A Geometrical Approach to Evaluate the Adversarial Robustness of Deep Neural Networks

YANG WANG, Dalian University of Technology
BO DONG, Princeton University
KE XU, City University of Hong Kong
HAIYIN PIAO, Northwestern Polytechnical University
YUFEI DING, University of California, Santa Barbara
BAOCAI YIN and XIN YANG, Dalian University of Technology

Deep neural networks (DNNs) are widely used for computer vision tasks. However, it has been shown that deep models are vulnerable to adversarial attacks—that is, their performances drop when imperceptible perturbations are made to the original inputs, which may further degrade the following visual tasks or introduce new problems such as data and privacy security. Hence, metrics for evaluating the robustness of deep models against adversarial attacks are desired. However, previous metrics are mainly proposed for evaluating the adversarial robustness of shallow networks on the small-scale datasets. Although the Cross Lipschitz Extreme Value for nEtwork Robustness (CLEVER) metric has been proposed for large-scale datasets (e.g., the ImageNet dataset), it is computationally expensive and its performance relies on a tractable number of samples. In this article, we propose the Adversarial Converging Time Score (ACTS), an attack-dependent metric that quantifies the adversarial robustness of a DNN on a specific input. Our key observation is that local neighborhoods on a DNN's output surface would have different shapes given different inputs. Hence, given different inputs, it requires different time for converging to an adversarial sample. Based on this geometry meaning, the ACTS measures the converging time as an adversarial robustness metric. We validate the effectiveness and generalization of the proposed ACTS metric against different adversarial attacks on the large-scale ImageNet dataset using state-of-the-art deep networks. Extensive experiments show that our ACTS metric is an efficient and effective adversarial metric over the previous CLEVER metric.

CCS Concepts: • **Computing methodologies** → **Computer vision**; **Adversarial learning;**

Additional Key Words and Phrases: Adversarial robustness, deep neural network (DNN), image classification

This work was supported in part by the National Key Research and Development Program of China (2022ZD0210500), the National Natural Science Foundation of China (grant 61972067/U21A20491/UI908214), and the Distinguished Young Scholars Funding of Dalian (no. 2022RJ01).

Authors' addresses: Y. Wang, B. Yin, and X. Yang (corresponding author), Dalian University of Technology, 2 Linggong Road, Dalian, Liaoning, China, 116024; emails: yangwang06@mail.dlut.edu.cn, {ybc, xinyang}@dlut.edu.cn; B. Dong, Princeton University, Princeton, NJ, 08540; email: bo.dong@princeton.edu; K. Xu, City University of Hong Kong, Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong, China, 518057; email: kkangwing@gmail.com; H. Piao, Northwestern Polytechnical University, 127 West Youyi Road, Beilin District, Xi'an, Shannxi, China, 710072; email: haiyinpiao@mail.nwpu.edu.cn; Y. Ding, University of California, Santa Barbara, Santa Barbara, CA, 93106; email: yufeiding@cs.ucsb.edu.

1551-6857/2023/05-ART172 $15.00
https://doi.org/10.1145/3587936

ACM Reference format:
Yang Wang, Bo Dong, Ke Xu, Haiyin Piao, Yufei Ding, Baocai Yin, and Xin Yang. 2023. A Geometrical Approach to Evaluate the Adversarial Robustness of Deep Neural Networks. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 5s, Article 172 (May 2023), 17 pages.
https://doi.org/10.1145/3587936

1 INTRODUCTION

In recent years, **Deep Learning (DL)** has widely impacted computer vision tasks, such as object detection, visual tracking, and image editing. Despite their outstanding performance, recent studies [2, 3, 8, 12, 29, 35, 48, 58] have shown that deep methods can be easily cheated by the adversarial inputs: inputs with human imperceptible perturbations to force an algorithm to produce adversary-selected outputs. The vulnerability of deep models to adversarial inputs is getting significant attention as they are used in various security and human safety applications. Hence, a robust adversarial performance evaluation method is needed for existing deep learning models. The l_p norm-ball theory may be used to indicate the adversarial robustness of neural networks. Specifically, this theory suggests that there should exist a perturbation radius l_p-distortion $\Delta_p = \|\delta x\|_p$ [51], where any sample point x within this radius would be correctly classified as true samples, and others would be regarded as adversarial ones. In other words, the smallest radius Δ_p (i.e., minimum adversarial perturbation $\Delta_{p,min}$) can be used as a metric to evaluate the robustness: a model with a larger radius indicates that it is more robust. However, determining the $\Delta_{p,min}$ has been proven in other works [25, 45] as an NP-complete problem. Existing methods mainly focused on estimating the lower and upper bounds of $\Delta_{p,min}$. Although estimating the upper bound [7, 21, 26] is typically attack dependent, easy-to-implement, and computational lightweight, it often suffers poor generalization and accuracy. On the contrary, estimating the lower bound [50, 56] can be attack independent but computational heavy. Moreover, the lower bound estimation often provides little clues for interpreting the prevalence of adversarial examples [17, 19, 39, 56].

To address the preceding limitations, this article presents a novel instance-specific adversarial robustness metric—the **Adversarial Converging Time Score (ACTS)**. Unlike **Cross Lipschitz Extreme Value for nEtwork Robustness (CLEVER)** [51], ACTS does not use an exact lower bound of minimum adversarial perturbation as a robustness metric. Instead, the ACTS estimates the desired robustness based on the $\Delta_{p,min}$ in the direction guided by an adversarial attack. The ACTS is resilient, which means that if an attack method can deliver a $\Delta_{p,min}$ attack, then the estimated robustness by the ACTS reflects the fact. The insight behind the proposed ACTS is the geometrical characteristics of a **Deep Neural Network (DNN)**-based classifier's output manifold. Specifically, given a M-dimensional input, each output element can be regarded as a point on a $M + 1$ dimensional hypersurface. Adding adversarial perturbations can be regarded as forcing the original output elements to move to new positions on those hypersurfaces. The movement driven by effective perturbations should push all output elements to a converging curve (i.e., the intersection of two or more hypersurfaces), where a clean input is converted to an adversarial one. Since the local areas around different points on hypersurfaces have different curvatures, different clean samples require different time to be converged to adversarial examples. The proposed ACTS measures the converging time and use the time as the adversarial robustness metric. To summarize, this article has the following contributions. We propose a novel ACTS method for measuring the adversarial robustness of DNNs. Our method leverages the geometry characteristics of a DNN's output manifolds, so it is effective, efficient, and easy to understand. We provide mathematical analysis to justify the correctness of the proposed ACTS and extensive experiments to demonstrate its superiority under different adversarial attacks.

This article is organized as follows. We first review the related work in Section 2. In Section 3, we describe the proposed method. Results from comparative experiments for different architectures and adversarial attack approaches are then given in Section 4. We present our conclusions and envision future work in Section 5.

