Pattern Recognition 137 (2023) 109308

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

A comprehensive evaluation framework for deep model robustness

Jun Guo[a], Wei Bao[b], Jiakai Wang[c], Yuqing Ma[a], Xinghai Gao[d], Gang Xiao[e], Aishan Liu[a,*], Jian Dong[a,b], Xianglong Liu[a,c,f,*], Wenjun Wu[a]

[a] State Key Lab of Software Development Environment, Beihang University, Beijing, China
[b] China Electronics Standardization Institute, Beijing, China
[c] Zhongguancun Laboratory, Beijing, China
[d] Institute of Unmanned System, Beihang University, Beijing, China
[e] National Key Laboratory for Complex Systems Simulation, Beijing, China
[f] Institute of Data Space, Hefei Comprehensive National Science Center, Hefei, Anhui, China

ARTICLE INFO

Article history:
Received 4 July 2022
Revised 25 October 2022
Accepted 4 January 2023
Available online 8 January 2023

Keywords:
Adversarial examples
Evaluation metrics
Model robustness

ABSTRACT

Deep neural networks (DNNs) have achieved remarkable performance across a wide range of applications, while they are vulnerable to adversarial examples, which motivates the evaluation and benchmark of model robustness. However, current evaluations usually use simple metrics to study the performance of defenses, which are far from understanding the limitation and weaknesses of these defense methods. Thus, most proposed defenses are quickly shown to be attacked successfully, which results in the “arm race” phenomenon between attack and defense. To mitigate this problem, we establish a model robustness evaluation framework containing 23 comprehensive and rigorous metrics, which consider two key perspectives of adversarial learning (i.e., data and model). Through neuron coverage and data imperceptibility, we use data-oriented metrics to measure the integrity of test examples; by delving into model structure and behavior, we exploit model-oriented metrics to further evaluate robustness in the adversarial setting. To fully demonstrate the effectiveness of our framework, we conduct large-scale experiments on multiple datasets including CIFAR-10, SVHN, and ImageNet using different models and defenses with our open-source platform. Overall, our paper provides a comprehensive evaluation framework, where researchers could conduct comprehensive and fast evaluations using the open-source toolkit, and the analytical results could inspire deeper understanding and further improvement to the model robustness.

1. Introduction

Deep learning models have achieved remarkable performance across a wide range of applications, however, they are susceptible to *adversarial examples* [1]. Since deep learning has been integrated into various safety-critical scenarios, the safety problem brought by adversarial examples has attracted extensive attention from the perspectives of both adversarial attack [2,3] and defense [4–6]. Evaluating and benchmarking the robustness of deep learning models, as a direct and effective approach, paves a very fundamental path to better understanding and further improving model robustness [7–9]. However, most of these works focus on providing practical advice or benchmarking the performance of adversarial defenses, which ignore the significance of evaluation metrics. By adopting the simple evaluation metrics (e.g., attack success rate, classification accuracy), most of the current studies could only use model outputs to conduct incomplete evaluations, which fail to provide comprehensive understandings of the limitations of these defenses. Thus, these defenses are quickly shown to be attacked successfully, which results in the “arm race” phenomenon between attacks and defenses. Therefore, it is of great significance and challenge to conduct rigorous and extensive evaluation on robustness for navigating the research field and further facilitating trustworthy deep learning in practice.

In this work, with a hope to facilitate future research, we establish a model robustness evaluation framework containing a comprehensive, rigorous, and coherent set of evaluation metrics. These metrics could fully evaluate model robustness and provide deep insights into building robust models. This paper focuses on the robustness of deep learning models on the most commonly studied image classification tasks with respect to ℓ_p-norm bounded adversaries and some other corruption. As illustrated in Fig. 1, our evaluation framework can be roughly divided into two parts: data-oriented and model-oriented, which focus on the two key factors of adversarial learning (i.e., data and model). Data-oriented met-

[*] Corresponding authors.
E-mail addresses: liuaishan@buaa.edu.cn (A. Liu), xlliu@buaa.edu.cn (X. Liu).

https://doi.org/10.1016/j.patcog.2023.109308

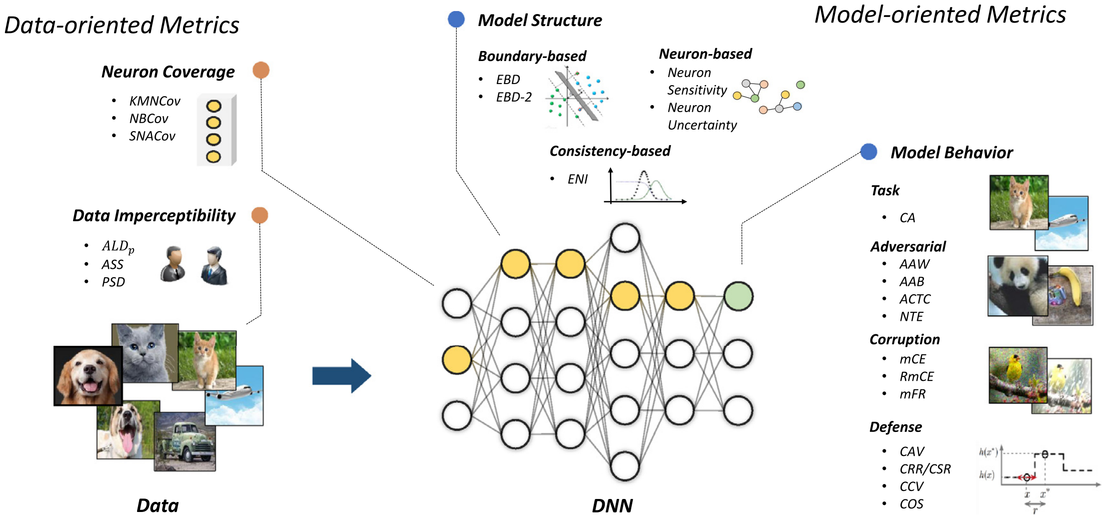


Fig. 1. With 23 evaluation metrics in total, our comprehensive evaluation framework primarily focuses on the two key factors of adversarial learning (i.e., data and model).

rics focus on the integrity of test examples (i.e., whether the conducted evaluation covers most of the neurons within a model), while model-oriented metrics consider both model structures (e.g., neurons, layers) and behaviors in the adversarial setting (e.g., adversarial performance, decision boundary). Our framework contains 23 evaluation metrics in total.

To fully demonstrate the effectiveness of the evaluation framework, we then conduct large-scale experiments on multiple datasets (i.e., CIFAR-10 [10], SVHN [11], and ImageNet [12]) using different models with different adversarial defense strategies. Through the experimental results, we could conclude that: (1) though showing high performance on some simple and intuitive metrics such as adversarial accuracy, some defenses are weak on more rigorous and insightful metrics; (2) besides ℓ_∞-norm adversarial examples, more diversified attacks should be performed to conduct comprehensive evaluations (e.g., corruption attacks, ℓ_2 adversarial attacks, etc.); (3) apart from model robustness evaluation, the proposed metrics shed light on the model robustness and are also beneficial to the design of adversarial attacks and defenses. All evaluation experiments are conducted on our new adversarial robustness evaluation platform, which we hope could facilitate fellow researchers for a better understanding of adversarial examples as well as further improvement of model robustness. Our contributions are as follows:

- We establish a comprehensive evaluation framework for model robustness containing 23 data-oriented and model-oriented metrics, which could fully evaluate model robustness through static structure and dynamic behavior, and provide deep insights into building robust models;
- Based on our framework, we provide an open-sourced platform, which supports continuous integration of user-specific algorithms and language-independent models;
- We conduct large-scale experiments, and we provide preliminary suggestions to the design of adversarial attacks/defenses in the future. Meanwhile, we provide suggestions on the selection of proper metrics with examples.

2. Related work

2.1. Adversarial attacks and defenses

Adversarial examples are inputs intentionally designed to mislead DNNs [1,2]. Given a DNN f and an input image $\mathbf{x} \in \mathbb{X}$ with the ground truth label $\mathbf{y} \in \mathbb{Y}$, an adversarial example $\mathbf{x}_{adv}$ satisfies

$$f(\mathbf{x}_{adv}) \neq \mathbf{y} \quad \text{s.t.} \quad \|\mathbf{x} - \mathbf{x}_{adv}\| \leq \epsilon,$$

where $\|\cdot\|$ is a distance metric measured by the ℓ_p-norm ($p \in \{1,2,\infty\}$).

In the past years, great efforts have been devoted to generating adversarial examples in different scenarios and tasks [13,14]. Adversarial attacks can be divided into two types: white-box attacks, in which adversaries have the complete knowledge of the target model and can fully access the model [2,5,15,16]; black-box attacks, in which adversaries have limited knowledge of the target classifier and can not directly access the model [17–19].

