Interpreting and Evaluating Neural Network Robustness

Fuxun Yu[1] , **Zhuwei Qin**[2] , **Chenchen Liu**[3] , **Liang Zhao**[4] , **Yanzhi Wang**[5] and **Xiang Chen**[6]
[1,2,4,6]George Mason University
[3]University of Maryland, Baltimore County
[5]Northeastern University
ccliu@umbc.edu, yanz.wang@northeastern.edu, {fyu2, zqin, lzhao9, xchen26}@gmu.edu

Abstract

Recently, adversarial deception becomes one of the most considerable threats to deep neural networks. However, compared to extensive research in new designs of various adversarial attacks and defenses, the neural networks' intrinsic robustness property is still lack of thorough investigation. This work aims to qualitatively interpret the adversarial attack and defense mechanism through loss visualization, and establish a quantitative metric to evaluate the neural network model's intrinsic robustness. The proposed robustness metric identifies the upper bound of a model's prediction divergence in the given domain and thus indicates whether the model can maintain a stable prediction. With extensive experiments, our metric demonstrates several advantages over conventional adversarial testing accuracy based robustness estimation: (1) it provides a uniformed evaluation to models with different structures and parameter scales; (2) it overperforms conventional accuracy based robustness estimation and provides a more reliable evaluation that is invariant to different test settings; (3) it can be fast generated without considerable testing cost.

1 Introduction

In the past few years, Neural Networks (NNs) have achieved superiors success in various domains, *e.g.*, computer vision [Szegedy *et al.*, 2016], speech recognition [Hinton *et al.*, 2012], autonomous systems [Huval *et al.*, 2015], *etc*. However, the recent appearance of adversarial attacks [Kurakin *et al.*, 2016] greatly challenges the security of neural network applications: by crafting and injecting human-imperceptible noises into test inputs, neural networks' prediction results can be arbitrarily manipulated [Ian J. Goodfellow, 2014]. Until now, the emerging pace, effectiveness, and efficiency of new attacks always take an early lead to the defense solutions [Carlini and Wagner, 2017], and the key factors of the adversarial vulnerabilities are still unclear, leaving the neural network robustness study in a vicious cycle.

In this work, we aim to qualitatively interpret neural network models' adversarial vulnerability and robustness, and establish a quantitative metric for the model-intrinsic robustness evaluation. To interpret the robustness, we adopt the loss visualization technique [Goodfellow *et al.*, 2015], which was widely used in model convergence studies. As adversarial attacks leverage perturbations in inputs, we switch the loss visualization from its original parameter space into the input space and illustrate how a neural network is deceived by adversarial perturbations. Based on the interpretation, we design a robustness evaluation metric to measure a neural network's maximum prediction divergence within a constrained perturbation range. We further optimize the metric evaluation process to keep its consistency under extrinsic factor variance, *e.g.*, model reparameterization [Dinh *et al.*, 2017].

Specifically, we have the following contributions:

- We interpret the adversarial vulnerability and robustness by defining and visualizing a new loss surface called decision surface. Compared to the cross-entropy based loss surface, the decision surface contains the implicit decision boundary and provides better visualization effect;
- We testify that adversarial deception is caused by the neural network's neighborhood under-fitting. Our visualization shows that adversarial examples are naturally-existed points lying in the close neighborhood of the inputs. However, the neural network fails to classify them, which caused the adversarial example phenomenon;
- We propose a robustness evaluation metric. Combined with a new normalization method, the metric can invariantly reflect a neural network's intrinsic robustness property regardless of attacks and defenses;
- We reveal that under certain cases, *e.g.*, defensive distillation, the commonly-used PGD adversarial testing accuracy can give unreliable robustness estimation, while our metric could reflect the model robustness correctly.

Extensive evaluation results show that our defined robustness metric could well indicate the model-intrinsic robustness across different datasets, various architectures, multiple adversarial attacks, and different defense methods.

2 Background and Related Work

2.1 Adversarial Attacks and Robustness

Adversarial examples were firstly introduced by [Szegedy *et al.*, 2013], which revealed neural networks' vulnerability to adversarial noises and demonstrated the gap between the artificial cognition and human visual perception. Since then, various adversarial attacks were proposed, such as L-BFGS attack [Kurakin *et al.*, 2016], FGSM attack [Ian J. Goodfellow, 2014], C&W attack [Carlini and Wagner, 2017], blackbox attack [Papernot *et al.*, 2017], *etc*.