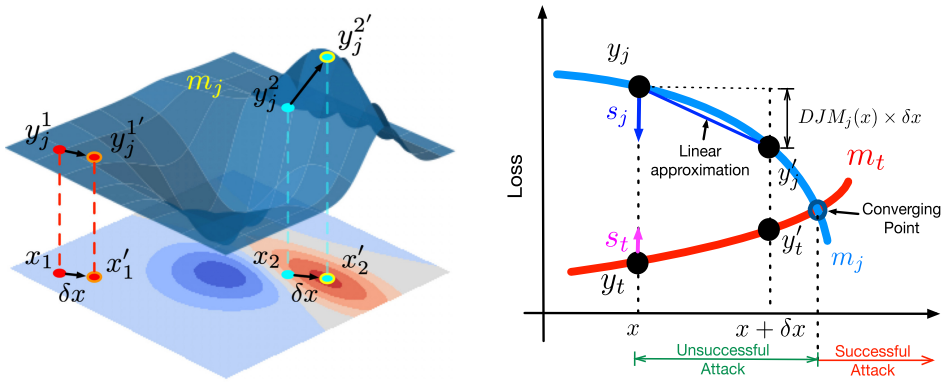
2 RELATED WORK

2.1 Adversarial Attacks

Over the past few years, extensive efforts have been made in developing new methods to generate adversarial samples [9, 13, 20, 33, 36, 53, 55, 57]. Szegedy et al. [48] proposed the L-BFGS algorithm to craft adversarial samples and showed the transferability property of these samples. Goodfellow et al. [21] proposed the **Fast Gradient Sign Method (FGSM)**, a fast approach for generating adversarial samples by adding perturbation proportional to the sign of the cost functions gradient. Rather than adding perturbation over the entire image, Papernot et al. [40] proposed the Jacobian Saliency Map Approach, which utilized the adversarial saliency maps to perturb the most sensitive input components. Kurakin et al. [26] extended the FGSM algorithm as the **Basic Iterative Method (BIM)**, which recurrently adds smaller adversarial noises. Madry et al. [34] proposed the attack **Projected Gradient Descent (PGD)** method by extending the BIM with a random start point. Carlini and Wagner [7] proposed an efficient method (i.e., CW attack) to compute good approximations while keeping low computational cost of perturbing examples. It further defined three similar targeted attacks based on different distortion measures (L_0, L_2, and L_∞). It is to be noted that all of the preceding attacks are white-box attacks that craft adversarial examples based on the input gradient. In the classical black-box attack, the adversarial algorithm has no knowledge of the architectural choices made to design the original architecture. There are different ways to generate adversarial samples [6, 10, 14–16, 23, 30, 54] under black-box schemes. Since this article focuses on the white-box attacks, for more detailed information, readers may refer to the work of Akhtar and Mian [1].

2.2 Adversarial Defenses

This line of works focuses on developing robust deep models to defend against adversarial attacks [18, 28, 49]. Goodfellow et al. [21] proposed the first adversarial defense method that uses adversarial training, in which the model is retrained with both adversarial images and the original clean dataset. A series of work [24, 34, 57] follows this adversarial training but investigates different adversarial attacks to generate different adversarial data. Papernot and McDaniel [38] extended defensive distillation [41] (which is one of the mechanisms proposed to mitigate adversarial examples) to address its limitation. They revisited the defensive distillation approach and used soft labels to train the distilled model. The resultant model was robust to attacks. Liang et al. [31] proposed a method where the perturbation to the input images is regarded as a kind of noise and the noise reduction techniques are used to reduce the adversarial effect. In their method, classical image processing operations such as scalar quantization and smoothing spatial filters were used to reduce the effect of perturbations. Bhagoji et al. [5] proposed dimensionality reduction as a defense against attacks on different machine learning classifiers. Another effective defense strategy in practice is to construct an ensemble of individual models [27]. Following this idea, Liu et al. [32] proposed the random self-ensemble method to defend the attacks by averaging the predictions over random noises injected to the model. Pang et al. [37] proposed to promote the diversity among the predictions of different models by introducing an adaptive diversity-promoting regularizer. However, these methods do not have an ideal robustness metric to help them correctly evaluate and improve their performance.



(a) 3D Hypersurface **(b)** Intuitions behind ACTS

Fig. 1. (a) An example of the 3D hypersurface. (b) Intuition behind our ACTS.

2.3 Robustness Metrics

With the development of adversarial attacks, there is a need for a robustness metric that quantifies the performance of a DNN against adversarial samples. A straightforward method is to use a specific attack method to find the adversarial examples, and use the distortions of adversarial examples (i.e., upper bound of $\Delta_{p,min}$) as the model robustness metric. For example, Bastani et al. [4] proposed a linear programming formulation to find adversarial examples and directly use the l_p-distortion as the robustness metric. Moosavi-Dezfooli et al. [36] proposed to compute a *minimal* perturbation for a given image in an iterative manner, to find the *minimal* adversarial samples across the boundary. They then define all the *minimal* perturbation expectation over the distribution of data as the robustness metric. Other methods focus on estimating the lower bound of $\Delta_{p,min}$ and use it as the evaluation metric. Weng et al. [50] exploited the ReLU property to bound the activation function (or the local Lipschitz constant) and provided two efficient algorithms (Fast-Lin and Fast-Lip) for computing a certified lower bound. Zhang et al. [56] proposed a general framework (CROWN) for computing a certified lower bound of minimum adversarial distortion and showed that the Fast-Lin algorithm is a special case under the CROWN framework. Recently, a robustness metric called *CLEVER* [51] was developed, which first estimates the local Lipschitz constant using an extreme value theory and then computes an attack-agnostic robustness score based on a first-order Lipschitz continuity condition. It can be scaled to deep networks and large datasets. However, the lower bound estimation of CLEVER is often incorrect and time consuming. Therefore, it is hard to be a robust and effective adversarial robustness metric.

3 METHODOLOGY

3.1 Adversarial Converging Time Score

Adversarial Attacks in Image Classification. Given an M-dimensional input $x \in \mathbf{R}^M$ and a K-class classification loss function $D : \mathbf{R}^M \rightarrow \mathbf{R}^K$, the predicted class label t of the input x is defined as follows:

$$t = C(x) = \underset{j}{\operatorname{argmin}}\{y_j \mid y_j \in \mathbf{R}^1\}, \tag{1}$$

where y_j is the jth element of the K-dimensional output of $D(x)$. From a geometrical point of view, y_j can be regarded as a point on an $M+1$ dimensional hypersurface m_j (Figure 1(a)).

DNN-based classifiers are typically non-linear systems, which is true for all state-of-the-art DNN models. In this case, the hypersurfaces m defined by D are also non-linear systems. Thus,

local areas around different points on a hypersurface m_j have different curvatures, which results in different inputs having different sensitivity to the same added noise δx. As shown in Figure 1(a), the changes on a hypersurface m_j driven by the same δx are significantly different in terms of magnitude. Inspired by this insight, we propose a novel ACTS as an instance-specific adversarial robustness metric. The key to the proposed ACTS is that the sensitivity is mapped to the "time" required to reach the converging curve (i.e., decision boundary) where a clean sample is converted to an adversarial sample. We first introduce the proposed ACTS in detail. Then, we provide a toy example to validate the proposed approach.