Meanwhile, to improve model robustness against adversarial examples, various defense approaches have been proposed, including defensive distillation [20], input transformation [21], robust training [5], and certified defense [22], among which adversarial training has been widely studied and demonstrated to be the most effective [2,5]. Besides, corruption such as snow and blur also frequently occur in the real world, which also presents critical challenges for the building of robust deep learning models. Average-case model performance on small, general, classifier-agnostic corruption can be used to define model corruption robustness.

2.2. Model robustness evaluation

Most proposed defenses conduct incomplete or incorrect evaluations, which are quickly shown to be attacked successfully due to limited understanding of these defenses [15,23,24]. Consequently, conducting rigorous and comprehensive evaluation on model robustness becomes particularly important. To comprehensively evaluate the model robustness for DNNs, a number of works have been proposed. A uniform platform for adversarial robustness analysis named DEEPSEC [8] is proposed to measure the vulnerability of deep learning models. Specifically, the platform incorporates 16 adversarial attacks with 10 attack utility metrics and 13 adversarial defenses with 5 defensive utility metrics. Unlike prior works, Carlini et al. [7] discussed the methodological foundations, reviewed commonly accepted best practices, and suggested new methods for evaluating defenses to adversarial examples. In particular, they provided principles for performing defense evaluations and a specific checklist for avoiding common evaluation pitfalls. Moreover, Ma et al. [9] proposed a set of multi-granularity metrics for deep learning systems, which aims at rendering a multi-faceted portrayal of the testbed (i.e., testing coverage). More recently, Robus-

Table 1
The taxonomy and illustration of the proposed evaluation metrics.

Metrics	Behavior	structure	Adversarial attacks	Corruption attacks	Whitebox	Blackbox	Single model	Multiple models
KMNCov [9]			✓	✓	✓		✓	
NBCov [9]			✓	✓	✓		✓	
SNACov [9]			✓	✓	✓		✓	
ALD$_p$ [8]			✓			✓	✓	
ASS [27]			✓			✓	✓	
PSD [28]			✓			✓	✓	
CA	✓					✓	✓	
AAW	✓		✓		✓		✓	
AAB	✓		✓			✓		✓
ACAC [29]	✓		✓		✓		✓	
ACTC [29]	✓		✓		✓		✓	
NTE [28]	✓		✓		✓		✓	
mCE [30]	✓			✓		✓	✓	
RmCE [30]	✓			✓		✓		✓
mFR [30]	✓			✓		✓		✓
CAV [8]	✓		✓	✓		✓		✓
CRR/CSR [8]	✓		✓	✓		✓		✓
CCV [8]	✓		✓	✓	✓			✓
COS [8]	✓		✓	✓	✓			✓
EBD [6]		✓	✓	✓	✓		✓	
EBD-2		✓	✓		✓		✓	
ENI [6]		✓	✓	✓		✓	✓	
Neuron Sensitivity [31]		✓	✓		✓		✓	
Neuron Uncertainty		✓	✓	✓	✓		✓	

tART [25] and RobustBench [26] have proposed large-scale benchmarks for evaluating adversarial robustness in terms of adversarial attacks, defenses and model structures, but they only utilize accuracy as the evaluation metric.

However, these studies mainly focus on establishing open-source libraries for attacks/defenses, which fail to provide a comprehensive evaluation considering several aspects of a deep learning model towards different noises.

3. Evaluation metrics

To mitigate the problem brought by incomplete evaluation, we establish a multi-view model robustness evaluation framework which consists of 23 evaluation metrics in total. As shown in Table 1, our evaluation metrics can be roughly divided into two parts: data-oriented and model-oriented.

3.1. Data-oriented evaluation metrics

Since model robustness is evaluated based on a set of perturbed examples, the quality of the test data plays a critical role in robustness evaluation. Thus, we use data-oriented metrics considering both neuron coverage and data imperceptibility to measure the integrity of test examples. DeepGauge [9] introduced coverage criteria into neural networks and proposed Neuron Coverage to leverage the output values of neuron and its corresponding boundaries obtained from training data to approximate the major function region and the corner-case region at the neuron level.

3.1.1. Neuron coverage

We first use the coverage criteria for DNNs to measure whether the generated test set could cover enough amount of neurons.

k-Multisection Neuron Coverage (KMNCov). Given a neuron $\mathbf{n}$, the KMNCov measures how thoroughly the given set of the test inputs $\mathbb{D}$ covers the range of neuron output value $[low_{\mathbf{n}}, high_{\mathbf{n}}]$, where $\mathbb{D} = \{x_1, x_2, \ldots\}$ is a set of input data. Specifically, we divide the range $[low_{\mathbf{n}}, high_{\mathbf{n}}]$ into k sections with the same size ($k > 0$), and $S_i^{\mathbf{n}}$ denotes the ith section where $1 \le i \le k$. Let $\phi(x, \mathbf{n})$ denote a function that returns the output of a neuron $\mathbf{n}$ under a given input sample $x \in \mathbb{D}$. We use $\phi(x, \mathbf{n}) \in S_i^{\mathbf{n}}$ to denote that the ith section of neuron n is covered by the input x. For a given test set $\mathbb{D}$ and a specific neuron $\mathbf{n}$, the corresponding k-Multisection Neuron Coverage is defined as the ratio of the sections covered by $\mathbb{D}$ and the overall sections as

$$\mathrm{KMNCov}(\mathbb{D}, k) = \frac{\sum_{\mathbf{n} \in N} |\{S_i^{\mathbf{n}} | \exists x \in \mathbb{D} : \phi(x, \mathbf{n}) \in S_i^{\mathbf{n}}\}}{k \times |N|}, \tag{1}$$

where $N = \{\mathbf{n}_1, \mathbf{n}_2, \ldots\}$ is a set of neurons for the model. It should be noticed that for a neuron n and input x, if $\phi(x, \mathbf{n}) \in [low_{\mathbf{n}}, high_{\mathbf{n}}]$ is satisfied with $\forall \mathbf{n} \in N$, we say that this DNN is located in its major function region. Otherwise, it is located in the corner-case region. In our experiments, we set the number of sections k as 100, and $[low_{\mathbf{n}}, high_{\mathbf{n}}]$ is determined by the activation value of neuron $\mathbf{n}$ on the training set. KMNCov reflects the comprehensiveness of the test set. A qualified test set should have a high value of KMNCov, which means the test set have thoroughly tested the neural network.

Neuron boundary coverage (NBCov) It measures how many corner-case regions have been covered by the given test input set $\mathbb{D}$. Given an input $\mathbf{x} \in \mathbb{D}$, a DNN is located in its corner-case region when given $\mathbf{x}$, $\exists \mathbf{n} \in N : \phi(\mathbf{x}, \mathbf{n}) \in (-\infty, low_{\mathbf{n}}) \cup (high_{\mathbf{n}}, \infty)$. Thus, the NBCov can be defined as the ratio of the covered corner cases and the total corner cased ($2 \times \|N\|$):

$$\frac{|UpperCornerNeuron| + |LowerCornerNeuron|}{2 \times |N|}, \tag{2}$$

where the $UpperCornerNeuron$ is the set consisting of neurons that satisfy $\exists \mathbf{x} \in \mathbb{D} : \phi(\mathbf{x}, n) \in (high_{\mathbf{n}}, +\infty)$. And the $LowerCornerNeuron$ is the set of neurons that satisfy $\exists \mathbf{x} \in \mathbb{D} : \phi(\mathbf{x}, \mathbf{n}) \in (-\infty, low_{\mathbf{n}})$.

Strong neuron activation coverage (SNACov) This metric is designed to measure the coverage status of upper-corner case (i.e., how many corner cases have been covered by the given test sets). It can be described as the ratio of the covered upper-corner cases and the total corner cases ($|N|$):

$$SNACov(\mathbb{D}) = \frac{|UpperCornerNeuron|}{|N|}. \tag{3}$$

According to the coverage criteria, model robustness is closely related to the number of corner cases. High NBCov and SNACov indicate the model meets many unexpected inputs and cannot handle them well.

3.1.2. Data imperceptibility

Here, we introduce several metrics to evaluate data visual imperceptibility by considering the magnitude of perturbations.

Average ℓ_p Distortion (ALD$_p$). Most adversarial attacks generate adversarial examples by constructing additive ℓ_p-norm adversarial perturbations (e.g., $p \in 0, 1, \ldots, \infty$). To measure the visual perceptibility of generated adversarial examples, we use ALD$_p$ as the average normalized ℓ_p distortion:

$$ALD_p = \frac{1}{m}\sum_{i=1}^{m}\frac{||\mathbf{x}_{adv}^{(i)} - \mathbf{x}^{(i)}||_p}{||\mathbf{x}^{(i)}||_p}, \tag{4}$$

where m denotes the number of adversarial examples that attack successfully, and the smaller ALD$_p$ is, the more imperceptible the adversarial example is.