Driven by the appearance of adversarial attacks, corresponding defense techniques also emerged, including adversarial training [Ian J. Goodfellow, 2014], defensive distillation [Papernot *et al.*, 2016], gradient regularization [Ross and Doshi-Velez, 2018], adversarial logit pairing [Kannan *et al.*, 2018], *etc*. Among those, MinMax robustness optimization [Madry *et al.*, 2018] is considered as one of the most potent defenses, which boosts model accuracy by integrating the worst-case adversarial examples into the model training.

Currently, testing accuracy under adversarial attacks is used to evaluate the model robustness. However, it is highly affected by the attack specifications and can't comprehensively reflect the actual robustness regarding model-intrinsic properties. For example, one commonly used way to evaluate the model robustness is adopting the testing accuracy under projected gradient descent (PGD) attack as an estimation. However, our experiments demonstrate that such a robustness estimation is highly unreliable: a model with a high PGD testing accuracy could be easily broken by other attacks.

In this work, we aim to provide an intrinsic robustness property evaluation metric that is invariant from the specifications of models, attacks, and defenses.

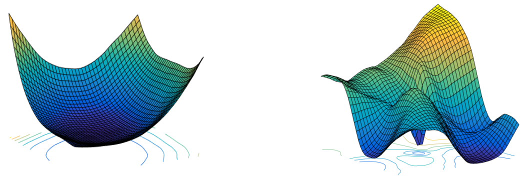
2.2 Neural Network Loss Visualization

Neural network loss visualization is considered as one of the most useful approaches in neural network analysis own to its intuitive interpretation. Proposed by [Goodfellow *et al.*, 2015], the loss visualization is utilized to analyze model training and convergence. Later, [Keskar *et al.*, 2017] further revealed that flat local minima is the key to model generalization in parameter space. However, a model reparameterization issue was identified by [Dinh *et al.*, 2017] that the model parameter scaling may distort the geometry properties.

In this work, we adopt the concept of the loss visualization to analyze the neural network's loss behaviors under adversarial perturbations. Meanwhile, we will also provide a normalization method to solve the model reparameterization problem and derive our scaling-invariant robustness metric.

2.3 Visualization Space Selection

Besides of solving the reparameterization issue, the loss visualization needs further customization for the adversarial perturbation analysis. As the loss visualization mainly evaluates a neural network's generalization ability, it focuses on the *parameter space* to analyze the model training and convergence in previous works. However, such an analysis focus doesn't fit well in the adversarial attacks and defenses, whose action scope is in *input space*. On the other hand, loss function in the *input space* measures the network's loss variations *w.r.t* the input perturbations. It naturally shows the influence of adversarial perturbations and is suitable for studying the robustness to adversarial perturbations. Therefore, we extend the previous methods into the *input space*.



(a) Loss surface in parameter space (b) Loss surface in input space

Figure 1: ResNet's loss surface in (a) Parameter space (b) Input space. The loss surface demonstrates significant non-smooth variation in input space, demonstrating its high sensitivity to noises.

Figure 1 showed two examples of the visualized loss surface of an ResNet model in the *parameter space* and the *input space*, which illustrate the difference between the two visualization spaces. Although the loss surface in the *parameter space* can show a flat minima, its significant non-smooth variations in the *input space* demonstrate the loss is highly sensitive to input perturbations, which can be adversarial vulnerabilities. In this work, we will adopt the *input space* as the default visualization setting for robustness interpretation.

3 Adversarial Robustness Interpretation

3.1 Neural Network Loss Visualization

Loss Visualization Basis

The prediction of a neural network can be evaluated by its loss function $F(\theta, x)$, where θ is the model parameter set (weight and bias) and x is the input. As the inputs x are usually constructed in a high-dimensional space, direct visualization analysis on the loss surface is impossible. To solve this issue, the loss visualization projects the high-dimensional loss surface into a low-dimensional space to visualize it (*e.g.* a 2D hyper-plane). During the projection, two vectors α and β are selected and normalized as the base vectors for x-y hyperplane. Given an starting input point o, the points around it can be interpolated, and the corresponding loss values can be calculated as:

$$V(i, j, \alpha, \beta) = F(o + i \cdot \alpha + j \cdot \beta), \qquad (1)$$

where, the original point o in the function F denotes the original image, α and β can be treated as the unit perturbation added into the image, and the coordinate (i, j) denotes the perturbation intensity. In the loss visualization, a point's coordinate also denotes its divergence degree from the original point along α and β direction. After sampling sufficient points' loss values, the function F with high-dimensional inputs could be projected to the chosen hyper-plane.

