algorithm to generate the adversarial example of image. *Optik* 207 (April 2020), 164477. DOI : <https://doi.org/10.1016/j.ijleo.2020.164477.%3c/bib>
- [36] M. Cheng, S. Singh, P.-Y. Chen, S. Liu, and C. Hsieh. 2020. Sign-opt: A query-efficient hard-label adversarial attack. In *International Conference on Learning Representations*. Retrieved June 25, 2021 from <https://openreview.net/forum?id=SkltQCNtvs>.
- [37] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin. 2018. Black-box adversarial attacks with limited queries and information. Retrieved June 25, 2021 from <https://arXiv:1804.08598>.
- [38] S. Cheng, Y. Dong, T. Pang, H. Su, and J. Zhu. 2019. Improving black-box adversarial attacks with a transfer-based prior. In *Advances in Neural Information Processing Systems*. MIT Press, Vancouver, B.C Canada, 10934–10944.
- [39] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin. 2019. On evaluating adversarial robustness. Retrieved June 25, 2021 from <https://arXiv:1902.06705>.
- [40] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. 2019. Theoretically principled trade-off between robustness and accuracy. Retrieved June 25, 2021 from <https://arXiv:1901.08573>.
- [41] P. Zhao, P.-Y. Chen, S. Wang, and X. Lin. 2020. Towards query-efficient black-box adversary with zeroth-order natural gradient descent. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, New York, NY USA, 6909–6916.
- [42] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh. 2017. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, Dallas, TX USA, 15–26.
- [43] A. Ilyas, L. Engstrom, and A. Madry. 2018. Prior convictions: Black-box adversarial attacks with bandits and priors. Retrieved June 25, 2021 from <https://arXiv:1807.07978>.
- [44] N. Narodytska and S. P. Kasiviswanathan. 2016. Simple black-box adversarial perturbations for deep networks. Retrieved June 25, 2021 from <https://arXiv:1612.06299>.
- [45] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security*. ACM, New York, NY USA, 506–519.

- [46] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. 2017. The space of transferable adversarial examples. Retrieved June 25, 2021 from <https://arXiv:1704.03453>.
- [47] Y. Luo, X. Boix, G. Roig, T. Poggio, and Q. Zhao. 2015. Foveation-based mechanisms alleviate adversarial examples. Retrieved June 25, 2021 from <https://arXiv:1511.06292>.
- [48] A. Fawzi, O. Fawzi, and P. Frossard. 2018. Analysis of classifiers' robustness to adversarial perturbations. *Machine Learning* 107, 3 (2018), 481–508.
- [49] J. Gilmer, L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, and I. Goodfellow. 2018. The relationship between high-dimensional geometry and adversarial examples. Retrieved June 25, 2021 from <https://arXiv:1801.02774>.
- [50] A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein. 2018. Are adversarial examples inevitable? Retrieved June 25, 2021 from <https://arXiv:1809.02104>.
- [51] A. Fawzi, H. Fawzi, and O. Fawzi. 2018. Adversarial vulnerability for any classifier. In *Advances in Neural Information Processing Systems*. MIT Press, Vancouver, B.C Canada, 1186–1195.
- [52] S. Mahloujifar, D. I. Diochnos, and M. Mahmoody. 2019. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, Hawaii, HI USA, 4536–4543.
- [53] A. Fawzi, S. M. Moosavi-Dezfooli, and P. Frossard. 2016. Robustness of classifiers: From adversarial to random noise. Retrieved June 25, 2021 from <https://arXiv:1608.08967>.
- [54] J. Y. Franceschi, A. Fawzi, and O. Fawzi. 2018. Robustness of classifiers to uniform lp and Gaussian noise. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*. AAAI, Beijing, China, 1280–1288.
- [55] N. Ford, J. Gilmer, N. Carlini and D. Cubuk. 2019. Adversarial examples are a natural consequence of test error in noise. Retrieved June 25, 2021 from <https://arXiv:1901.10513>.
- [56] Y. Dong, H. Su, J. Zhu, and F. Bao. 2017. Towards interpretable deep neural networks by leveraging adversarial examples. Retrieved June 25, 2021 from <https://arXiv:1708.05493>.
- [57] B. Kim, J. Seo and T. Jeon. 2019. Bridging adversarial robustness and gradient interpretability. Retrieved June 25, 2021 from <https://arXiv:1903.11626>.