Average structural similarity (ASS) To evaluate the imperceptibility of adversarial examples, we further use SSIM which is considered to be effective to measure human visual perception. SSIM [27] is the most commonly used metric to evaluate the structure similarity between two images. It separates the task of similarity measurement into three comparisons: luminance, contrast, and structure as

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{5}$$

where μ_x and μ_y represent the means of input image x and y, σ_x^2 and σ_y^2 stand for the variances of x and y, σ_{xy} is the covariance of x and y, c_1 and c_2 are constants. In our experiments, we set $c_1 = 0.01$ and $c_2 = 0.03$. Thus, ASS can be defined as the average of SSIM between all adversarial examples and the corresponding clean examples, i.e.,

$$ASS = \frac{1}{m}\sum_{i=1}^{m} SSIM(\mathbf{x}_{adv}^{(i)}, \mathbf{x}^{(i)}), \tag{6}$$

where m denotes the number of successful adversarial examples, and the higher ASS is, the more imperceptible the adversarial example is.

Perturbation sensitivity distance (PSD) Based on the contrast masking theory [32], PSD is proposed to evaluate human perception of perturbations. Thus, PSD is defined as:

$$PSD = \frac{1}{m}\sum_{i=1}^{m}\sum_{i=1}^{t}\delta_{(j)}^{(i)} Sen\Big(R\big(\mathbf{x}_{(j)}^{(i)}\big)\Big), \tag{7}$$

where m denotes the number of adversarial examples that attacks successfully, t is the total number of pixels, $x_{(j)}^{(i)}$ represents the jth pixel of the ith example, and $\delta_{(j)}^{(i)}$ represents the perturbations added at the specific pixel. $R\big(\mathbf{x}_{(j)}^{(i)}\big)$ stands for the square surrounding region of $\mathbf{x}_{(j)}^{(i)}$, and $Sen\Big(R\big(\mathbf{x}_{(j)}^{(i)}\big)\Big) = 1/std\Big(R\big(\mathbf{x}_{(j)}^{(i)}\big)\Big)$. According to contrast masking theory in image processing [32], human eyes are more sensitive to perturbations on pixels in low variance regions than those in high variance regions. Therefore, to make adversarial examples imperceptible, we should perturb pixels at high variance zones rather than low variance ones. Evidently, the smaller PSD is, the more imperceptible the adversarial example is.

3.2. Model-oriented evaluation metrics

To evaluate robustness, the most intuitive direction is to measure the model performance in the adversarial setting. Given an adversary $\mathcal{A}_{\epsilon,p}$, it uses specific attack methods to generates adversarial examples $\mathbf{x}_{adv} = \mathcal{A}_{\epsilon,p}(\mathbf{x})$ for a clean example $\mathbf{x}$ with the perturbation magnitude ϵ under ℓ_p norm.

3.2.1. Model behaviors

- **Task Performance**

Clean accuracy (CA) Model accuracy on clean examples is one of the most important properties in the adversarial setting. CA is defined as the percentage of clean examples that are successfully classified by a classifier f into the ground truth classes as follows

$$CA(f, \mathbb{D}) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}(f(\mathbf{x}_i) = \mathbf{y}_i), \tag{8}$$

where $\mathbb{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1\ldots n}$ is the test set, $\mathbf{1}(\cdot)$ is the indicator function.

- **Adversarial Performance**

Adversarial accuracy on white-box attacks (AAW) In the untargeted attack scenario, AAW is defined as the percentage of adversarial examples generated in the white-box setting that are successfully misclassified into an arbitrary class except for the ground truth class; for targeted attack, it can be measured by the percentage of adversarial examples generated in the white-box setting classified as the target class. In the rest of the paper, we mainly focus on untargeted attacks. Thus, AAW can be defined as:

$$AAW(f, \mathbb{D}, \mathcal{A}_{\epsilon,p}) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}(f(\mathcal{A}_{\epsilon,p}(\mathbf{x}_i)) \neq \mathbf{y}_i). \tag{9}$$

Adversarial accuracy on black-box attacks (AAB) Similar to AAW, AAB is defined by the percentage of black-box adversarial examples classified correctly by the classifier.

Average confidence of adversarial class (ACAC) Besides prediction accuracy, prediction confidence on adversarial examples gives further indications of model robustness. Thus, for an adversarial example, ACAC can be defined as the average prediction confidence towards the incorrect class

$$ACAC(f, \mathbb{D}, \mathcal{A}_{\epsilon,p}) = \frac{1}{m}\sum_{i=1}^{m}P(\mathcal{A}_{\epsilon,p}(\mathbf{x}_i)), \tag{10}$$

where m is the number of adversarial examples that attack successfully, P is the prediction confidence of classifier f towards the wrong classes.

Average confidence of true class (ACTC) Meanwhile, we also use ACTC to further evaluate to what extent the attacks escape from the ground truth. In other words, ACTC can be defined as the average model prediction confidence on adversarial examples towards the ground truth labels, i.e.,

$$ACTC(f, \mathbb{D}, \mathcal{A}_{\epsilon,p}) = \frac{1}{m}\sum_{i=1}^{m}P_{\mathbf{y}_i}(\mathcal{A}_{\epsilon,p}(\mathbf{x}_i)), \tag{11}$$

where m is the number of adversarial examples that attack successfully, $P_{\mathbf{y}_i}$ is the prediction confidence of classifier f towards the ground truth class y_i.

Noise tolerance estimation (NTE) Moreover, given the generated adversarial examples, we further calculate the gap between the probability of misclassified class and the max probability of all other classes, which measures the tolerance of model against adversarial noises.

$$NTE(f, \mathbb{D}, \mathcal{A}_{\epsilon,p}) = \frac{1}{m}\sum_{i=1}^{m}[P_{f(\mathbf{x}_i)}(\mathcal{A}_{\epsilon,p}(\mathbf{x}_i)) - \max P_j(\mathcal{A}_{\epsilon,p}(\mathbf{x}_i))], \tag{12}$$

where $j \in 1, \ldots, k$ and $j \neq f(\mathbf{x}_i)$, and m is the number of adversarial examples that attack successfully. As the NTE becomes increasingly high, the model becomes more vulnerable towards adversarial examples.

- **Corruption Performance**

To further comprehensively measure the model robustness against different corruption, we introduce evaluation metrics following [30].

mCE. This metric denotes the mean corruption error of a model compared to the baseline model [30]. Different from the original paper, we simply calculate the error rate of the classifier f on each corruption type c at each level of severity s denoted as $E_{s,c}^{f}$ and compute mCE as follows:

$$mCE_c^f = \frac{1}{t}\sum_{s=1}^{t} E_{s,c}^{f}, \tag{13}$$

where t denotes the number of severity levels. Thus, mCE is the average value of Corruption Errors (CE) using different corruption.

Relative mCE A more nuanced corruption robustness measure is Relative mCE (RmCE) [30]. If a classifier withstands most corruption, the gap between mCE and the clean data error is minuscule. So, RmCE is

$$RmCE_c^f = \frac{1}{t}\sum_{s=1}^{t} E_{s,c}^{f} - E_{clean}^{f}, \tag{14}$$

where E_{clean}^{f} is the error rate of f on clean examples.

mFR Hendrycks et al. [30] introduce mFR to represent the classification differences between two adjacent frames in the noise sequence for a specific image. Let us denote q noise sequences with $S = \{(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \dots, \mathbf{x}_i^{(n)})\}_{i=1}^{q}$ where each sequence is created with a specific noise type c as

$$FP_c^f = \frac{1}{q(n-1)}\sum_{i=1}^{q} sum_{j=2}^{n}\mathbf{1}(f(\mathbf{x}_i^{(j)}) \neq f(\mathbf{x}_i^{(j-1)})). \tag{15}$$

Then, the Flip Rate (FR) can be obtained by $FR_c^f = FP_c^f / FP_c^{base}$ and mFR is the average value of FR. Different from the original paper, we set FP_c^{base} to be 1, which means $FR_c^f = FP_c^f$.

- **Defense Performance**

We further explore to what extent the model performance has been influenced when defense strategies are added.

CAV Classification Accuracy Variance (CAV) is used to evaluate the impact of defenses based on the accuracy. We expect the defense-enhanced model f^d to maintain the classification accuracy on normal testing examples as much as possible. Therefore, it is defined as follows:

$$CAV = ACC(f^d, D) - ACC(f, D), \tag{16}$$

where $ACC(f, D)$ denotes model f accuracy on dataset D.

CRR/CSR CRR is the percentage of testing examples that are misclassified by f previously but correctly classified by f^d. Inversely, CSR is the percentage of testing examples that are correctly classified by f but misclassified by f^d. Thus, they are defined as follows:

$$CRR = \frac{1}{n}\sum_{i=1}^{n} count((f(\mathbf{x}_i) \neq \mathbf{y}_i) \& (f^d(\mathbf{x}_i) = \mathbf{y}_i)), \tag{17}$$

$$CSR = \frac{1}{n}\sum_{i=1}^{n} count((f(\mathbf{x}_i) = \mathbf{y}_i) \& (f^d(\mathbf{x}_i) \neq \mathbf{y}_i)), \tag{18}$$

where n is the number of examples.

