INEQUALITY PHENOMENON IN l_∞-ADVERSARIAL TRAINING, AND ITS UNREALIZED THREATS

Anonymous authors
Paper under double-blind review

ABSTRACT

The appearance of adversarial examples raises attention from both academia and industry. Along with the attack-defense arms race, adversarial training is the most effective against adversarial examples. However, we find inequality phenomena occur during the l_∞-adversarial training, that few features dominate the prediction made by the adversarially trained model. We systematically evaluate such inequality phenomena by extensive experiments and find such phenomena become more obvious when performing adversarial training with increasing adversarial strength (evaluated by ϵ). We hypothesize such inequality phenomena make l_∞-adversarially trained model less reliable than the standard trained model when few "important features" are influenced. To validate our hypothesis, we proposed two simple attacks that either perturb or replace important features with noise or occlusion. Experiments show that l_∞-adversarially trained model can be easily attacked when a few important features are influenced. Our work shed light on the limitation of the practicality of l_∞-adversarial training.

1 INTRODUCTION

Adversarial examples of deep neural networks (DNNs) discovered by Szegedy et al. (2013) have brought great threats to the deep learning-based applications, e.g., autonomous driving and face recognition. Defending against adversarial examples is an urgent task before we can deploy DNNs-based applications in real-world scenarios safely and securely. Following the emergence of adversarial examples, various defense methods have been proposed (Guo et al., 2018; Prakash et al., 2018; Mummadi et al., 2019; Akhtar et al., 2018). Among various defenses against adversarial attacks, adversarial training (Goodfellow et al., 2015; Zhang et al., 2019; Madry et al., 2018b) is regarded as the most effective way that increases models' robustness by retraining l_p-norm bounded adversarial samples generated in each training loop. l_∞ adversarial training is the most common adversarial training, which applies adversarial samples with l_∞ bounded perturbation by ϵ.

Many works have been devoted to the understanding of adversarial training from both theoretical and empirical perspectives (Andriushchenko & Flammarion, 2020; Allen-Zhu & Li, 2022; Kim et al., 2021). Ilyas et al. (2019) suggested that adversarially trained model learns robust features from adversarial examples and cleanses non-robust features. Engstrom et al. (2019) also suggested that adversarial training forces the model learning to be invariant to signals that humans are also invariant to. Thus the adversarial robust training leads to feature representations that are more similar to what humans use. Chalasani et al. (2020) theoretically validates that the l_∞-adversarial training suppresses the importance of the redundant features, and the adversarially trained model, therefore, produces more **sparse** and better-behaved feature representations. In general, previous work suggest robust model have a property of **sparseness** regarding feature representation and regard such sparse feature representation as an advantage as it's more human-aligned. Several works study the such property of adversarially trained model and attempt to transfer it to standard trained model by various techniques (Ross & Doshi-Velez, 2018; Salman et al., 2020; Deng et al., 2021).

However, in contrast to the point of previous work regarding such sparse feature representation as merit, we find such sparseness also indicates inequality phenomena that may bring unexpected threats to adversarially trained model. To be specific, we find during l_∞-adversarial training, the model not only suppresses the importance assigned to redundant features (Chalasani et al., 2020) but also suppresses the importance of other features including robust ones. The degree of suppression

is proportional to adversarial strength (evaluated by ϵ). As a result, for a l_∞-adversarially trained model, few features dominate the prediction with given input images. Intuitively, standard trained models make decisions from various features, and some redundant features perform as a "bulwark" when a few important features are distorted unmaliciously. However, regarding a l_∞ robust model, the decision is mostly decided by a few features, such that the prediction could be easily influenced when these important features are altered (see Figure 1). As Figure 1 indicates, adverarially trained model relies on very few features to recognize the street sign. When we occlude the features that the model regard most important of the street sign, the adversarially trained model fails to recognize the street sign even with very small occlusions (but well recognized by human and standard trained model). Supposed for an autonomous car deployed with an adversarially trained model, even if it achieves high adversarial robustness against worst-case adversarial examples, but at a price vulnerable to small occlusions. The practicality of such adversarially trained model is thus questionable.

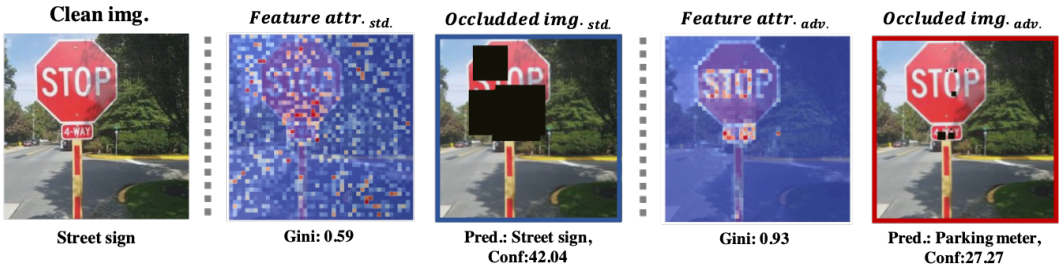

Figure 1: **l_∞-adversarially trained model fails to recognize street sign with small occlusions**. With given feature attribution maps that attribute the importance of each feature, we occlude the image's features of high importance with small patches. The resultant image fools the adversarially trained model successfully.)