- [58] D. Hendrycks, D., K. Lee, and M. Mazeika. 2019. Using pre-training can improve model robustness and uncertainty. In *Proceedings of the 36th International Conference on Machine Learning*. ACM, Long Beach, CA USA, 97, 2712–2721.
- [59] Y. Carmon, A. Raghunathan, L. Schmidt, P. Liang, and J. C. Duchi. 2019. Unlabeled data improves adversarial robustness. Retrieved June 25, 2021 from <https://arXiv:1905.13736>.
- [60] R. Stanforth, A. Fawzi, and P. Kohli. 2019. Are labels required for improving adversarial robustness? Retrieved June 25, 2021 from <https://arXiv:1905.13725>.
- [61] C. Mao, A. Gupta, V. Nitin, B. Ray, S. Song, J. Yang, and C. Vondrick. 2020. Multitask learning strengthens adversarial robustness. In *Proceedings of the European Conference on Computer Vision*. IEEE, Edinburgh, UK, 158–174.
- [62] B. Li, C. Chen, W. Wang, and L. Carin. 2019. Certified adversarial robustness with additive noise. In *Advances in Neural Information Processing Systems*. MIT Press, Vancouver, B.C Canada, 9459–9469.
- [63] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao. 2018. Is robustness the cost of accuracy?—A comprehensive study on the robustness of 18 deep image classification models. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*. IEEE, Munich, Free State of Bavaria Germany, 631–648.
- [64] P. W. Koh and P. Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*. ACM, Sydney, NSW Australia, 70, 1885–1894.
- [65] Z. Gong, W. Wang, and W.-S. Ku. 2017. Adversarial and clean data are not twins. Retrieved June 25, 2021 from <https://arXiv:1704.04960>.
- [66] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel. 2017. On the (statistical) detection of adversarial examples. Retrieved June 25, 2021 from <https://arXiv:1702.06280>.
- [67] X. Li and F. Li. 2017. Adversarial examples detection in deep networks with convolutional filter statistics. In *Proceedings of the IEEE International Conference on Computer Vision*. 5764–5772.
- [68] J. Lu, T. Issaranon, and D. Forsyth. 2017. Safetynet: Detecting and rejecting adversarial examples robustly. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, Venice, Italy, 446–454.
- [69] D. Hendrycks and K. Gimpel. 2016. Early methods for detecting adversarial images. Retrieved June 25, 2021 from <https://arXiv:1608.00530>.
- [70] A. N. Bhagoji, D. Cullina, C. Sitawarin, and P. Mittal. 2018. Enhancing Robustness of Machine Learning Systems via Data Transformations. In *52nd Annual Conference on Information Sciences and Systems (CISS'18)*. IEEE, Princeton, NJ USA, 1–5.
- [71] W. Xu, D. Evans, and Y. Qi. 2017. Feature squeezing: Detecting adversarial examples in deep neural networks. Retrieved June 25, 2021 from <https://arXiv:1704.01155>.
- [72] N. Carlini and D. Wagner. 2017. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, Dallas, TX USA, 3–14.

- [73] K. Lee, K. Lee, H. Lee and J. Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*. MIT Press, Vancouver, B.C Canada, 7167–7177.
- [74] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, D. Song, M. E. Houle, and J. Bailey. 2018. Characterizing adversarial subspaces using local intrinsic dimensionality. Retrieved June 25, 2021 from <https://arXiv:1801.02613>.
- [75] J. Wang, G. Dong, J. Sun, X. Wang, and P. Zhang. 2019. Adversarial sample detection for deep neural network through model mutation testing. In *Proceedings of the 41st International Conference on Software Engineering*. IEEE, Montreal, QC Canada, 1245–1256.
- [76] S. Ma, Y. Liu, G. Tao, W.-C. Lee and X. Zhang. 2019. NIC: Detecting Adversarial Samples with Neural Network Invariant Checking. In *Proceedings of the 26th Network and Distributed Systems Security*. OpenReview.net, San Diego, CA USA. Retrieved from <https://openreview.net/forum?id=D54VV3ic4L>.
- [77] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy. 2016. A study of the effect of JPG compression on adversarial images. Retrieved June 25, 2021 from <https://arXiv:1608.00853>.
- [78] C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten. 2017. Countering adversarial images using input transformations. Retrieved June 25, 2021 from <https://arXiv:1711.00117>.