Decision Surface Construction

As the loss visualization is mostly used to analyze model convergence, the loss function $F(\theta, x)$ is usually represented by the cross-entropy loss, which constructs a conventional *loss surface* in the visualization. However, one critical limitation of the cross-entropy based loss surface is that, it cannot qualitatively show the explicit decision boundary of one input test, and less helpful for adversarial deception analysis.

Therefore, we propose a *decision surface* to replace the *loss surface* in the loss visualization:

$$S(x) = Z(x)_t - max\{Z(x)_i, \ i \neq t\}, \qquad (2)$$

where, $Z(x)$ is the logit output before the softmax layer, and t is the true class index of the input x. The decision function $S(x)$ evaluates the confidence of prediction. In the correct prediction cases, $S(x)$ should always be positive, while $S(x) < 0$ denotes a wrong prediction. Specifically, $S(x) = 0$ indicates the equal confidence for both correct and wrong prediction, which is the *decision boundary* of model. Consequently, the visualization surface constructed by the function $S(x)$ is defined the decision. Different from the cross-entropy based *loss surface*, the *decision surface* demonstrates explicit decision boundaries, and assist the adversarial analysis.

3.2 Visualizing Adversarial Vulnerability

Experimental Analysis

Based on the loss visualization, we project a neural network's loss behavior into 2D hyper-planes. By comparing the model's 4 different types loss behavior in *decision surface*, we provide a experimental analysis for the adversarial vulnerability.

As shown in Figure 2, the visualized hyper-planes have the central points as the original neural network inputs, and their *x*-axes share the same random input divergence direction – α. Meanwhile, each hyper-plane has a dedicated input divergence direction – β along the *y*-axis, which indicate 4 kinds of perturbations, including random noise, cross-entropy based non-targeted FGSM attack [Kurakin *et al.*, 2016], Least-likely targeted FGSM attack [Kurakin *et al.*, 2016], and non-targeted C&W attack [Carlini and Wagner, 2017]. Specifically β values in the three adversarial attacks can be determined as:

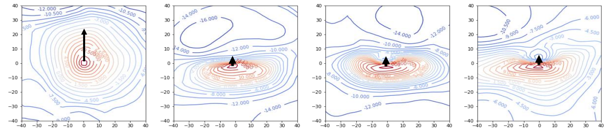
$$\begin{aligned}
\beta_0 &= sign(N(\mu = 0, \sigma = 1)), \\
\beta_1 &= sign(-\nabla_x\, y_t \cdot log(softmax(Z))), \\
\beta_2 &= sign(\nabla_x\, y_l \cdot log(softmax(Z))), \\
\beta_3 &= sign(\nabla_x\, max\{Z(x)_i, i \neq t\} - Z(x)_t),
\end{aligned} \tag{3}$$

where N is normal distribution, Z is the logit output, y_t is the true class label, y_l is least likely class label (both one-hot).

In Figure 2, we use arrows to show the shortest distance to cross the decision boundary $L(x)$=0. As projected in Figure 2(a), when the input is diverged by the perturbation along a random direction, it will take much longer distance to cross the decision boundary. This explains the common sense that natural images with small random noises won't degrade neural network accuracy significantly. By contrast, for the adversarial attacks projected in Figure 2(b)$\sim$(d), the attacks find aggressive directions (β direction shown in *y*-axis), towards which the decision boundary is in the close neighborhood around the original input. Therefore, adding those small perturbations that even human can't perceive into input can mislead the model decision and generates adversarial examples.

Vulnerability Interpretation

The above experimental analysis reveals the nature of adversarial examples: Although a neural network seems to converge well after the model training (the demonstrated model achieves 90% accuracy on CIFAR10), there still exist large regions of image points that the neural network fails to classify correctly (as shown by the large regions beyond the decision boundary in Figure 2(b)$\sim$(d)). What's worse, some of these regions are extremely close to the original input point (even within $\ell_{\text{inf}} < 1$ distance).