CCV Defense strategies may not have negative influences on the accuracy performance, however, the prediction confidence of correctly classified examples may decrease. Classification Confidence Variance (CCV) can measure the confidence variance induced by robust models:

$$CCV = \frac{1}{m}\sum_{i=1}^{m} |P_{\mathbf{y}_i}(\mathbf{x}_i) - P_{\mathbf{y}_i}^{d}(\mathbf{x}_i)|, \tag{19}$$

where $P_{\mathbf{y}_i}(\mathbf{x}_i)$ denotes the prediction confidence of model f towards $\mathbf{y}_i$ and m is the number of examples correctly classified by both f and f^d.

COS Classification Output Stability (COS) uses JS divergence to measure the similarity of the classification output stability between the original model and the robust model as:

$$COS = \frac{1}{m}\sum_{i=1}^{m} JSD(P(\mathbf{x}_i) \| P^d(\mathbf{x}_i)), \tag{20}$$

where $P(\mathbf{x}_i)$ and $P^d(\mathbf{x}_i)$ denotes the prediction confidence of model f and f^d on $\mathbf{x}_i$, respectively. m is the number of examples correctly classified by both f and f^d. JSD stands for the JS divergence, which is a commonly used method of measuring the similarity between two probability distributions.

3.2.2. Model structures

- **Boundary-based**

Empirical boundary distance (EBD) The minimum distance to the decision boundary among data points reflects the model robustness to small noises. EBD calculates the minimum distance to the model decision boundary in a heuristic way. A larger EBD value means a stronger model. Given a learnt model f and point $\mathbf{x}_i$ with class label $\mathbf{y}_i$ $(i = 1, \dots, k)$, it first generates a set V of m random orthogonal directions [33]. Then, for each direction in V it estimates the root mean square (RMS) distances $\phi_i(V)$ to the decision boundary of f, until the model's prediction changes, i.e., $f(\mathbf{x}_i) \neq \mathbf{y}_i$. Among $\phi_i(V)$, d_i denotes the minimum distance moved to change the prediction for instance $\mathbf{x}_i$. Then, EBD is defined as follows:

$$EBD = \frac{1}{n}\sum_{i=1}^{n} d_i, \quad d_i = \min \phi_i(V), \tag{21}$$

where n denotes the number of instances used.

Empirical boundary distance-2 (EBD-2) Additionally, we introduce the evaluation metrics EBD-2, which calculates the minimum distance of the model decision boundary for each class. Given a learnt model f and dataset $\mathbb{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1\dots n}$, for each direction j in the k classes, the metric estimates the distances d_j to change the model prediction of $\mathbf{x}^{(i)}$, i.e., $f(\mathbf{x}^{(i)}) \neq \mathbf{y}^{(i)}$. Specifically, we use iterative adversarial attacks (e.g., BIM) in practice and calculate the steps used as the distance d_j.

- **Consistency-based**

ε-Empirical noise insensitivity Xu and Mannor [34] first introduced the concept of learning algorithms robustness from the idea that if two samples are "similar" then their test errors are very close. ε-Empirical Noise Insensitivity measures the model robustness against noise from the view of Lipschitz constant, and a lower value indicates a stronger model. We first select n clean examples randomly, then m examples are generated from each clean example via various methods, e.g., adversarial attack, Gaussian noise, blur, etc. The differences between model loss function are computed when clean example and corresponding polluted examples are fed to. The different severities in loss function is used to measure model insensitivity and stability to generalized small noise within constraint ε:

$$I_f(\varepsilon) = \frac{1}{n \times m}\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{|l_f(\mathbf{x}_i|\mathbf{y}_i) - l_f(\mu_{ij}|\mathbf{y}_i)|}{|\mathbf{x}_i - \mu_{ij}|_\infty}$$
$$\text{s.t.} \quad |\mathbf{x}_i - \mu_{ij}|_\infty \leq \varepsilon, \tag{22}$$

where $\mathbf{x}_i$, μ_{ij} and $\mathbf{y}_i$ denote the clean example, corresponding polluted example and the class label, respectively.

- **Neuron-based**

Neuron sensitivity Intuitively, for a model that owns strong robustness, namely, insensitive to adversarial examples, the clean example $\mathbf{x}$ and the corresponding adversarial example $\mathbf{x}'$ share a similar representation in the hidden layers of the model [34]. Neuron Sensitivity can be deemed as the deviation of the feature representation in hidden layers between clean examples and corresponding adversarial examples, which measures model robustness from the perspective of neuron. Specifically, given a benign example $\mathbf{x}_i$, where $i = 1, \ldots, n$, from $\mathbb{D}$ and its corresponding adversarial example $\mathbf{x}'_i$ from $\mathbb{D}'$, we can get the dual pair set $\bar{\mathbb{D}} = \{(\mathbf{x}_i, \mathbf{x}'_i)\}$, and then calculate the neuron sensitivity σ as follows:

$$\sigma(f_{l,m}, \bar{\mathbb{D}}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{dim(f_{l,m}(\mathbf{x}_i))} \| f_{l,m}(\mathbf{x}_i) - f_{l,m}(\mathbf{x}'_i) \|_1, \quad (23)$$

where $f_{l,m}(\mathbf{x}_i)$ and $f_l^m(\mathbf{x}'_i)$ respectively represents outputs of the mth neuron at the lth layer of f towards clean example $\mathbf{x}_i$ and corresponding adversarial example $\mathbf{x}'_i$ during the forward process. $dim(\cdot)$ denotes the dimension.

Neuron uncertainty Model uncertainty has been widely investigated in safety critical applications to induce the confidence and uncertainty behaviors during model prediction. Motivated by the fact that model uncertainty is commonly induced by predictive variance, we use the variance of neuron $f_{l,m}$ to calculate the Neuron Uncertainty as:

$$U(f_{l,m}) = \frac{1}{n}\sum_{i=1}^{n} variance(f_{l,m}(\mathbf{x}^{(i)})). \quad (24)$$

4. Experiments

Here, we evaluate model robustness using our proposed evaluation framework on image classification benchmarks CIFAR-10, SVHN, and ImageNet. The experimental results of VGG-16 on SVHN and CIFAR-10 on WideResNet-28 can be found in supplementary materials.

4.1. Model-oriented evaluation

4.1.1. Model behaviors

As for adversarial robustness, we report metrics including CA, AAW, AAB, ACAC, ACTC, and NTE. The experimental results regarding CA and AAW can be found in Table 2; the results of AAB are shown in Table 3; and the results in terms of ACAC, ACTC, and NTE are listed in supplementary material. Besides standard black-box attacks (NAttack, SPSA, and BA), we also generate adversarial examples using an Inception-V3 then perform transfer attacks on the target model (denoted “PGD-ℓ_1”, “PGD-ℓ_2”, and “PGD-ℓ_∞” in Table 3).

As for corruption robustness, the results of mCE, relative mCE, and mFR can be found in Fig. 2. Moreover, the results of CAV, CRR/CSR, CCV, and COS are illustrated in supplementary materials.

From the above experimental results, we can draw several conclusions as follows: (1) for small datasets like CIFAR-10 and SVHN, TRADES achieves the highest adversarial robustness for almost all adversarial attacks in both black-box and white-box settings, but it is vulnerable against corruptions; however, current defenses (especially adversarial training) are still suffering problems when scaling to large-scale datasets like ImageNet (e.g., TRADES fails to perform well on clean and adversarial attacks); (2) models trained on one specific perturbation type are vulnerable to other norm-bounded perturbations (e.g., ℓ_∞ trained models are weak towards ℓ_1 and ℓ_2 adversarial examples); (3) standard adversarially-trained models (SAT and PAT) are still vulnerable from a more rigorous perspective by showing high confidence of adversarial classes and low confidence of true classes; (4) transfer-based black-box attacks may fail to serve as a good indicator for showing robustness especially in large-scale dataset (see Table 3(b)); and (5) as the model becomes increasingly robust, its COS and CCV values become comparatively high, which in turn indicates that the model is less stable towards clean examples. The reason might be there exists trade-off between standard accuracy and robust accuracy, thus models behaving high accuracy against adversarial attacks (i.e., TRADES) show low stability towards clean examples.

4.1.2. Model structures

We then evaluate model robustness with respect to structures. The results of EBD and EBD-2 are illustrated in Table 4 and Fig. 3; the results of ϵ-Empirical Noise Insensitivity can be found in supplementary material; Neuron Sensitivity and Neuron Uncertainty can be found in Fig. 4, respectively.