Adversarial Converging Time Score. To easily convey the intuitive idea of the proposed ACTS, we use a 1D input domain and a 2D hypersurface (i.e., lines). Figure 1(b) shows our idea intuitively. As we can see, based on Equation (1), the original input x is classified as the tth class since y_t is in a lower position than y_j in the loss domain. Although adding a noise δx to x results in two new positions y'_t and y'_j, the predicted label of $x + \delta x$ is not changed (still t). If $x + \delta x$ passes the converging point, the predicted label of $x + \delta x$ changes to j. From this point of view, the robustness of an input can be reflected by the magnitude of the added δx for reaching the converging point. For a DNN-based classifier, the collection of converging points forms the decision boundary. However, such decision boundary is extremely hard to be estimated, especially in a high-dimensional space. Instead, we can look at the converging point from the perspective of the loss domain, where the distance between y'_j and y'_t is 0. In other words, the robustness of an input can be reflected by the time used to cover the distance $y_j - y_t$—that is, the less time it requires, the less robust it is. Compared to the decision boundary estimation, estimating the distance $y_j - y_t$ is much easier. Hence, we propose the ACTS to estimate such time, which takes the following form:

$$ACTS := \underset{j}{\operatorname{argmin}} \left(f\left(\frac{y_j - y_t}{s_t - s_j} \right) \right) \quad j \in 1 \dots K, j \neq t, \tag{2}$$

$$f(x) = \begin{cases} C, & x \leq 0 \\ x, & x > 0 \end{cases}$$

where s_j and s_t are the moving speeds in the loss domain, which are driven by the added noise δx. However, the minus sign in the denominator may be a bit tricky. An ideal misclassification attack should increase the target error value, results in a positive s_t, and it should also decrease the error value of the potential misclassified class, which gives a negative s_j (as shown in Figure 1(b)). Hence, the value of $s_t - s_j$ should always be positive. However, the $s_t - s_j$ could be a negative value in the following situations: (a) s_t decreases and s_j increases; (b) both s_t and s_j decrease, but s_t decreases faster; and (c) both s_t and s_j increase, but s_j increases faster. If any of the preceding cases happen to an input, it means that it is impossible to deliver a successful attack, and hence the ACTS of the specific input is a maximum score C. The $f(x)$ used in the Equation (2) is for this purpose. Since the ACTS represents the time to cover the distance $y_j - y_t$ with a speed $s_t - s_j$, an input with a smaller ACTS is more vulnerable to an adversarial attack and vice versa.

The key to the proposed ACTS is to estimate the moving speed. However, a local neighborhood on an output hypersurface is non-linear. It is quite challenging to estimate the moving speed directly. To this end, we propose a novel **Data Jacobian Matrix (DJM)**-based scheme to estimate the required moving speed, which takes the non-linearity nature of an output hypersurface into account.

Data Jacobian Matrix. Given an input x, the DJM of D is defined as follows:

$$DJM(x) = \frac{\partial D(x)}{\partial x} = \left[\frac{\partial D_j(x)}{\partial x_i} \right]_{j \in 1 \dots K, i \in 1 \dots M}. \tag{3}$$

On a hypersurface m_j, the $DJM_j(x)$ (i.e., jth row of $DJM(x)$) defines the best linear approximation of D for points close to point x [52]. Therefore, with $DJM(x)$, a small change δx in the input domain of D can be linearly mapped to the change on the hypersurfaces m_j. Mathematically, it can be described as follows:

$$D(x + \delta x) = D(x) + DJM(x) \times \delta x + \delta e, \tag{4}$$

where $\delta e \in R^K$ is the approximation error. Essentially, the $DJM(x)$ is quite similar to the gradient backpropagated through a DNN during a training process. The only difference is that $DJM(x)$ differentiates with respect to the input x rather than network parameters.

One-Step Attack. Based on Equation (4), with an input x and an added noise δx, the original point y_j (a.k.a. $D_j(x)$) shifted to the point y'_j on the hypersurface m_j, and the approximated shifted position of y'_j can be estimated as follows (shown in Figure 1(b)):

$$y'_j \approx D_j(x) + DJM_j(x) \times \delta x, \tag{5}$$

where $DJM_j(x)$ is the jth row of the $DJM(x)$. For one-step attack (e.g., FGSM), δx can be regarded as a vector $\vec{d}$. The direction of $\vec{d}$ is fixed and only the length of $\vec{d}$ varies for delivering a successful attack. Therefore, the moving speed s_j from point y_j to y'_j on the surface m_j driven by the shift δx in the input domain can be estimated as follows:

$$s_j = \frac{y'_j - y_j}{\|\delta x\|} \approx \frac{DJM_j(X) \times \delta x}{\|\delta x\|} \quad j \in 1 \ldots K. \tag{6}$$

It is worth mentioning that the DJM is an linear approximation for a small δx. The approximation accuracy decreases while δx increases.

Multi-Step Attack. In a multi-step attack (e.g., BIM), each step changes the $\vec{d}$ (i.e., δx) in terms of both direction and length. Compared to one-step attacks, the different directions reveal more curvatures of a local neighborhood, and it increases the probability of discovering a more optimal moving speed to reduce the "time" (i.e., added noise) for converting a clean sample to an adversarial one. This is also the reason that multi-step attacks are more effective than one-step attacks. However, the dynamics introduced by multi-step attacks is also troublesome to estimate the desired moving speed. To deal with it, we propose an average moving speed from y_j to y'_j based on all explored directions as follows:

$$s_j \approx \frac{1}{N} \sum_{q}^{N} \frac{DJM_j(x) \times \delta x_q}{\|\delta x_q\|} \quad j \in 1 \ldots K, \tag{7}$$

where N is the total steps used in the multi-step attack, and δx_q is the added noise in the qth step. Even though the estimated average speed has limited accuracy, our experiments show the effectiveness of the proposed average speed.

3.2 Toy Example

We design a toy experiment to validate the proposed ACTS, where a simple two-layer feed-forward network was trained to proximate an AND gate. The testing accuracy of the trained model was 99.7%. Mathematically, we define the AND gate as follows:

$$x_1 \wedge x_2 = 1, \quad x_1 \geq 0.5 \text{ and } x_2 \geq 0.5$$
$$x_1 \wedge x_2 = 0, \quad \text{otherwise}$$

where $x_i \in [0, 1.0]$. Based on this definition, as shown in Figure 2(a), [0.5, 1.0] is the decision boundaries on both x_1 and x_2 axes, where lower ACTSs are expected. We use the FGSM method, with $\epsilon = 0.1$, to generate adversarial samples only for the clear sample pairs of $x_1 \wedge x_2 = 1$. For