(a) Random (b) Cross Entropy (c) Least-Likely (d) C&W

Figure 2: Adversarial vulnerability demonstration when loss surface in input space is projected onto different hyperplanes.

Base on these analysis, we could conclude that, rather than being "generated" by attackers, the adversarial examples are "naturally existed" already that models fail to learn correctly. To fix such intrinsic vulnerability of neural networks, the essential and ultimate robustness enhancement should focus on solving the "neighborhood under-fitting" issue.

3.3 Interpreting Adversarial Robustness

To verify our geometric robustness theory, we compare two pairs of robust and natural models trained on MNIST and CIFAR10 respectively. These models are released from the adversarial attacking challenges [1][2], and built with the same structure but different robustness degrees (natural training and MinMax training [Madry *et al.*, 2018]).

To verify our theory, we visualize the models' decision surfaces for interpretation: (1) As shown in Figure 3, dramatic differences between the natural and robust decision surfaces can be observed: Natural (vulnerable) model's decision surfaces show sharp peaks and large slopes, where the decision confidence could quickly drop to negative areas (wrong classification regions). (2) By comparison, on robust decision surfaces (shown in Figure 3(c)(d)), all neighborhood points around the original input point are located on a high plateau with $L(x) > 0$ (correct classification regions). (3) The surface in the neighborhood is rather flat with negligible slopes until it reaches approximately $\ell_\infty = 0.3$ constraints, which is exactly the adversarial attack constraint used in robust training. Similar phenomenon could be observed in Figure 4 on CIFAR10.

Such robust model's loss geometry verifies our previous conclusion that, fixing the neighborhood under-fitting issue is the essential robustness enhancement solution for neural networks. And a flat and wide plateau around the original point on decision surface is one of the most desired properties of a robust model.

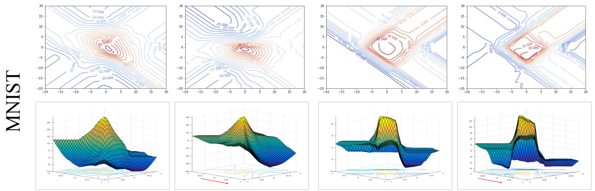
4 Adversarial Robustness Evaluation

4.1 Formal Definition of Robustness Metric

As aforementioned, the decision surface of a robust model should have a flat neighborhood around the input point x. Intuitively explanation is that a robust model should have good prediction stability–its prediction does not have significant change with small perturbations. In fact, models are not always robust–the predictions of a model on clean and noisy inputs are not always the same and can diverge to a large

[1] https://github.com/MadryLab/mnist_challenge

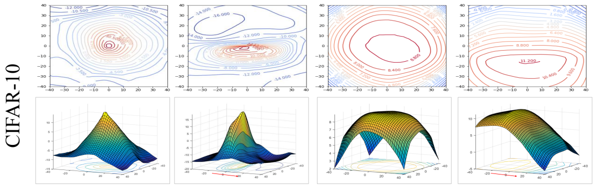
[2] https://github.com/MadryLab/cifar10_challenge



(a) (b)
Adversarial Accuracy: 1.3%

(c) (d)
Adversarial Accuracy: 96.8%

Figure 3: Decision surfaces of the natural and robust models on MNIST. (a)-(b): natural model surfaces in random and adversarial projection; (c)-(d): robust model surfaces in random and adversarial projection (each unit denotes 0.05 perturbation step size)



(a) (b)
Adversarial Accuracy: 0.3%

(c) (d)
Adversarial Accuracy: 44.7%

Figure 4: Decision Surface of natural and robust model on CIFAR10 (step size = 1). As assumed, natural model's surface shows sharp peaks and cliffs while robust model's shows flat plateau.

extent with small adversarial noises. As such, *given a feasible perturbation set, the maximum divergence between the original prediction and the worst-case adversarial prediction could be used to denote the model's vulnerability degree (i.e., the inverses of model robustness).*