In summary, we can draw several interesting observations: *(1) in most cases, models with higher adversarial accuracy are showing better structure robustness; (2) though showing the highest adversarial accuracy, TRADES does not have the largest EBD value as shown in Table 4.*

4.2. Data-oriented evaluation

We then report the data-oriented evaluation metrics. We randomly sample 1000 images from test set for CIFAR-10 and SVHN, and 5000 images for ImageNet; we then adversarially perturb these images using FGSM and PGD, respectively. We finally compute and report the neuron-coverage related metrics (KMNCov, NBCov, SNACov) using these test sets. The results can be found in Tables 5–7. Further, we show the results of ALD$_p$, ASS, and PSD on these test sets in Tables 8–10.

In summary, we can draw conclusions as follows: *(1) adversarial examples generated by ℓ_∞-norm attacks show significantly higher neuron coverage than other perturbation types (e.g., ℓ_1 and ℓ_2), which indicate that ℓ_∞-norm attacks cover more “paths” for a DNN when perform test or evaluation; (2) meanwhile, ℓ_∞-norm attacks are more imperceptible to the human vision (lower ALD$_p$, PSD, and higher ASS values compared to ℓ_1 and ℓ_2 attacks).*

5. Discussions and suggestions

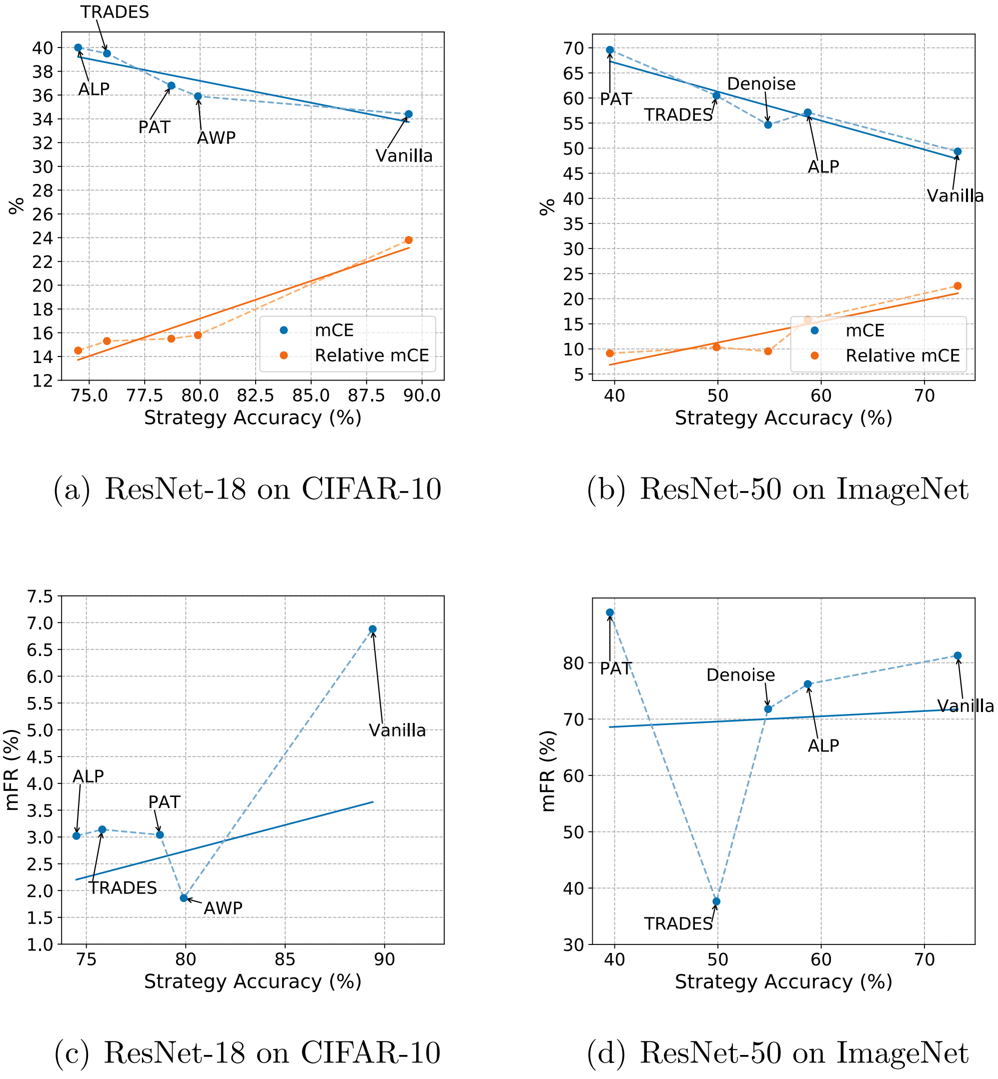
Having demonstrated extensive experiments on these datasets using our comprehensive evaluation framework, we now take a

Table 2
White-box adversarial attacks (%) on CIFAR-10 using ResNet-18 and ImageNet using ResNet-50.

	(a) ResNet-18 on CIFAR-10							(b) ResNet-50 on ImageNet				
	Clean	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞	AA	C&W		Clean	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞	AA
Vanilla	89.4	51.5	0.9	0.0	0.0	0.4	Vanilla	82.3	0.0	0.0	0.0	0.0
PAT	78.7	71.7	52.3	35.1	34.5	52.1	PAT	74.2	1.1	0.7	5.5	0.2
TRADES	75.8	66.8	50.1	36.7	35.5	49.7	TRADES	40.6	7.9	0.3	4.4	0.1
ALP	74.5	68.0	50.9	39.8	37.7	49.8	ALP	56.2	1.2	0.1	0.4	0.0
AWP	79.9	75.8	63.5	55.5	50.3	59.7	Denoise	62.0	5.6	1.4	6.8	1.5

Table 3
Black-box adversarial attacks (%) on CIFAR-10 using ResNet-18 and ImageNet using ResNet-50.

	(a) ResNet-18 on CIFAR-10							(b) ResNet-50 on ImageNet					
	NAttack	SPSA	BA	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞		NAttack	SPSA	BA	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞
Vanilla	0.0	8.8	0.0	81.2	53.0	20.9	Vanilla	0.1	19.7	0.9	76.8	69.3	70.6
PAT	35.8	71.9	0.0	78.6	78.2	77.1	PAT	14.8	25.6	0.3	70.5	26.7	26.6
TRADES	36.7	71.8	0.0	75.3	74.6	73.9	TRADES	8.3	42.7	1.3	39.3	37.6	37.8
ALP	38.4	70.4	0.0	74.1	74.0	73.4	ALP	21.6	34.7	0.2	58.6	56.4	55.7
AWP	52.1	74.9	0.0	80.0	79.2	78.2	Denoise	29.6	61.7	0.7	63.1	62.1	63.1



(a) ResNet-18 on CIFAR-10

(b) ResNet-50 on ImageNet

(c) ResNet-18 on CIFAR-10

(d) ResNet-50 on ImageNet

Fig. 2. Experimental results of mCE, RmCE, and mFR on CIFAR-10 and ImageNet corruption dataset.

Table 4
Experiments results of EBD (measured as RMS distance) and EBD-2 (measured as number of iterations with $\alpha = 0.0005$) on CIFAR-10 using ResNet-18 and ImageNet using ResNet-50.

	(a) ResNet-18 on CIFAR-10						(b) ResNet-50 on ImageNet				
Model	Vanilla	PAT	TRADES	ALP	AWP	Model	Vanilla	PAT	SAT	TRADES	ALP
EBD	3.5	9.3	9.0	9.1	10.0	EBD	35.2	18.1	20.0	26.1	45.2
EBD-2	8.6	46.9	46.4	48.3	61.0	EBD-2	2.2	3.9	19.7	19.9	3.0

Table 5
Experiments results of KMNCov on CIFAR-10 using ResNet-18 and ImageNet using ResNet-50.

	(a) ResNet-18 on CIFAR-10							(b) ResNet-50 on ImageNet					
	FGSM	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞	NAttack	SPSA		FGSM	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞	NAttack	SPSA
Vanilla	89.1	90.2	92.0	93.2	88.9	88.3	Vanilla	56.7	82.6	85.9	86.6	58.5	59.3
PAT	90.5	91.0	90.6	90.3	90.3	90.7	PAT	47.0	69.2	47.7	47.4	52.5	47.2
TRADES	91.1	91.2	91.0	90.9	90.9	91.0	TRADES	57.0	60.2	68.1	66.7	57.7	57.8
ALP	89.7	90.9	90.0	89.6	89.7	90.5	ALP	55.3	64.0	68.0	66.1	55.4	55.9
AWP	83.0	87.0	84.1	83.2	83.6	86.2	Denoise	58.9	60.5	64.2	59.7	33.0	59.2

Table 6
Experiments results of NBCov on CIFAR-10 using ResNet-18 and ImageNet using ResNet-50.