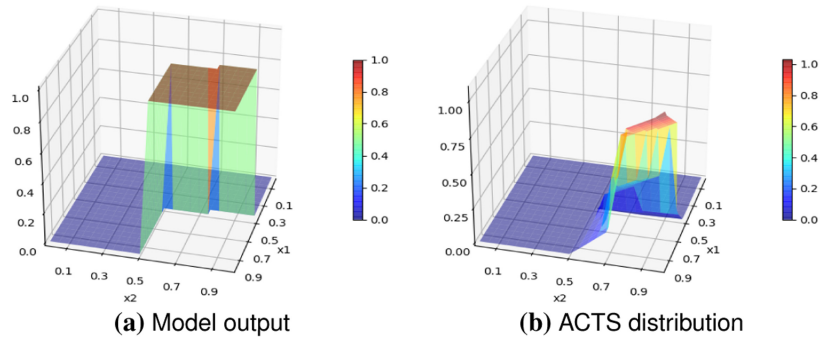


Fig. 2. (a) The output of the toy example AND gate model. (b) The ACTS distribution of the toy example AND gate model for all input samples where $x_1 \wedge x_2 = 1$.

the rest pairs, ACTSs are set to 0. As shown in Figure 2(b), the input pairs closer to the decision boundary have lower ACTSs. In addition, we observe an increasing trend in the ACTSs as the input values move further away from the boundary. The maximum ACTS is observed at point $(x_1, x_2) = (1.0, 1.0)$. These observations illustrate that the proposed ACTS is able to reflect the robustness under FGSM attack.

4 EXPERIMENTS

In this section, we first validate the effectiveness and generalization capacity of the proposed ACTS metric against different state-of-the-art DNN models and adversarial attack approaches on the ImageNet [11] dataset in Section 4.2. We then compare the proposed ACTS with CLEVER [51] (the only method that can be adapted to deep models and large-scale ImageNet dataset) to show that our method provides a more effective and practical robustness metric in different adversarial settings in Section 4.3.

4.1 Experimental Setting

Evaluation Dataset and Methods. To evaluate the effectiveness of the proposed method on large-scale datasets, we chose the ImageNet Large Scale Visual Recognition Challenge (ILSVR) 2012 dataset, which has 1.2 million training and 50,000 validation images. We evaluate our method on three representative state-of-the-art deep networks with pre-trained models provided by PyTorch [42]—that is, the InceptionV3 [47], ResNet50 [22], and VGG16 [44], as these deep networks have their own network architectures. To evaluate the robustness of our method against different attacks, we consider three different state-of-the-art white-box attack approaches: FGSM [21], BIM [26], and PGD [34].

Implementation Details. We have implemented our ACTS using the PyTorch framework, and all attack methods using the adversarial robustness PyTorch library: Torchattacks [43]. A GPU-Server with an Intel E5-2650 v4 2.20GHz CPU (with 32 GB of RAM) and one NVIDIA Tesla V100 GPU (with 24 GB of memory) is used in our experiments. For preprocessing, we normalize the data using mean and standard deviation. The images are loaded in the range of $[0, 1]$ and then normalized using a $mean = [0.485, 0.456, 0.406]$ and $std = [0.229, 0.224, 0.225]$ [42]. To control the noise levels so as not to bring any noticeable perceptual differences and show the consistent performance of the proposed ACTS, we add the noise of three different levels: $\epsilon = \{0.00039\ (0.1/255), 0.00078\ (0.2/255), 0.00117\ (0.3/255)\}$ to the FGSM [21], BIM [26], and PGD [34], respectively. We use N1, N2, and N3 to represent these three different noise levels, respectively. We use three steps

Table 1. Clean and Adversarial Accuracy in Different Adversarial Environments

Attack	Model	Clean	Adversarial Accuracy		
			N1	N2	N3
FGSM	InceptionV3	77.21%	61.16%	50.03%	43.05%
	ResNet50	76.13%	57.45%	43.94%	34.77%
	VGG16	71.59%	52.35%	37.73%	27.71%
BIM	InceptionV3	77.21%	55.10%	43.05%	36.68%
	ResNet50	76.13%	50.08%	34.77%	25.47%
	VGG16	71.59%	44.38%	27.71%	18.46%
PGD	InceptionV3	77.21%	60.20%	45.86%	36.31%
	ResNet50	76.13%	56.13%	39.15%	26.56%
	VGG16	71.59%	51.77%	34.48%	22.28%

and set the step size = $\epsilon/2$ to both BIM [26] and PGD [34]. We use the untargeted attack setting in all attacks. For each image, we evaluate its top-10 class (i.e., the class with the top-10 maximum probabilities except for the true class, which is usually the easiest target to attack) [51] in DJM.

4.2 ACTS Validation Results

This section evaluates the effectiveness and generalization properties of our proposed ACTS method in various adversarial environments.

Evaluating the Effectiveness of the ACTS. To be an effective adversarial robustness metric, the proposed ACTS should faithfully reflect that the samples with lower ACTSs are more prone to be attacked successfully than those with higher scores. To validate such property of the ACTS, we design the following experiments. First, we apply these three DNN models on the ImageNet validation dataset and selected those correctly classified images. Second, we estimate the ACTSs for all the selected images and apply the three chosen attack methods to them. It can be seen from Table 1 that the adversarial accuracy is gradually decreased to a moderate extent with increased levels of noise. Third, to show the consistent performance of the proposed ACTS, we increase the noise level to N1, N2, and N3 and record the ACTSs for those who are successfully attacked. Figure 3 shows the histograms of the three chosen DNN models under different attacks, respectively. The blue color indicates the ACTSs of the images that are correctly classified on ImageNet validation dataset under the initial noise $\epsilon = 0.0002$, and the other three colors indicate the ones that are attacked successfully with noise levels N1, N2, and N3. For all the models and attacks, the green, yellow, and red regions are always on the very left side of the respective charts. This shows that the inputs with lower ACTSs are easier to be attacked successfully. We can also see that with increased noise levels, images with relatively lower ACTSs would be attacked successfully first (from green to red). In addition, Figure 3 and Table 1 show that these aforementioned observations are consistent in different adversarial environments (i.e., different DNN architectures and different attacks). Based on the distribution of the obtained ACTS, we are able to gain a relatively precise intuition about DNNs' performance under different attack methods. For example, from the horizontal comparison of each row in Figure 3, it is obvious that the green, yellow, and red bands in the second and third columns of ACTs histograms are wider than those in the first column, indicating that BIM and PGD are more powerful attack methods when compared to the FGSM attack method. These observations can be confirmed by the corresponding adversarial accuracy rates shown in Table 1.