Based on this definition, firstly, we calculate the divergence between two predictions on an original input and an adversarial input with perturbations in a defined range. Specifically, we use the Kullback–Leibler divergence, which is known as KL Divergence (D_{KL}) and is a common evaluation metric on measuring the divergence between two probability distributions. The formal robustness could be estimated by:

$$\psi(x) = \frac{1}{\max\limits_{\delta \in set} D_{KL}(P(x),\ P(x+\delta))}, \tag{4}$$

where $P(\cdot)$ is the prediction results from the evaluated model. A lower divergence D_{KL} indicates the model is more robust as a more stable prediction is maintained. The final robustness metric $\psi(x)$ is defined inversely proportional to the maximum D_{KL} since the largest divergence will generate the smallest robustness score $\psi(x)$. To obtain the $\max$ term in Eq. 4, we use the gradient ascent algorithm to directly optimize the KL-divergence, which demonstrates accurate and stable estimations that we will show in Sec. 5.

4.2 Invariant Normalization against Model Reparameterization

The robustness metric defined in previous works has a problem called "model re-parameterization": on the condition that weights and biases are enlarged by the same coefficients simultaneously, a neural network model's prediction results and its robustness property will not change, while the defined KL divergence can have dramatic change [Dinh *et al.*, 2017].

To solve this problem, we design a simple but effective normalization method: the basic idea is to add a scale-invariant normalization layer after the logit layer output. Since the neural network before the logit layer is piecewise-linear, we could then use normalization to safely remove the scaling effect of model reparameterization. The basic process is as follows: firstly, we attain a confidence vector of the logit layer, which can contain either positive or negative values; then we divide them by the max-absolute-value to normalize the confidence vector to the range of (-1, 1) and re-center them into positive range (0, 2). Owning to the max division, the final confidence vector will not change even when the parameters are linearly scaled up (or down). Finally, we use a simple sum-normalization to transform the confidence vector to a valid probability distribution. The overall normalization is:

$$P(x) = \frac{\tilde{F}(x)}{\sum_i \tilde{F}(x_i)},\ \tilde{F}(x) = \frac{F(x)}{\max|F(x)|} + 1. \tag{5}$$

Here $P(x)$ is the final normalized probability distribution, $\tilde{F}$ is the normalized confidence vector, $F(x)$ is the original logit layer output, and x is the input. By the above normalization method, we could successfully alleviate the model reparameterization effect, which is shown in Sec. 5.

5 Robustness Evaluation Experiments

5.1 Experiment Setup

To test the generality of our metric for neural networks' robustness evaluation, we adopt three common datasets (*i.e.* MNIST, CIFAR10, and ImageNet) and different models for the experiment, including FcNet, LeNet, ConvNet, ResNet18, ResNet152, and DenseNet.

To further test our metric on neural networks with different robustness degrees, the following defense settings are applied: No Defense, Adversarial Training [Kurakin *et al.*, 2016], Gradient Regularization Training [Ross and Doshi-Velez, 2018], Defensive Distillation [Papernot *et al.*, 2016], Gradient Inhibition [Liu *et al.*, 2018] and MinMax Training [Madry *et al.*, 2018][3].

Correspondingly, the robustness verification is based on referencing the adversarial testing accuracies from two currently strongest attacks: 30-step PGD (PGD-30) attack based on cross-entropy loss and 30-step CW (CW-30) attacks based on C&W loss. The adversarial perturbations are constrained by the ℓ_∞-norm as 0.3/1.0, 8.0/255.0, 16.0/255.0 on MNIST, CIFAR10, and ImageNet respectively.

5.2 Robustness Metric Evaluation

MNIST Experiments. On MNIST dataset, the results are shown in Table 1: (1) The results firstly demonstrate that our metric could well reflect different robustness degrees on the same neural network model. For example, three FcNet models show increasing robustness in $\psi(x)$, which aligns well with their reference accuracies from both PGD-30 and CW-30 attack; (2) The results also show the generality of our metric on FcNet and LeNet models.

[3] The gradient regularization and MinMax training is re-implemented with Pytorch, which may cause small deviations from the original reported accuracy.