- [79] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. 2017. Mitigating adversarial effects through randomization. Retrieved June 25, 2021 from <https://arXiv:1711.01991>.
- [80] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer. 2018. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, UT USA, 8571–8580.
- [81] C. Kou, H. K. Lee, E.-C. Chang, and T. K. Ng. 2020. Enhancing transformation-based defenses against adversarial attacks with a distribution classifier. In *International Conference on Learning Representations*. OpenReview.net, Millennium Hall, Addis Ababa Ethiopia, 8571–8580.
- [82] S. Gu and L. Rigazio. 2014. Towards deep neural network architectures robust to adversarial examples. Retrieved June 25, 2021 from <https://arXiv:1412.5068>.
- [83] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. 2017. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. Retrieved June 25, 2021 from <https://arXiv:1710.10766>.
- [84] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu. 2018. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, UT USA, 1778–1787.
- [85] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He. 2019. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Long Beach, CA USA, 501–509.
- [86] X. Cao and N. Z. Gong. 2018. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference (ACSAC'18)*. ACM, San Juan, PR USA, 278–287.
- [87] V. Zantedeschi, M. I. Nicolae, and A. Rawat. 2017. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, Dallas, TX USA, 39–49.
- [88] T. Pang, C. Du, Y. Dong, and J. Zhu. 2018. Towards robust detection of adversarial examples. In *Advances in Neural Information Processing Systems*. Montreal, A.B Canada, 4584–4594.
- [89] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow. 2018. Thermometer encoding: One hot way to resist adversarial examples. In *6th International Conference on Learning Representations*. OpenReview.net, Vancouver, B.C Canada. Retrieved June 25, 2021 from <https://openreview.net/forum?id=S18Su-CW>.
- [90] A. Mustafa, S. H. Khan, M. Hayat, R. Goecke, J. Shen, and L. Shao. 2020. Deeply supervised discriminative learning for adversarial defense. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. IEEE. 1–1. DOI : 10.1109/tpami.2020.2978474
- [91] X. Liu, M. Cheng, H. Zhang, and C.-J. Hsieh. 2018. Towards robust neural networks via random self-ensemble. In *15th Proceedings of the European Conference on Computer Vision (ECCV'18)*. IEEE, Munich, Free State of Bavaria Germany, 369–385.
- [92] G. S. Dhillon, K. Azizzadenesheli, Z. C. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, and A. Anandkumar. 2018. Stochastic activation pruning for robust adversarial defense. Retrieved June 25, 2021 from <https://arXiv:1803.01442>.
- [93] S. Wang, X. Wang, P. Zhao, W. Wen, D. Kaeli, P. Chin, and X. Lin. 2018. Defensive dropout for hardening deep neural networks under adversarial attacks. In *Proceedings of the International Conference on Computer-Aided Design*. ACM, San Diego, CA USA, 71:1–8.
- [94] X. Wang, S. Wang, P.-Y. Chen, Y. Wang, and S. Chin. 2019. Protecting neural networks with hierarchical random switching: Towards better robustness-accuracy trade-off for stochastic defenses. In *28th International Joint Conference on Artificial Intelligence (IJCAI'19)*. Morgan Kaufmann, Macao, China, 6013–6019.

- [95] A. Bui, T. Le, H. Zhao, P. Montague, O. deVel, T. Abraham, and D. Phung. 2020. Improving adversarial robustness by enforcing local and global compactness. In *Proceedings of the European Conference on Computer Vision. ECCV*, Edinburgh, UK, 209–223.
- [96] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvari. 2016. Learning with a strong adversary. Retrieved June 25, 2021 from <https://arXiv:1511.03034>.
- [97] U. Shaham, Y. Yamada, and S. Negahban. 2018. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. *Neurocomputing* 307 (2018), 195–204.
- [98] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. 2017. Ensemble adversarial training: Attacks and defenses. Retrieved June 25, 2021 from <https://arXiv:1705.07204>.
- [99] M. Cheng, Q. Lei, P.-Y. Chen, I. Dhillon, and C. Hsieh. 2020. CAT: Customized adversarial training for improved robustness. Retrieved June 25, 2021 from <https://arXiv:2002.06789>.