	(a) ResNet-18 on CIFAR-10							(b) ResNet-50 on ImageNet					
	FGSM	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞	NAttack	SPSA		FGSM	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞	NAttack	SPSA
Vanilla	9.4	7.1	15.4	27.8	6.9	7.7	Vanilla	58.8	91.3	88.4	90.6	57.0	60.6
PAT	7.5	7.0	7.0	7.2	7.1	7.0	PAT	54.9	92.8	56.2	57.4	58.3	54.5
TRADES	7.7	7.3	7.3	7.4	7.4	6.9	TRADES	57.7	81.4	84.4	83.0	52.6	55.2
ALP	6.9	6.8	6.8	6.8	6.9	6.5	ALP	65.2	90.2	88.4	90.1	55.1	62.0
AWP	6.1	6.2	6.2	6.2	6.2	6.1	Denoise	45.7	66.7	82.1	58.8	3.6	47.8

Table 7
Experiments results of SNACov on CIFAR-10 using ResNet-18 and ImageNet using ResNet-50.

	(a) ResNet-18 on CIFAR-10							(b) ResNet-50 on ImageNet					
	FGSM	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞	NAttack	SPSA		FGSM	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞	NAttack	SPSA
Vanilla	4.7	2.2	12.6	27.3	2.0	3.0	Vanilla	29.3	83.0	78.1	82.0	23.8	30.7
PAT	2.8	2.2	2.3	2.5	2.2	2.3	PAT	27.4	86.6	29.7	30.2	28.2	26.9
TRADES	3.1	2.5	2.5	2.7	2.4	2.2	TRADES	27.7	68.6	76.0	72.0	10.2	24.6
ALP	2.3	2.0	2.1	2.1	1.9	1.9	ALP	40.0	81.8	78.7	81.7	17.2	33.1
AWP	1.0	1.0	1.2	1.1	1.0	1.1	Denoise	41.5	61.6	75.1	55.1	1.4	47.1

Table 8
Experiments results of ALD$_p$(%) on CIFAR-10 using ResNet-18 and ImageNet using ResNet-50.

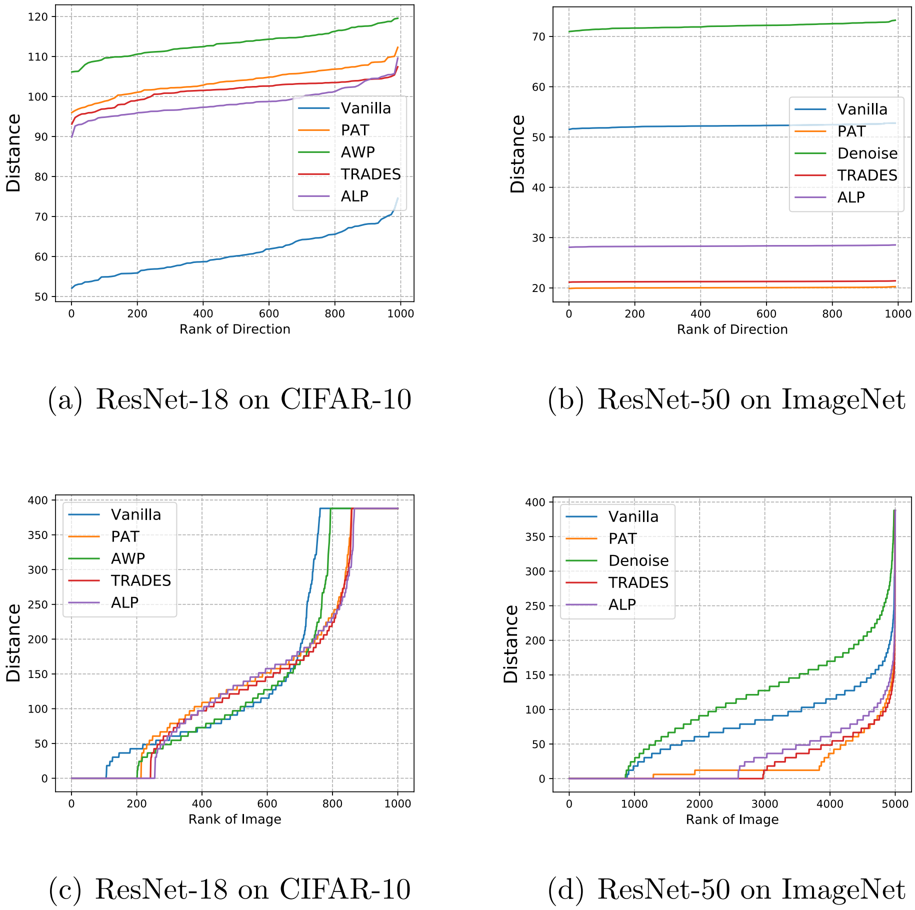
	(a) ResNet-18 on CIFAR-10					(b) ResNet-50 on ImageNet			
	FGSM	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞		FGSM	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞
Vanilla	6.5	0.7	1.9	5.4	Vanilla	6.2	2.0	8.8	7.5
PAT	6.6	0.7	1.9	6.1	PAT	6.2	2.2	8.8	10.0
TRADES	6.5	0.7	1.9	6.1	TRADES	6.2	3.8	8.8	10.1
ALP	6.6	0.7	1.9	6.1	ALP	6.3	3.7	8.8	9.0
AWP	6.5	0.7	1.9	6.0	Denoise	6.3	3.8	8.8	10.2

Table 9
Experiments results of ASS(%) on CIFAR-10 using ResNet-18 and ImageNet using ResNet-50.

	(a) ResNet-18 on CIFAR-10					(b) ResNet-50 on ImageNet			
	FGSM	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞		FGSM	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞
Vanilla	93.0	99.9	99.1	95.2	Vanilla	82.1	97.3	77.9	77.5
PAT	95.5	100.0	99.7	96.1	PAT	82.0	97.4	83.0	82.0
TRADES	95.5	100.0	99.8	96.2	TRADES	85.5	97.4	82.3	77.4
ALP	95.4	100.0	99.8	96.1	ALP	82.7	97.1	79.6	75.9
AWP	95.4	100.0	99.7	96.1	Denoise	87.3	98.0	81.3	78.6

Table 10
Experiments results of PSD(%) on CIFAR-10 using ResNet-18 and ImageNet using ResNet-50.

	(a) ResNet-18 on CIFAR-10					(b) ResNet-50 on ImageNet			
	FGSM	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞		FGSM	PGD-ℓ_1	PGD-ℓ_2	PGD-ℓ_∞
Vanilla	70.9	2.9	22.7	59.1	Vanilla	127.0	17.0	90.0	69.4
PAT	159.5	1.1	11.7	153.8	PAT	171.0	16.6	88.0	227.1
TRADES	153.5	1.2	11.6	139.9	TRADES	266.5	4.9	86.6	186.5
ALP	176.0	1.2	11.8	155.6	ALP	109.2	7.6	91.0	106.6
AWP	201.4	1.2	10.1	175.8	Denoise	326.0	4.4	86.5	226.4



(a) ResNet-18 on CIFAR-10

(b) ResNet-50 on ImageNet

(c) ResNet-18 on CIFAR-10

(d) ResNet-50 on ImageNet

Fig. 3. Specific distance values on CIFAR10 and ImageNet related to EBD: (a)(b)(c) the average distance moved in each orthogonal direction, and (d)(e)(f) the Empirical Boundary Distance moved for 1000 different images.

further step and provide additional suggestions to the evaluation of model robustness as well as the design of adversarial attacks/defenses in the future.

5.1. Evaluate model robustness using more attacks

For most studies in the adversarial learning literature [6,21,35], they evaluate model robustness primarily on ℓ_∞-norm bounded PGD attacks, which has been shown to be the most effective and representative adversarial attack. However, according to our experimental results, we suggest to provide more comprehensive evaluations on different types of attacks:

(1) *Evaluate model robustness on ℓ_p-norm bounded adversarial attacks*. However, as shown in Tables 2 and 3, most adversarial defenses are designed to counteract a single type of perturbation (e.g., small ℓ_∞-noise) and offer no guarantees for other perturbations (e.g., ℓ_1, ℓ_2), sometimes even increase model vulnerability [36]. Thus, to fully evaluate adversarial robustness, we suggest to use ℓ_1, ℓ_2, and ℓ_∞ attacks.

(2) *Evaluate model robustness on adversarial attacks as well as corruption attacks*. In addition to adversarial examples, corruption such as snow and blur also frequently occur in the real world, which also presents critical challenges for the building of strong deep learning models. According to our studies, deep learning models behave distinctly subhuman to input images with different corruption. Meanwhile, adversarially robust models may also vulnerable to corruption as shown in Fig. 2. Therefore, we suggest to take both adversarial robustness and corruption robustness into consideration, when measuring the model robustness against noises.