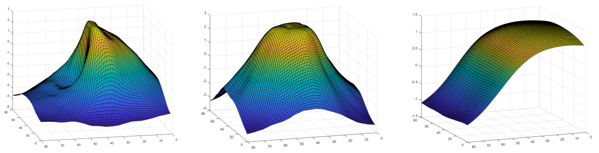
Model	Defense	ψ(x)	PGD-30 Accuracy	C&W-30 Accuracy
FcNet	No Defense	**73.36**	0.73%	0.2%
	AdvTrain	**80.43**	4.43%	2.12%
	MinMax	**297.2**	82.9%	80.3%
LeNet	No Defense	**93.8**	2.82%	1.01%
	AdvTrain	**264.7**	51.8%	46.2%
	MinMax	**958.4**	92.3%	90.3%

*AdvTrain: [Kurakin *et al.*, 2016], MinMax: [Madry *et al.*, 2018].

Table 1: Robustness Metric Evaluation on MNIST

CIFAR10 Experiments. Table 2 shows the experimental results on CIFAR10, including three common neural network models (*i.e.* ConvNet, ResNet18, and DenseNet), as well as three robustness settings (*i.e.* No defense, Gradient Regularization, and MinMax Training). The experiment results show that our metric has the same scale with the referenced adversarial testing accuracies, implying our metric's good generality on complex neural network models and different defenses. To better illustrate a neural network model's robustness, we visualized three ResNet18 models with different robustness degrees in Figure 5. As the robustness degree increases, the models' loss surfaces become more and more smooth. Our empirical visualization results imply that the smoother decision surface in the input space indicates better adversarial robustness, which coincidentally matches the parameter space generalization hypothesis [Keskar *et al.*, 2017].

ImageNet Experiments. In the experiment on MNIST and CIFAR10, our proposed robustness metric aligns well with adversarial testing accuracies of PGD-30 and CW-30. However, when we evaluate the MinMax model on ImageNet, two reference accuracies demonstrate certain inconsistency: The MinMax model is released as the base model of the state-of-the-art defense on ImageNet by [Xie *et al.*, 2018]. To conduct the MinMax training, the reported time needed is about 52 hours on 128 V100 GPUs. Despite that, the reported accuracy showed very good robustness estimation of the model, which can achieve 42.6% under 2000-iteration PGD attacks. However, when we more thoroughly evaluated the model by CW-30 attack, we found the model's testing accuracy is only 12.5% under the attack. We call such a case as "*unreliable estimation*" in PGD-based adversarial testing, whose robustness estimation cannot generalize to all attacks. We will discuss this case and several other similar ones in details in Sec. 5.3, and reveal the current deficiency of adversarial testing based robustness estimation.



(a) Natural Model (Adv. Acc: 0.0%)

(b) GradReg Model (Adv. Acc: 18.72%)

(c) MinMax Model (Adv. Acc: 44.3%)

Figure 5: Different models' loss visualizations: model with higher robustness demonstrates more smooth and stable geometry.

Model	Defense	$\psi(x)$	PGD-30 Accuracy	C&W-30 Accuracy
ConvNet	No Defense	**58.3**	0.0%	0.0%
	GradReg	**86.5**	16.0%	14.8%
	MinMax	**182.6**	39.6%	38.7%
ResNet18	No Defense	**67.9**	0.0%	0.0%
	GradReg	**77.8**	18.7%	17.5%
	MinMax	**162.7**	44.3%	43.1%
DenseNet	No Defense	**59.1**	0.1%	0.0%
	GradReg	**77.9**	18.6%	17.2%
	MinMax	**142.4**	39.1%	38.8%

*GradReg: [Ross and Doshi-Velez, 2018], MinMax: [Madry *et al.*, 2018].

Table 2: Robustness Metric Evaluation on CIFAR10

5.3 Our Metric vs. Adversarial Testing Accuracy

As mentioned above, the adversarial testing accuracy from different adversarial attacks may demonstrate certain inconsistency, and therefore mislead the robustness estimation. In addition to the ImageNet example, we also include another two cases that the adversarial testing accuracies yield unreliable robustness estimation: defensive distillation [Papernot *et al.*, 2016] and gradient inhibition [Liu *et al.*, 2018].

To demonstrate the unreliability of these cases, we train three new models on MNIST and CIFAR10 respectively, using natural training, defensive distillation, and gradient inhibition methods. For the ImageNet model, we use a public released model [4], which can achieve a state-of-the-art accuracy 45.5% against PGD-30 attack (within $\ell_\infty \leq 16/255$).