- [100] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu. 2020. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*. OpenReview.net, Millennium Hall, Addis Ababa Ethiopia. Retrieved June 25, 2021 from <https://openreview.net/forum?id=rklOg6EFwS>.
- [101] A. Shafahi, M. Najibi, Z. Xu, J. P. Dickerson, L. S. Davis, and T. Goldstein. 2020. Universal adversarial training. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 4 (2020), 5636–5643.
- [102] T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii. 2015. Distributional smoothing with virtual adversarial training. Retrieved June 25, 2021 from <https://arXiv:1507.00677>.
- [103] A. Ilyas, A. Jalal, E. Asteri, C. Daskalakis, and A. G. Dimakis. 2017. The robust manifold defense: Adversarial training using generative models. Retrieved June 25, 2021 from <https://arXiv:1712.09196>.
- [104] H. Zhang and J. Wang. 2019. Defense against adversarial attacks using feature scattering-based adversarial training. In *Advances in Neural Information Processing Systems*. MIT Press, Vancouver, B.C. Canada, 1829–1839.
- [105] A. Sinha, H. Namkoong, and J. Duchi. 2017. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*. OpenReview.net, Vancouver, B.C. Canada. Retrieved from <https://openreview.net/forum?id=Hk6kPgZA>.
- [106] E. Wong and J. Z. Kolter. 2017. Provable defenses against adversarial examples via the convex outer adversarial polytope. Retrieved June 25, 2021 from <https://arXiv:1711.00851>.
- [107] A. Raghunathan, J. Steinhardt, and P. Liang. 2018. Certified defenses against adversarial examples. Retrieved June 25, 2021 from <https://arXiv:1801.09344>.
- [108] T. W. Weng, H. Zhang, P. Y. Chen, J. Yi, and L. Daniel. 2018. Evaluating the robustness of neural networks: An extreme value theory approach. Retrieved June 25, 2021 from <https://arXiv:1801.10578>.
- [109] T.-W. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, D. Boning, I. S. Dhillon, and L. Daniel. 2018. Towards fast computation of certified robustness for ReLU networks. Retrieved June 25, 2021 from <https://arXiv:1804.09699>.
- [110] H. Zhang, T.-W. Weng, and P.-Y. Chen. 2018. Efficient neural network robustness certification with general activation functions. Retrieved June 25, 2021 from <https://arXiv:1811.00866>.
- [111] E. Wong, F. Schmidt, J. H. Metzen, and J. Z. Kolter. 2018. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems*. MIT Press, Montreal, A.B. Canada, 8410–8419.
- [112] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. 2018. Certified robustness to adversarial examples with differential privacy. Retrieved June 25, 2021 from <https://arXiv:1802.03471>. 2018.
- [113] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter. 2019. Certified adversarial robustness via randomized smoothing. Retrieved June 25, 2021 from <https://arXiv:1902.02918>.
- [114] R. Zhai, C. Dan, D. He, H. Zhang, B. Gong, P. Ravikumar, C.-J. Hsieh, and L. Wang. 2020. MACER: Attack-free and scalable robust training via maximizing certified radius. In *International Conference on Learning Representations*. OpenReview.net, Millennium Hall, Addis Ababa Ethiopia. Retrieved June 25, 2021 from <https://openreview.net/forum?id=rjx1Na4Fwr>.
- [115] F. Croce, and M. Hein. 2020. Provable robustness against all adversarial ℓ_p -perturbations for $p \geq 1$. In *International Conference on Learning Representations*. OpenReview.net, Millennium Hall, Addis Ababa Ethiopia. Retrieved June 25, 2021 from https://openreview.net/forum?id=rklk_ySYPB.
- [116] A. S. Ross and F. Doshi-Velez. 2018. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *32nd AAAI Conference on Artificial Intelligence*. AAAI, New Orleans, LA USA. Retrieved June 25, 2021 from <https://ojs.aaai.org/index.php/AAAI/article/view/11504>.
- [117] M. Hein and M. Andriushchenko. 2017. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems*. MIT Press, Los Angeles, CA USA, 2263–2273.
- [118] Y. Tsuzuku, I. Sato, and M. Sugiyama. 2018. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In *Advances in Neural Information Processing Systems*. MIT Press, Montreal, A.B. Canada, 6542–6551.

- [119] A. N. Bhagoji, D. Cullina, and P. Mittal. 2017. Dimensionality reduction as a defense against evasion attacks on machine learning classifiers. Retrieved June 25, 2021 from <https://arXiv:1704.02654>.