In addition to the qualitative results in Figure 3, we present the quantitative results to show the effectiveness of the ACTS. Figures 4, 5, and 6 show the detailed histogram results in different adversarial environments. The orange color indicates the samples that are attacked successfully, and the

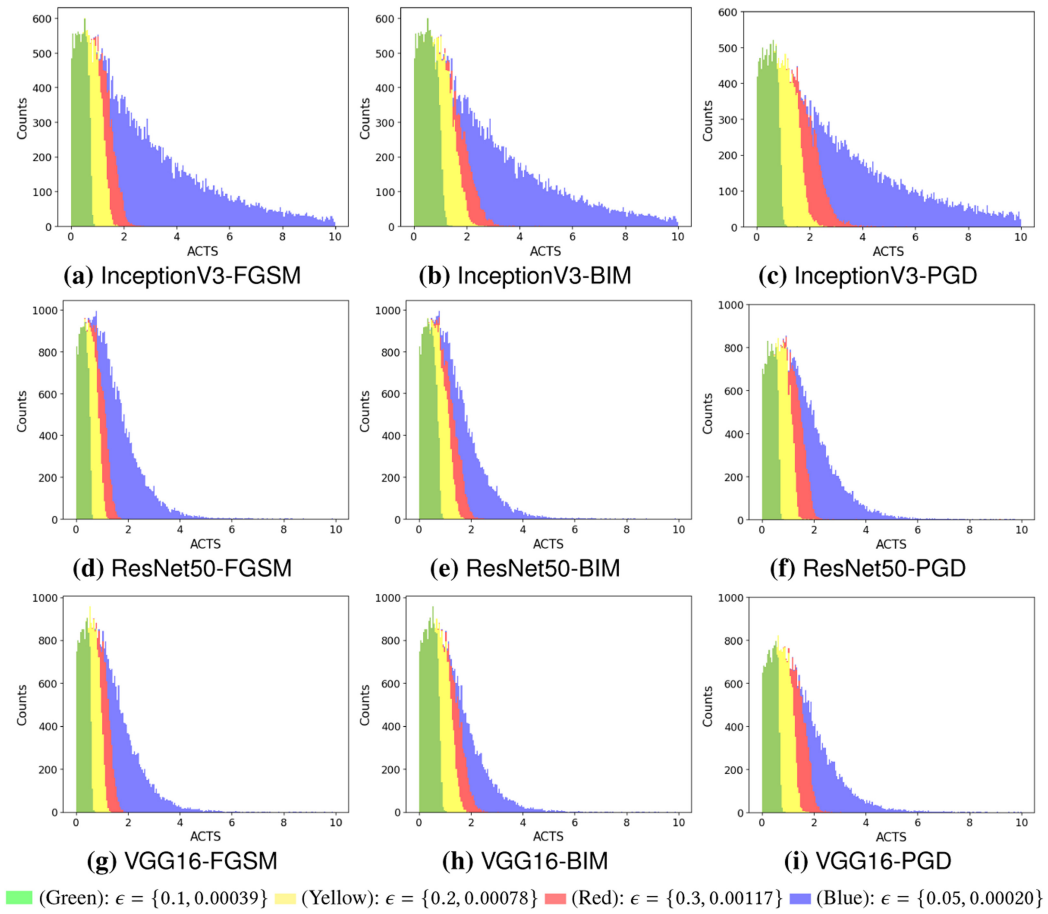


Fig. 3. Noise effectiveness charts for different models under different attacks. The area under the blue color denotes ACTSs for the correctly classified samples on the ImageNet validation dataset. Green, yellow, and red colors denote ACTSs of the samples that were successfully attacked. Each color denotes the noise level added to the dataset with respect to the corresponding attack.

light blue color indicates the ones that are attacked unsuccessfully. Only an ideal robustness metric could separate the two groups without any overlap, and existing approaches may have different overlap regions between the two groups. Hence, the size of overlap regions can be leveraged as an indicator to show the effectiveness of a robustness metric. For each histogram, we calculate the overlap percentage by S_o/S_a, where S_o is the size (i.e., count) of an overlap area and S_a is the total area of a histogram. Hence, for Overlap%, the lower its values, the better the evaluating results are. All results are shown in Table 2. As we can see, almost all Overlap% values are below 10%. In terms of DNN architecture, the ACTS shows better performance on InceptionV3 and VGG16. We guess the reason is that the local areas on output hypersurfaces of InceptionV3 and VGG16 around the output points of all tested images are flatter (i.e., the radius of curvature is small) than ResNet50. In this case, the DJM provides a more accurate linear approximation. It is worth mentioning that the overlap area is getting larger when ϵ increases in different adversarial environments. This confirms with the limitation of the DJM that the linear approximation accuracy decreases while

Table 2. ACTS Overlap% Values in Different Adversarial Environments

Attack	Model	Overlap% N1	Overlap% N2	Overlap% N3
FGSM	InceptionV3	1.46%	3.54%	4.71%
	ResNet50	2.95%	6.42%	9.14%
	VGG16	2.14%	4.29%	5.71%
BIM	InceptionV3	2.53%	4.71%	6.26%
	ResNet50	4.89%	9.13%	10.89%
	VGG16	3.02%	5.71%	6.47%
PGD	InceptionV3	1.33%	3.26%	4.85%
	ResNet50	1.72%	4.70%	6.62%
	VGG16	1.87%	3.73%	4.89%

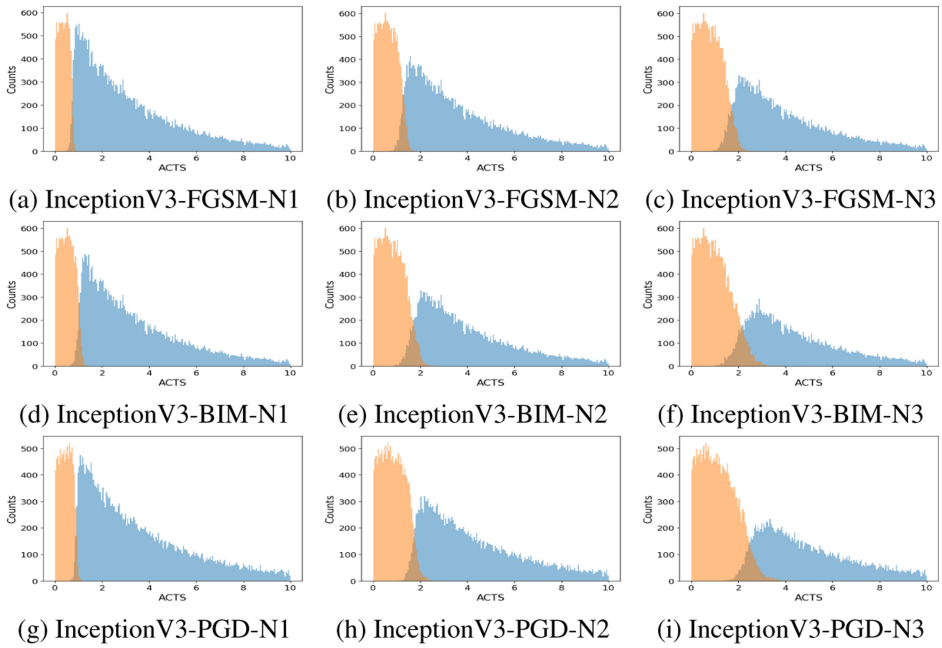


Fig. 4. ACTS histograms of InceptionV3 in different experimental configurations. In each histogram, the orange color indicates the samples that are attacked successfully, and the light blue color indicates the ones that are attacked unsuccessfully.