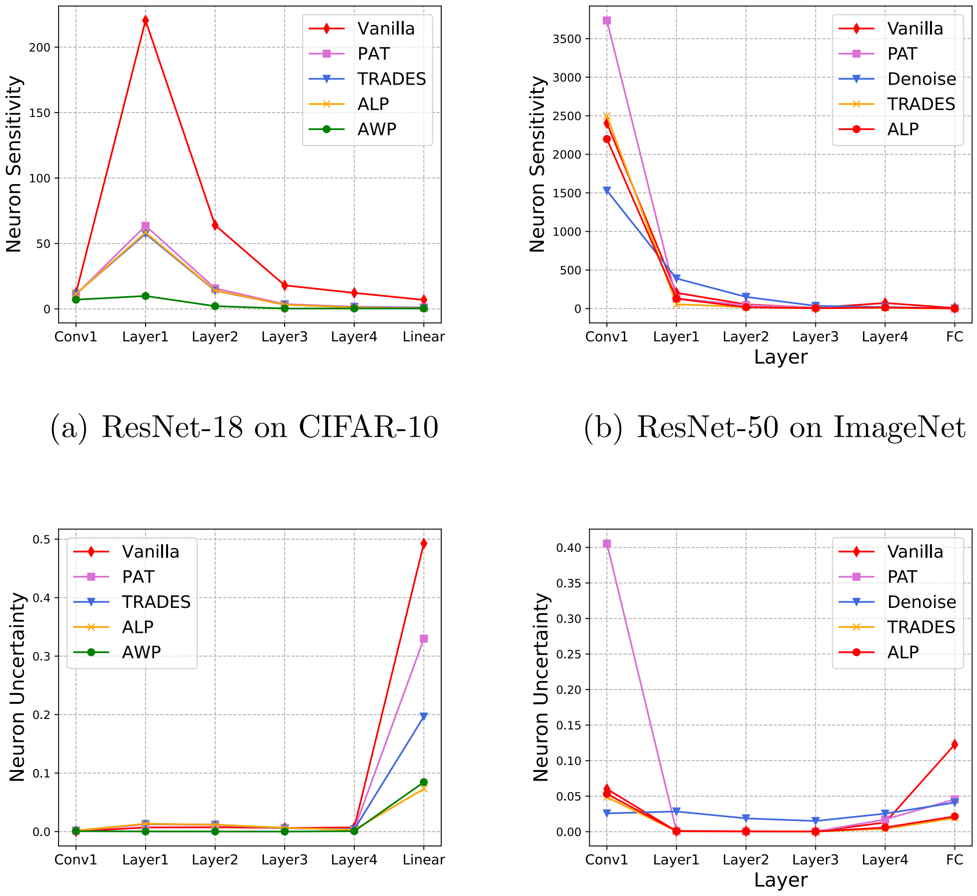
(3) *Perform black-box or gradient-free adversarial attacks.* Black-box attacks are effective to elaborating whether obfuscated gradients [24] have been introduced to a specific defense. Moreover, black-box attacks are also shown to cover more neurons when perform test as shown in Table 5. However, it seems that transfer-based black-box attacks are not a good indicator for robustness evaluation, especially there exists huge-differences between source and targets models or testing on large datasets.

5.2. Evaluate model robustness considering multiple views

To mitigate the problem brought by incomplete evaluations, we suggest to evaluate model robustness using more rigorous metrics, which consider multi-view robustness.

(1) *Consider model behaviors with respect to more profound metrics, e.g., prediction confidence.* For example, though showing high adversarial accuracy, SAT are vulnerable by showing high confidence of adversarial classes and low confidence of true classes, which are similar to vanilla models.

(2) *Evaluate model robustness in terms of model structures, e.g., boundary distance.* For example, though ranking high among other baselines on adversarial accuracy, TRADES is not strong enough in terms of Neuron Sensitivity, EBD compared to other baselines.



(a) ResNet-18 on CIFAR-10

(b) ResNet-50 on ImageNet

(c) ResNet-18 on CIFAR-10

(d) ResNet-50 on ImageNet

Fig. 4. Experimental results of neuron sensitivity and neuron uncertainty on CIFAR-10 using ResNet-18 and ImageNet using ResNet-50, with PGD-ℓ_∞ adversarial examples. We report the mean value of the metrics for each layer.

5.3. Design of attacks and defenses

Besides model robustness evaluation, the proposed metrics are also beneficial to the design of adversarial attacks and defenses. Most of these metrics provide deep investigations of the model behaviors or structures towards noises, which can be used for researchers to design adversarial attack or defense methods. Regarding metrics in terms of model structures, we can develop new attacks or defenses by either enhancing or impairing them, since these metrics capture the structural pattern that manifests model robustness.

5.4. Selection of proposed metrics

Our proposed metrics reflect the model robustness from different perspectives. According to the specific scenario, we should select the proper metric to measure the model robustness in order to improve it. For instance, CAV, CRR, CSR, COS and CCV should be applied to compare different defensive methods, and EBD should be applied to evaluate robustness of random noises. When it comes to model structure adjustment, we should examine the neuron sensitivity and neuron uncertainty. Proper metrics can depict the model robustness more clearly in suitable scenarios, and can provide more insight that is helpful for model robustness enhancement.

In conclusion, for comprehensive robustness evaluation, we suggest authors to: (1) evaluate robustness towards different noises (adversarial noises and common corruption); (2) use both white-box and black-box attacks for evaluating adversarial robustness; and (3) evaluate robustness from different perspectives including behaviors and structures for deeper analyses.

6. An open-Sourced platform

To fully support our multi-view evaluation and facilitate further research, we provide an open-sourced platform[1] based on Pytorch. The framework of our platform is illustrated in Fig. 5. Our platform contains several highlights as follows:

(1) *Multi-language environment.* To facilitate the user flexibility, we support the use of language-independent models (e.g., Java, C, Python, etc.). To achieve the goal, we establish standardized input and output systems with a uniform format with the help of Docker.
(2) *High extendibility*. Our platform also supports continuous integration of user-specific algorithms and models. In other words, users are able to introduce externally personal-designed attack, defense, evaluation methods, by simply inheriting the base classes through several public interfaces.

[1] https://git.openi.org.cn/OpenI/AISafety.

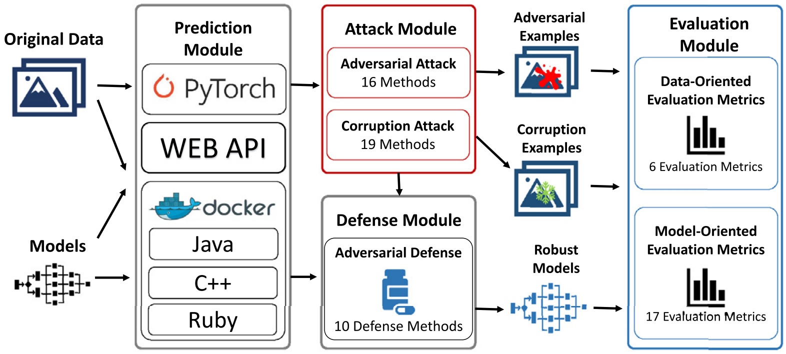


Fig. 5. The framework of our open-source platform, which primarily consists of Attack module, Defense module, Evaluation module, Prediction module, and Database module.

(3) *Multiple scenarios*. Our platform integrates multiple real-world application scenarios, e.g., auto-driving, automatic check-out, interactive robots.

Our platform enjoys several advantages as follows: (1) Attacks and defenses. Our platform contains 15 adversarial attacks, 19 corruption attacks, and 10 adversarial defenses. (2) Robustness evaluation. We have 23 different evaluation metrics. (3) Static/dynamic analysis. Our platform is the only platform that could perform static and dynamic analysis. (4) Competition. Our platform could further enable users to organize or participate in competitions on our open-source platform using the embedded interfaces. (5) Multiple scenarios. Our platform could evaluate model robustness for several real-world scenarios, e.g., automatic-checkout and auto-driving using the sandbox inside. Prevailing platforms mainly focus on attack and defense algorithms, while not able to provide comprehensive robustness evaluations. Adversarial toolboxes, e.g., Foolbox [37] and Cleverhans [38], only have (1), and test platforms, e.g., DeepXplore [39], have (1) and (2).

7. Conclusion

In this work, we establish a model robustness evaluation framework containing 23 comprehensive and rigorous metrics, which consider two key perspectives of adversarial learning (i.e., data and model). Moreover, we provide an open-sourced model robustness evaluation platform providing multiple views of model robustness with the help of the metric framework, which supports continuous integration of user-specific algorithms and language-independent models. To fully demonstrate the effectiveness of our framework, we conduct large-scale experiments on multiple datasets using different models and defenses with our open-source platform. The experimental results show the limit of prevailing defense methods, as they behave differently on our proposed metrics. Finally, we provide discussions and suggestions according to our experiment, hoping to shed light on model robustness. The objective of this work is to provide a comprehensive evaluation framework which could conduct more rigorous evaluations of model robustness. We hope our paper can facilitate fellow researchers for a better understanding of the adversarial examples as well as further improvement of model robustness.