The overall experimental results are shown in Table. 3, which shows that though all these defenses can achieve high PGD-30 adversarial testing accuracy, they actually bring very limited robustness improvement:

On MNIST and CIFAR10, the distillation and gradient inhibition defenses provide the models with high adversarial testing accuracy against both FGSM and PGD-30 attacks (even higher than state-of-the-art MinMax methods), which seemly indicates these models are significantly robust. However, when measured by our metric, we have the opposite conclusion: these models are merely as robust as no-defense models and incomparable to the robust models trained by MinMax. To further verify this conclusion, we test these models with more adversarial settings and the testing accuracy dramatically degrades to almost zero in all the tests.

The tests above further prove our statement: the adversarial testing accuracy based on PGD-30 may yield unreliable robustness estimation, which cannot reflect the model's intrinsic robustness. This is because the distillation and gradient inhibition both rely on the input gradient vanishing to achieve robustness enhancement, which is mainly provided by the nonlinear softmax and negative log loss. Since C&W attack doesn't rely on the cross-entropy loss, it can easily crack those two defenses. Such a case also applies to the ImageNet model trained with MinMax defenses as shown in the last two rows of Table. 3.

In contrast, our robustness metric can successfully reflect

[4] MinMax model is obtained in following link: https://github.com /facebookresearch/imagenet-adversarial-training.

Dataset	Defense	FGSM Accuracy	PGD-30 Accuracy	$\psi(x)$	C&W-30 Accuracy
MNIST	No Defense	23.4%	3.5%	89.8	0.5%
	Distillation	**97.3%***	**97.1%***	70.5	0.0%
	GradInhib	**98.3%***	**97.8%***	87.0	0.0%
	MinMax	98.3%	92.3%	958.4	90.3%
CIFAR10	No Defense	7.6%	0.1%	58.3	0.0%
	Distillation	**72.6%***	**72.3%***	60.5	0.0%
	GradInhib	**79.8%***	**79.7%***	70.0	0.1%
	MinMax	55.7%	39.6%	182.6	38.7%
ImageNet	No Defense	15.5%	7.3%	1.7×10^5	4.6%
	MinMax	**46.9%**	**45.7%**	2.3×10^5	12.5%

*Distillation: [Papernot *et al.*, 2016], GradInhib: [Liu *et al.*, 2018], MinMax: [Madry *et al.*, 2018] *Bold accuracies denote the unreliable robustness estimation cases.

Table 3: Unreliable Cases of Adversarial Testing Accuracy

the model true robustness property with different defenses. Under all the above cases, the robustness metric gives reliable robustness estimation, remaining un-affected by defense methods and the unreliable PGD adversarial testing accuracy.

5.4 Reparemeterization Invariance Evaluation

The reliability of our proposed metric is also reflected in its invariance from the model parameter scaling. Previous work [Keskar *et al.*, 2017] tried to define a metric, named ϵ-sharpness, to evaluate the loss surface's geometry properties. We adopt its original definition and apply it into our input space to evaluate sharpness of input space loss surface, which can empirically reflect the adversarial generalization as aforementioned.

The experiment results are shown in Table. 4, where ϵ denotes the ϵ-sharpness, ψ_s denotes our robustness metric based on softmax layer without normalization, and ψ_n denotes our robustness metric with normalization. For the test cases, *Org.* indicates the tests with the original model without reparemeterization, *100 and /100 denote the model's logit layer weights and biases are scaled accordingly. Please note that, such scaling won't introduce accuracy and robustness change in practice [Dinh *et al.*, 2017].

The experiments show that, both ϵ-sharpness and unnormalized ψ_s give very distinct robustness estimations influenced by the reparameterization. By contrast, the normalization method successfully alleviates the scaling influence and enables our metric ψ_n to keep a stable estimation under model reparameterization. Therefore, our metric could thus be used to more precisely capture one model's robustness degree without being affected by model reparameterization.

5.5 Efficiency of the Robustness Metric

Here we show the efficiency of our metric compared to adversarial testing methods. Since we are evaluating the model properties, theoretically it should be invariant to how many input we choose. Here we show that as the test-batch-size increases, the calculated robustness metric gradually converge to a stable robustness estimation which is close to the whole test set average robustness. Figure 6 shows the relation with the batch size and the robustness deviation between batches with same batch-size. We can see that on both datasets, as the batch size increases, the robustness measurement become more accurate since they have much smaller deviations. With the batch-size equals to 1000 (or less), we could get the model's robustness estimation with less than 10% deviation on MNIST and 5% on CIFAR10, which demonstrate higher efficiency than accuracy testing running on the whole test set.