δx increases. In the process of statistics, we found an interesting phenomenon called the *attack flip*: the image with a successful attack at a lower noise level may fail at a higher noise level. The result is shown in Table 3. *Attack flip* is a good explanation for why there are very small ACTSs in the overlap at a higher noise level. In other words, some small ACTSs are counted as an orange histogram at a lower noise level and then counted as a blue histogram at a higher noise level. This flip will result in small ACTSs in the overlap at a higher noise level. Besides, another reason is the limitation of the DJM. The linear approximation accuracy of the DJM decreases while δx increases, which will lead to the error. Attack flip also suggests that the lower bound may not always make sense.

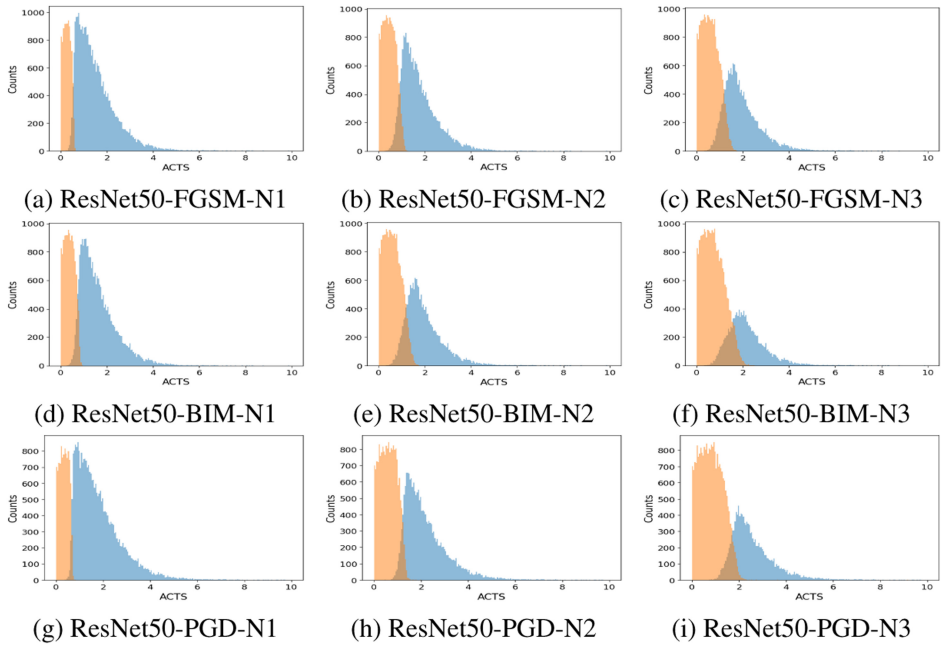


Fig. 5. ACTS histograms of ResNet50 in different experimental configurations. In each histogram, the orange color indicates the samples that are attacked successfully, and the light blue color indicates the ones that are attacked unsuccessfully.

Table 3. Numbers of Images in ImageNet for Different Architectures Found in Different " Flip" Experimental Configurations

Attack Type	FGSM			BIM			PGD		
Attack Flip	1	2	3	1	2	3	1	2	3
InceptionV3	0	0	1	1	10	10	0	0	2
ResNet50	0	1	1	0	3	5	1	0	3
VGG16	0	0	0	0	3	4	2	3	8

The "1" means attack flip between different noise levels N1 and N2. The "2" means attack flip between different noise levels N1 and N3. The "3" means attack flip between different noise levels N2 and N3.

Evaluating the Generalization of the ACTS. In Figure 3, the histogram of each row represent the results of the same model under different attack, and each column represent the results of different models under the same attack method. From the results, we can see that the ACTS has a good generalization ability across different attack methods and models.

Correlations to CLEVER. We are interested in whether our ACTS aligns with CLEVER. To this end, we compute the average score of all the tested images as CLEVER's reported robustness number. The higher the CLEVER score, the more robust the model is. We also calculate the average ACTS of all the tested images to represent the robustness of the network. From the results shown in Table 4, we get the same ranking correlation as CLEVER. This also demonstrates that models with higher ACTSs are more robust. The results we obtained are basically consistent with the results in CLEVER method [51]. Besides, we conclude that the VGG16 model with the highest scores is

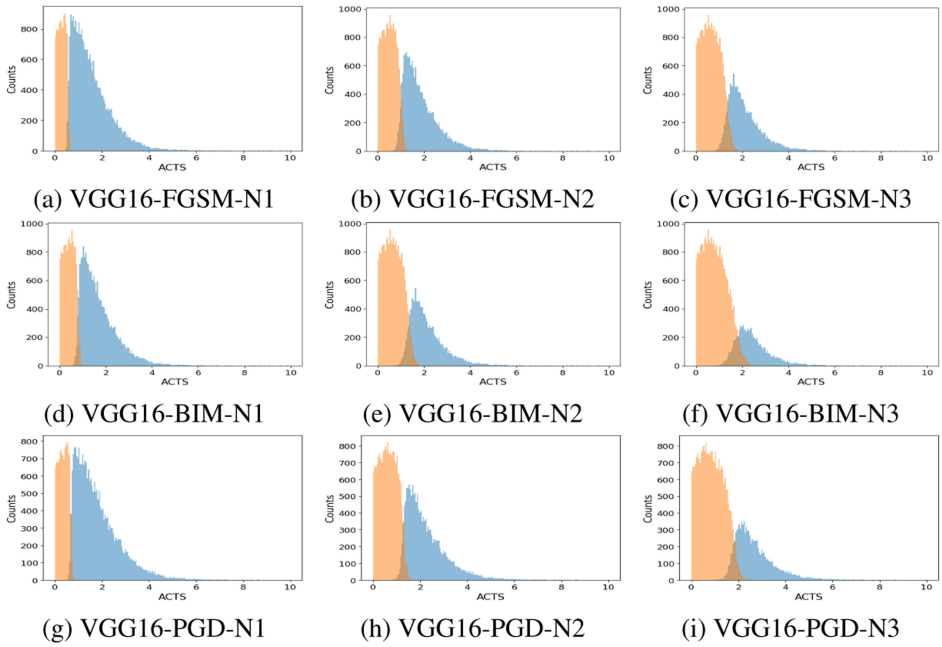


Fig. 6. ACTS histograms of VGG16 in different experimental configurations. In each histogram, the orange color indicates the samples that are attacked successfully, and the light blue color indicates the ones that are attacked unsuccessfully.

Table 4. Using the CLEVER Method and the ACTS Method to Measure the Ranking Correlation of the Robustness of Different Models

Model	CLEVER	ACTS
VGG16	0.370	4.459
InceptionV3	0.215	3.047
ResNet50	0.126	2.558

more robust than others on the test image set. This conclusion can also be found in the work of Su et al. [46]. It is worth mentioning that the score distribution may change dramatically on different test image sets.

Determining k. To investigate the impact of the top-k class in DJM, for each image, we evaluate its top-k class ACTSs in Table 5. From the results, we can see that with the increase of k, the value of Overlap% changes very slightly. Considering the balance between computational consumption and ACTSs' performance, it is reasonable to set the k to 10.