- [120] X. Liu, M. Cheng, H. Zhang, and C.-J. Hsieh. 2018. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*. IEEE, Munich, Free State of Bavaria Germany, 369–385.
- [121] S. Ye, K. Xu, S. Liu, H. Cheng, J.-H. Lambrechts, H. Zhang, A. Zhou, K. Ma, Y. Wang, and X. Lin. 2019. Adversarial robustness vs. model compression, or both? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, Seoul, South Korea, 111–120.
- [122] J. Wang and H. Zhang. 2019. Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, Seoul, South Korea, 6629–6638.
- [123] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. 2017. Reluplex: An efficient SMT solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*. Springer, Heidelberg, BW Germany, 97–117.
- [124] Y. Sharma and P. Chen. 2018. Attacking the madry defense model with L1-based adversarial examples. Retrieved June 25, 2021 from <https://arXiv:1710.10733>.
- [125] D. Kang, Y. Sun, T. Brown, D. Hendrycks, and J. Steinhardt. 2019. Transfer of adversarial robustness between perturbation types. Retrieved June 25, 2021 from <https://arXiv:1905.01034>.
- [126] B. Li, C. Chen, W. Wang, and L. Carin. 2018. Second-order adversarial attack and certifiable robustness. Retrieved June 25, 2021 from <https://arXiv:1809.03113>.
- [127] H. Zheng, Z. Zhang, J. Gu, H. Lee, and A. Prakash. 2020. Efficient adversarial training with transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Seattle, WA USA, 1181–1190.
- [128] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu. 2017. Safety verification of deep neural networks. In *International Conference on Computer Aided Verification*. Springer, Heidelberg, BW Germany, 3–29.
- [129] V. Tjeng, K. Xiao, and R. Tedrake. 2017. Evaluating robustness of neural networks with mixed integer programming. Retrieved June 25, 2021 from <https://arXiv:1711.07356>.
- [130] J. Zhou, C. Liang, and J. Chen. 2020. Manifold projection for adversarial defense on face recognition. In *European Conference on Computer Vision*. Springer, Glasgow, UK, 288–305.
- [131] H. Zhou, K. Chen, W. Zhang, H. Fang, W. Zhou, and N. Yu. 2019. DUP-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, Seoul, South Korea, 1961–1970.
- [132] X. Jia, X. Wei, and X. Cao. 2019. Identifying and resisting adversarial videos using temporal consistency. Retrieved June 25, 2021 from <https://arXiv:1909.04837>.
- [133] X. Xu, H. Zhao, and J. Jia. 2020. Dynamic divide-and-conquer adversarial training for robust semantic segmentation. Retrieved June 25, 2021 from <https://arXiv:2003.06555>.
- [134] H. Li, G. Li, and Y. Yu. 2019. ROSA: Robust salient object detection against adversarial attacks. *IEEE Transactions on Cybernetics* 50, 11 (2019), 4835–4847.
- [135] S. Bai, Y. Li, Y. Zhou, Q. Li, and P. Torr. 2019. Adversarial metric attack and defense for person re-identification. Retrieved June 25, 2021 from <https://arXiv:1901.10650>.
- [136] D. Li and Q. Li. 2020. Adversarial deep ensemble: Evasion attacks and defenses for malware detection. *IEEE Transactions on Information Forensics and Security* 15(), 3886–3900.
- [137] H. Kwon, H. Yoon, and K.-W. Park. 2019. POSTER: Detecting audio adversarial example through audio modification. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM, London, UK, 2521–2523.
- [138] X. He, Z. He, X. Du, and T.-S. Chua. 2018. Adversarial personalized ranking for recommendation. In *41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, Ann Arbor, MI USA, 355–364.
- [139] X. Li, D. Pan, and D. Zhu. 2020. Defending against adversarial attacks on medical imaging AI system, classification or detection? Retrieved June 25, 2021 from <https://arXiv:2006.13555>.
- [140] Y. Cheng, L. Jiang, and W. Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. Retrieved June 25, 2021 from <https://arXiv:1906.02443>.
- [141] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff. 2017. On detecting adversarial perturbations. Retrieved June 25, 2021 from <https://arXiv:1702.04267>.

Received January 2020; revised April 2021; accepted May 2021