Model	Metric	No Defense Model			MinMax Model		
		Org.	*100	/100	Org.	*100	/100
ConvNet	ϵ	22.7	109.6	0.095	0.43	3.20	0.004
	ψ_s	0.96	0.012	1677.8	39.6	5.33	377443.3
	ψ_n	**58.3**	**59.5**	**57.9**	**182.5**	**183.1**	**177.36**
ResNet18	ϵ	15.4	87.4	0.048	0.085	5.63	0.005
	ψ_s	0.963	0.0097	3178.8	17.11	0.158	128089
	ψ_n	**110.9**	**110.8**	**102.5**	**193.0**	**192.62**	**172.5**

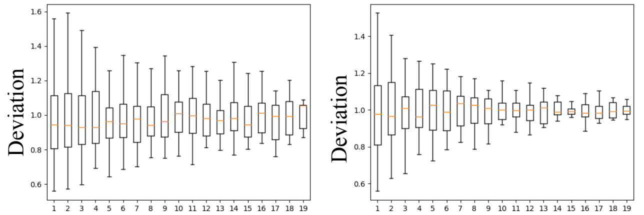
Table 4: Robustness Metrics Comparison under Reparemeterization

5.6 Robustness Estimation Grading

Based on our experiments, we could derive a rough relationship between different robustness evaluation score and the adversarial accuracy. For example, on MNIST dataset within common threat model ($\ell_\infty < 0.3$), we can define the model robustness by three levels: Vulnerable ($acc < 0.3$), Fairly Robust ($0.3 \leq acc < 0.6$) and Robust ($0.6 \leq acc \leq 1.0$). In such a case, the corresponding robust metric range will be (0, 100), (100, 270), (270, ∞), which could be used to quickly grade a neural network's robustness. The robustness grading for CIFAR and ImageNet cannot be well developed yet due to the limited robustness currently (40% and 15%).

6 Conclusion

In this work, through visualizing and interpreting neural networks' decision surface in input space, we show that adversarial examples are essentially caused by neural networks' neighborhood under-fitting issue. Oppositely, robust models manage to smoothen their neighborhood and relieve such under-fitting effect. Guided by such observation, we propose a model intrinsic robustness evaluation metric based on the model predictions' maximum KL-divergence in a given neighborhood constrain. Combined with our new-designed normalization layer, the robustness metric shows multiple advantages than previous methods, including: great generality across dataset/models/attacks/defenses, invariance under reparameterization, and excellent computing efficiency.



(a) MNIST (b) CIFAR10

Figure 6: The robustness measurement is increasingly stable with the increasing batch size (100 to 1000).

References

[Carlini and Wagner, 2017] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[Dinh *et al.*, 2017] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028, 2017.

[Goodfellow *et al.*, 2015] Ian J Goodfellow, Oriol Vinyals, and Andrew M Saxe. Qualitatively characterizing neural network optimization problems. In *International Conference on Learning Representations (ICLR)*, 2015.

[Hinton *et al.*, 2012] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

[Huval *et al.*, 2015] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, et al. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*, 2015.

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

[Kannan *et al.*, 2018] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

[Keskar *et al.*, 2017] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations (ICLR)*, 2017.

[Kurakin *et al.*, 2016] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[Liu *et al.*, 2018] Qi Liu, Tao Liu, Zihao Liu, Yanzhi Wang, Yier Jin, and Wujie Wen. Security analysis and enhancement of model compressed deep learning systems under adversarial attacks. In *Proceedings of the 23rd Asia and South Pacific Design Automation Conference*, pages 721–726. IEEE Press, 2018.

[Madry *et al.*, 2018] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. 2018.

[Papernot *et al.*, 2016] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[Papernot *et al.*, 2017] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

[Ross and Doshi-Velez, 2018] Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *AAAI*, 2018.

[Szegedy *et al.*, 2013] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[Szegedy *et al.*, 2016] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[Xie *et al.*, 2018] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. *arXiv preprint arXiv:1812.03411*, 2018.