4.3 Comparing with the State-of-the-Art CLEVER

We compare our method with the state-of-the-art method CLEVER in this section. The CLEVER score is designed for estimating the lower bound on the minimal distortion required to craft an adversarial sample, and it used L_2 and L_∞ norms for their validations. We follow the setting in the work of Weng et al. [51] to compute CLEVER L_2 and L_∞ norms scores for 1,000 images out

Table 5. Top-k Class ACTS Overlap% Values in Different Environments

Attack	Model	Metric	Overlap% N1	Overlap% N2	Overlap% N3
FGSM	InceptionV3	ACTS-10	1.56%	2.08%	5.2%
		ACTS-20	1.56%	2.08%	5.2%
		ACTS-50	1.56%	2.08%	5.2%
	ResNet50	ACTS-10	2.39%	7.98%	9.84%
		ACTS-20	2.39%	7.98%	9.71%
		ACTS-50	2.39%	7.98%	9.71%
	VGG16	ACTS-10	1.3%	3.46%	6.63%
		ACTS-20	1.3%	3.46%	6.77%
		ACTS-50	1.3%	3.46%	6.92%
BIM	InceptionV3	ACTS-10	1.43%	5.2%	6.37%
		ACTS-20	1.43%	5.2%	6.37%
		ACTS-50	1.43%	5.2%	6.37%
	ResNet50	ACTS-10	4.39%	9.84%	10.9%
		ACTS-20	4.26%	9.71%	11.17%
		ACTS-50	4.26%	9.71%	11.17%
	VGG16	ACTS-10	2.74%	3.63%	5.91%
		ACTS-20	2.74%	3.63%	5.91%
		ACTS-50	2.74%	3.63%	5.91%
PGD	InceptionV3	ACTS-10	0.91%	3.25%	4.81%
		ACTS-20	0.91%	3.25%	4.42%
		ACTS-50	0.91%	2.99%	4.68%
	ResNet50	ACTS-10	0.93%	5.32%	6.12%
		ACTS-20	0.93%	5.45%	6.52%
		ACTS-50	0.93%	5.19%	6.38%
	VGG16	ACTS-10	1.3%	3.03%	4.61%
		ACTS-20	1.44%	3.17%	4.61%
		ACTS-50	1.44%	3.17%	4.76%

of the all 50,000 in the ImageNet validation set, as CLEVER is more computationally expensive. The same set of randomly selected 1,000 images from the ImageNet validation set is also used in our method. Instead of sampling a high-dimension-space ball, our method only requires normal backpropagations, which is significantly faster than CLEVER. Our experimental results in Table 6 confirm this.

For each image, we calculate its CLEVER score and ACTSs on an NVIDIA Tesla V100 graphics card, and the average computation speed of our method is three orders of magnitude faster than the CLEVER method on different models. We also use the Overlap% indicator to compare the effectiveness of different robustness metrics, inspired by the ROC curve, which visualizes all possible classification thresholds to quantify the performance of a classifier. Since the ACTS and CLEVER only care about whether the distribution of image scores are consistent with successful/unsuccessful results in different adversarial environments, we can use the Overlap% indicator as the "mis-classification rate." In Table 7, we calculate L_2 CLEVER, L_∞ CLEVER, and ACTS Overlap% values, respectively. From the results, we can see that the value range of the L_2 CLEVER and L_∞ CLEVER Overlap% is almost in the 10% to 20% range, and the value range of the ACTS Overlap% is almost in the 0% to 10% range. CLEVER scores have almost more than twice larger

Table 6. Average Computation Time of CLEVER and the ACTS on Different Models for a Single Image in ImageNet

Model	Metric	Average Computation Time (seconds)	ACTS Speedup
InceptionV3	CLEVER	331.42	6,628/2,549
	ACTS	0.05/0.13	
ResNet50	CLEVER	196.25	4,906/2,181
	ACTS	0.04/0.09	
VGG16	CLEVER	286.85	5,737/2,207
	ACTS	0.05/0.13	

Blue and red fonts in the third column represent ACTS average computation time under one-step and multi-step attacks, respectively. The fourth column is the corresponding lifting multiple.

Table 7. Comparing the ACTS with CLEVER Overlap% Values in Different Adversarial Environments

Attack	Model	Metric	Overlap% N1	Overlap% N2	Overlap% N3
FGSM	InceptionV3	L_2 CLEVER	14.34%	15.36%	17.80%
		L_∞ CLEVER	13.7%	15.88%	17.67%
		ACTS	**1.56%**	**2.08%**	**5.2%**
	ResNet50	L_2 CLEVER	17.11%	18.6%	19.14%
		L_∞ CLEVER	13.61%	15.5%	17.25%
		ACTS	**2.39%**	**7.98%**	**9.84%**
	VGG16	L_2 CLEVER	10.43%	11.59%	13.48%
		L_∞ CLEVER	8.99%	10.43%	12.75%
		ACTS	**1.3%**	**3.46%**	**6.63%**
BIM	InceptionV3	L_2 CLEVER	11.91%	12.16%	10.88%
		L_∞ CLEVER	11.91%	11.65%	10.5%
		ACTS	**1.43%**	**5.2%**	**6.37%**
	ResNet50	L_2 CLEVER	16.44%	16.31%	14.42%
		L_∞ CLEVER	13.34%	15.23%	13.34%
		ACTS	**4.39%**	**9.84%**	**10.9%**
	VGG16	L_2 CLEVER	10.72%	11.45%	14.2%
		L_∞ CLEVER	9.13%	10.58%	13.19%
		ACTS	**2.74%**	**3.63%**	**5.91%**
PGD	InceptionV3	L_2 CLEVER	12.04%	11.4%	11.91%
		L_∞ CLEVER	11.01%	11.27%	11.78%
		ACTS	**0.91%**	**3.25%**	**4.81%**
	ResNet50	L_2 CLEVER	14.15%	15.77%	14.82%
		L_∞ CLEVER	12.13%	12.94%	12.8%
		ACTS	**0.93%**	**5.32%**	**6.12%**
	VGG16	L_2 CLEVER	7.54%	10.14%	13.77%
		L_∞ CLEVER	7.68%	8.7%	12.17%
		ACTS	**1.3%**	**3.03%**	**4.61%**

Overlap% values on average for all testing configurations. Even though L_∞ CLEVER scores give slightly less Overlap% values than the ones based on L_2 CLEVER scores, ACTSs still outperform them in a significant margin with all testing configurations. These results indicate that the ACTS is a more effective metric than CLEVER in different adversarial environments.

5 CONCLUSION AND FUTURE WORK

In this work, we proposed the ACTS as an instance-specific adversarial robustness metric. The ACTS is inspired by the geometrical insight of the output hypersurfaces of a DNN classifier. We performed a comprehensive set of experiments to substantiate the effectiveness and generalization of our proposed metric. Compared to CLEVER, we proved that the ACTS can provide a faster and more effective adversarial robustness prediction for different attacks across various DNN models. More importantly, the ACTS solved the adversarial robustness problem from a geometrical point of view. We believe that it provides a meaningful angle and insight into the adversarial robustness problem, which will help future work in the same vein.