Declaration of Competing Interest

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

Data availability

The authors do not have permission to share data.

Acknowledgements

The authors would like to thank the editors and the anonymous referees for their helpful comments to improve the quality of the paper. This work was supported by The National Key Research and Development Plan of China under Grant 2020AAA0103502, National Natural Science Foundation of China under Grant 62022009 and 62206009, and State Key Lab of Software Development Environment under Grant SKLSDE-2022ZX-02.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.patcog.2023.109308.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I.J. Goodfellow, R. Fergus, Intriguing properties of neural networks, ICLR, 2014.
[2] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, ICLR, 2015.
[3] A. Athalye, L. Engstrom, A. Ilyas, K. Kwok, Synthesizing robust adversarial examples, in: International Conference on Machine Learning, PMLR, 2018, pp. 284–293.
[4] T. Dai, Y. Feng, B. Chen, J. Lu, S.-T. Xia, Deep image prior based defense against adversarial examples, Pattern Recognit 122 (2022) 107309.
[5] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, ICLR, 2018.
[6] A. Liu, X. Liu, C. Zhang, H. Yu, Q. Liu, D. Tao, Training robust deep neural networks via adversarial noise propagation, IEEE TIP 30 (2021) 5769–5781.
[7] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, On evaluating adversarial robustness, arXiv preprint arXiv:1902.06705(2019).
[8] X. Ling, S. Ji, J. Zou, J. Wang, C. Wu, B. Li, T. Wang, DeepSec: a uniform platform for security analysis of deep learning model, IEEE S&P, 2019.
[9] L. Ma, F. Juefei-Xu, F. Zhang, J. Sun, M. Xue, B. Li, C. Chen, T. Su, L. Li, Y. Liu, et al., DeepGauge: multi-granularity testing criteria for deep learning systems, ACM/IEEE ASE, 2018.
[10] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).
[11] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A.Y. Ng, Reading digits in natural images with unsupervised feature learning (2011).
[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, CVPR, 2009.
[13] A. Liu, T. Huang, X. Liu, Y. Xu, Y. Ma, X. Chen, S. Maybank, D. Tao, Spatiotemporal attacks for embodied agents, ECCV, 2020.
[14] A. Liu, J. Wang, X. Liu, b. Cao, C. Zhang, H. Yu, Bias-based universal adversarial patch attack for automatic check-out, ECCV, 2020.
[15] C. Nicholas, W. David, Towards evaluating the robustness of neural networks, IEEE S&P, 2017.
[16] Y. Shi, Y. Han, Q. Zhang, X. Kuang, Adaptive iterative attack towards explainable adversarial robustness, Pattern Recognit 105 (2020) 107309.

[17] J. Hang, K. Han, H. Chen, Y. Li, Ensemble adversarial black-box attacks against deep learning systems, Pattern Recognit 101 (2020) 107184.
[18] Y. Wang, Z. Yin, R. Gong, J. Wang, J. Wang, A. Liu, X. Liu, Generating transferable adversarial examples against vision transformers, ACM Multimedia, 2022.
[19] S. Liang, B. Wu, Y. Fan, X. Wei, X. Cao, Parallel rectangle flip attack: a query-based black-box attack against object detection, ICCV, 2021.
[20] N. Papernot, P. Mcdaniel, X. Wu, S. Jha, A. Swami, Distillation as a defense to adversarial perturbations against deep neural networks, arXiv preprint arXiv:1511.04508(2015).
[21] C. Xie, J. Wang, Z. Zhang, Z. Ren, A. Yuille, Mitigating adversarial effects through randomization, ICLR, 2018.
[22] F. Croce, M. Hein, Provable robustness against all adversarial l_p-perturbations for $p \geq 1$, ICLR, 2020.
[23] C. Nicholas, W. David, Defensive distillation is not robust to adversarial examples, arXiv preprint arXiv:1607.04311(2016).
[24] A. Athalye, N. Carlini, D. Wagner, Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples, ICML, 2018.
[25] S. Tang, R. Gong, Y. Wang, A. Liu, J. Wang, X. Chen, F. Yu, X. Liu, D. Song, A. Yuille, P.H.S. Torr, D. Tao, RobustART: benchmarking robustness on architecture design and training techniques, arXiv preprint arXiv:2109.05211(2021).
[26] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, M. Hein, Robustbench: a standardized adversarial robustness benchmark, NeurIPS Datasets and Benchmarks Track, 2021.
[27] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE TIP 13 (4) (2004) 600–612.
[28] B. Luo, Y. Liu, L. Wei, Q. Xu, Towards imperceptible and robust adversarial example attacks against neural networks, AAAI, 2018.
[29] T.D. Do, S.C. Hui, A.C.M. Fong, Prediction confidence for associative classification (2005).
[30] D. Hendrycks, T. Dietterich, Benchmarking neural network robustness to common corruptions and perturbations, ICLR, 2019.
[31] C. Zhang, A. Liu, X. Liu, Y. Xu, H. Yu, Y. Ma, T. Li, Interpreting and improving adversarial robustness of deep neural networks with neuron sensitivity, IEEE TIP 30 (2021) 1291–1304.
[32] G.E. Legge, J.M. Foley, Contrast masking in human vision, JOSA 70 (12) (1980) 1458–1471.
[33] W. He, B. Li, D. Song, Decision boundary analysis of adversarial examples, ICLR, 2018.
[34] H. Xu, S. Mannor, Robustness and generalization, Mach, Learn 86 (3) (2012) 391–423.
[35] T. Zhang, Z. Zhu, Interpreting adversarially trained convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 7502–7511.
[36] F. Tramèr, D. Boneh, Adversarial training and robustness for multiple perturbations, NeurIPS, 2019.
[37] J. Rauber, W. Brendel, M. Bethge, Foolbox: a python toolbox to benchmark the robustness of machine learning models (2017).
[38] N. Papernot, I. Goodfellow, R. Sheatsley, R. Feinman, P. McDaniel, Cleverhans v2. 0.0: an adversarial machine learning library, arXiv preprint arXiv:1610.00768 10 (2016).
[39] K. Pei, Y. Cao, J. Yang, S. Jana, DeepXplore: automated whitebox testing of deep learning systems, SOSP, 2017.

Jun Guo is a postgraduate student in Beihang University. He receives his bachelor's degree from Beihang University in 2021. His main research interests include adversarial example, model robustness and AI safety.

Wei Bao received the B.S. degree from Jiangsu Normal University, Xuzhou, China, in 2015 and the Ph.D. degree from Minzu University of China, Beijing, China, in 2019. She is currently an engineer in China Electronics Standardization Institute. Her research interests include natural language processing, speech recognition and artificial intelligence standardization.

Jiakai Wang is now a Research Scientist in Zhongguancun Laboratory, Beijing, China. He received the Ph.D. degree in 2022 from Beihang University, supervised by Prof. Wei Li and Prof. Xianglong Liu. His research interests is Trustworthy AI in Computer Vision, which consists of physical adversarial example generation and adversarial defense.

Yuqing Ma received the Ph.D. degree in 2021 from Beihang University, China. She is currently working as a PostDoc at the school of Computer Science and Engineering, Beihang University. Her current research interests include computer vision, and open world recognition.

Xinghai Gao, Researcher, Beihang University. His main research interests include system engineering and complex system. He has served as PI for multiple foundations and projects in intelligent manufacturing.

Gang Xiao is a researcher in National Key Laboratory for Complex Systems. His main research interests include system engineering and complex system. He has served as PI for multiple foundations and published over 20 papers on top-tier journals.

Aishan Liu received his B.S., M.S. and Ph.D. degrees from Beihang University, in 2013, 2016 and 2021, respectively. He is currently an Assistant Professor at Beihang University. He has published over 30 papers on top-tier conferences and journals. His current research interests include adversarial example, model robustness and AI safety.

Jian Dong received the B.S. degree in software engineering from Beijing Institute of Technology, Beijing, China, in 2012. He is currently a senior engineer in China Electronics Standardization Institute. His research interests include artificial intelligence, big data and basic software standardization.

Xianglong Liu (Member, IEEE) received the B.S. and Ph.D. degrees in computer science from Beihang University, Beijing, China, in 2008 and 2014. He has published over 80 research papers at top venues. His research interests include machine learning, computer vision and multimedia information retrieval.

Wenjun Wu is a Professor with Beihang University, the Deputy Director of the State Key Laboratory of Software Development Environment, and the Deputy Head of the National Artificial Intelligence Standards Group. He studies the service software theory and service system in the field of scientific big data and computing.