In the future, we will focus on improving the DNN's adversarial performance by leveraging the proposed ACTS. Another interesting direction to look into is extending the ACTS to make it work under black-box attack methods.

REFERENCES

[1] Naveed Akhtar and Ajmal Mian. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* 6 (2018), 14410–14430.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the International Conference on Machine Learning*.

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. 2018. Synthesizing robust adversarial examples. In *Proceedings of the International Conference on Machine Learning*.

[4] Osbert Bastani, Yani Ioannou, Leonidas Lampropoulos, Dimitrios Vytiniotis, Aditya Nori, and Antonio Criminisi. 2016. Measuring neural net robustness with constraints. In *Advances in Neural Information Processing Systems*.

[5] Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. 2017. Dimensionality reduction as a defense against evasion attacks on machine learning classifiers. *arXiv:1704.02654* (2017).

[6] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2018. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *Proceedings of the International Conference on Learning Representations*.

[7] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *Proceedings of the IEEE Symposium on Security and Privacy*.

[8] Pin Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho Jui Hsieh. 2017. EAD: Elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[9] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui 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*.

[10] Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. 2018. Query-efficient hard-label black-box attack: An optimization-based approach. In *Proceedings of the International Conference on Learning Representations*.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[12] Jianchuan Ding, Bo Dong, Felix Heide, Yufei Ding, Yunduo Zhou, Baocai Yin, and Xin Yang. 2022. Biologically inspired dynamic thresholds for spiking neural networks. In *Advances in Neural Information Processing Systems*.

[13] Yinpeng Dong, Qi-An Fu, Xiao Yang, Tianyu Pang, Hang Su, Zihao Xiao, and Jun Zhu. 2020. Benchmarking adversarial robustness on image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[14] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*.

[15] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*.

[16] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. 2019. Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*.

[17] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. 2017. A rotation and a translation suffice: Fooling CNNs with simple transformations. *arXiv:1712.02779* (2017).

[18] Claudio Ferrari, Federico Becattini, Leonardo Galteri, and Alberto Del Bimbo. 2022. (Compress and Restore)^N: A robust defense against adversarial attacks on image classification. *ACM Transactions on Multimedia Computing, Communications, and Applications* 19, 1s (2022), Article 26, 16 pages.

[19] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. 2018. AI2: Safety and robustness certification of neural networks with abstract interpretation. In *Proceedings of the IEEE Symposium on Security and Privacy*.

[20] Partha Ghosh, Arpan Losalka, and Michael J. Black. 2019. Resisting adversarial attacks using Gaussian mixture variational autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[21] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv:1412.6572* (2014).

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[23] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018. Black-box adversarial attacks with limited queries and information. In *Proceedings of the International Conference on Machine Learning*.

[24] Harini Kannan, Alexey Kurakin, and Ian J. Goodfellow. 2018. Adversarial logit pairing. *arXiv:1803.06373* (2018).

[25] Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. 2017. Reluplex: An efficient SMT solver for verifying deep neural networks. In *Proceedings of the International Conference on Computer Aided Verification*.

[26] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv:1607.02533* (2016).

[27] Alexey Kurakin, Ian J. Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, et al. 2018. Adversarial attacks and defences competition. *arXiv:1804.00097* (2018).

[28] H. Li, G. Li, and Y. Yu. 2020. ROSA: Robust salient object detection against adversarial attacks. *IEEE Transactions on Cybernetics* 50, 11 (2020), 4835–4847.

[29] Jiguo Li, Xinfeng Zhang, Jizheng Xu, Siwei Ma, and Wen Gao. 2021. Learning to fool the speaker recognition. *ACM Transactions on Multimedia Computing* 17, 3s (2021), Article 109, 21 pages.

[30] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. 2019. NATTACK: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In *Proceedings of the International Conference on Machine Learning*.

[31] Bin Liang, Hongcheng Li, Miaoqiang Su, Xirong Li, Wenchang Shi, and Xiaofeng Wang. 2017. Detecting adversarial examples in deep networks with adaptive noise reduction. *arXiv:1705.08378* (2017).

[32] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. 2018. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision*.

[33] Xuanqing Liu, Yao Li, Chongruo Wu, and Cho-Jui Hsieh. 2018. Adv-BNN: Improved adversarial defense through robust Bayesian neural network. In *Proceedings of the International Conference on Learning Representations*.

[34] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv:1706.06083* (2017).

[35] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, Pascal Frossard, and Stefano Soatto. 2018. Robustness of classifiers to universal perturbations: A geometric perspective. In *Proceedings of the International Conference on Learning Representations*.

[36] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal 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*.

[37] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. 2019. Improving adversarial robustness via promoting ensemble diversity. In *Proceedings of the International Conference on Machine Learning*.

[38] Nicolas Papernot and Patrick McDaniel. 2017. Extending defensive distillation. *arXiv:1705.05264* (2017).

[39] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the ACM on Asia Conference on Computer and Communications Security*.

[40] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *Proceedings of the IEEE Symposium on Security and Privacy*.

[41] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *Proceedings of the IEEE Symposium on Security and Privacy*.
[42] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS'17)*.
[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, et al. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*.
[44] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556* (2014).
[45] Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. 2017. Certifying some distributional robustness with principled adversarial training. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17)*.
[46] Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng 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)*. 631–648.
[47] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[48] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv:1312.6199* (2013).
[49] Chao Tong, Mengze Zhang, Chao Lang, and Zhigao Zheng. 2021. An image privacy protection algorithm based on adversarial perturbation generative networks. *ACM Transactions on Multimedia Computing* 17, 2 (2021), Article 43, 14 pages.
[50] Tsui Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho Jui Hsieh, Duane Boning, Inderjit S. Dhillon, and Luca Daniel. 2018. Towards fast computation of certified robustness for ReLU networks. In *Proceedings of the International Conference on Machine Learning*.
[51] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. 2018. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv:1801.10578* (2018).
[52] Wikipedia. 2018. Jacobian Matrix and Determinant. Retrieved February 26, 2018 from https://www.en.wikipedia.org/wiki/Jacobian_matrix_and_determinant.
[53] Eric Wong, Leslie Rice, and J. Zico Kolter. 2020. Fast is better than free: Revisiting adversarial training. In *Proceedings of the 8th International Conference on Learning Representations*.
[54] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. 2019. Improving transferability of adversarial examples with input diversity. In *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*.
[55] E. Yang, T. Liu, C. Deng, and D. Tao. 2020. Adversarial examples for Hamming space search. *IEEE Transactions on Cybernetics* 50, 4 (2020), 1473–1484.
[56] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. 2018. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems*.
[57] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. 2019. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the International Conference on Machine Learning*.
[58] Jiqing Zhang, Xin Yang, Yingkai Fu, Xiaopeng Wei, Baocai Yin, and Bo Dong. 2021. Object tracking by jointly exploiting frame and event domain. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13043–13052.

Received 3 September 2022; revised 19 December 2022; accepted 26 February 2023