In this work, we first study the "inequality phenomenon" of l_∞-adversarial training quantitatively and qualitatively. We find the inequality of l_∞-adversarially trained model reflected in two aspects: 1) global inequality: few features dominate the prediction. 2) regional inequality: features the model considers important tend to cluster in specific regions. We analyze such phenomenon on both ImageNet and Cifar10 with models of different architectures. The inequality phenomenon is much more serious when adversarial training on ImageNet. Based on our findings, we design corresponding attacks to reveal the vulnerabilities caused by such inequality. Experiments show that under the premise that the resultant images can be recognized by human observers, l_∞-adversarially trained models are much more vulnerable than the standard trained models, that l_∞-adversarially trained models can be fooled by occlusion and noise with 100.0% and 94.0% error rate, but regarding standard trained model, only 30.1% and 34.5% are affected respectively. In summary, our contribution can be summarized as follows:

- We identify inequality phenomenon occurs during l_∞-adversarial training. We design corresponding indexes and evaluate such inequality phenomena from multiple aspects (global and regional). We systematically evaluate such phenomena by extensive experiments on broad datasets and models.
- We then identify unrealized threats brought by such inequality phenomena that l_∞-adversarially trained models are much more vulnerable than standard trained models under inductive occlusion or noise. Namely, during the l_∞-adversarial training, the adversarial robustness is obtained with the sacrifice of another robustness.
- Our work gives an intuitive understanding about the weakness of l_∞-adversarially trained model's feature representation from a novel perspective. And our work sheds light on the limitation and the hardness of l_∞-adversarial training.

This paper is organized as follows. Background and related work are discussed in Section 2. Our proposed methodology is described in Section 3, and evaluated in Section 4. Section 5 gives a discussion and conclusion about the work.

2 BACKGROUND AND RELATED WORK

2.1 ADVERSARIAL EXAMPLE

Given a model as $f(x;\theta) : x \to \mathbb{R}^k$ and training dataset denoted as D, empirical risk minimization (ERM) is a standard way (also denoted as standard training) to train the model f through:

$$\min_\theta E_{(x,y)\in D} loss(x, y) \tag{1}$$

where $y \in \mathbb{R}^k$ is the one-hot label for the image and $loss\,(x, y)$ is usually cross-entropy loss. With such a training scheme, deep learning models typically perform well on clean test samples. Adversarial examples (Szegedy et al., 2013) aim to generate perturbation superimposed on clean sample x to fool a well-trained model f. Adversarial example x' can be crafted by following the direction of adversarial gradients (Goodfellow et al., 2015; Kurakin et al., 2016; Madry et al., 2018a) or optimizing perturbation with a given loss (Carlini & Wagner, 2017; Chen et al., 2018). For most adversarial attacks, the generated adversarial perturbation is bounded by a l_p norm ball, where l_∞ is the most commonly used norm.

2.2 l_∞-ADVERSARIAL TRAINING

Various techniques are proposed to improve the models' adversarial robustness (Wong & Kolter, 2018; Akhtar et al., 2018; Meng & Chen, 2017; Raghunathan et al., 2018). However, analysis by Athalye et al. (2018) shows that among various defense methods against adversarial examples, only the adversarial training framework does not rely on the obfuscated gradient and truly increases model robustness. With an adversarial perturbation of magnitude ϵ at input x, a model is considered robust against this attack if:

$$argmax f\,(x;\theta) = argmax f(x+\sigma;\boldsymbol{\theta}), s.t. ||\sigma||_\infty \leq \epsilon \tag{2}$$

To gain adversarial robustness, the core idea of adversarial training is to train models with adversarial examples. Formally, adversarial training minimized the loss function:

$$loss(x, y) = E_{(x,y)\in D}[\max_{||\sigma||_\infty \leq \epsilon} loss(x+\sigma, y)], \tag{3}$$

Where ϵ is a hyper-parameter governing how invariant the resulting l_∞-adversarially robust model should be. The objective $\max_{|\sigma|\leq\epsilon} loss(x+\sigma, y)$ introduces the model to minimize empirical risk on the training data points while also being locally stable in the (radius-ϵ) neighborhood around each of data points x. The objective is approximated via gradient-based optimization methods, e.g., PGD (Madry et al., 2018b). Following several works to improve adversarial training by various techniques (Shafahi et al., 2019; Sriramanan et al., 2021; Jia et al., 2022; Cui et al., 2021).

Interestingly, Ilyas et al. (2019) proposes that by suppressing the importance of non-robust features, adversarial training makes the trained model more focused on robust and also more perceptually-aligned feature representations. In this process, both the gradient and feature representation becomes more sparse. Chalasani et al. (2020); Salman et al. (2020); Utrera et al. (2020) suggests that the feature representation generated by adversarially trained model is concise as it is sparse and human-friendly. It only assigns the feature that is truly predictive of the output with significant contributions.

2.3 HOW INEQUALITY FORMS DURING ADVERSARIAL TRAINING

In (Chalasani et al., 2020), they theoretically prove the connection between adversarial robustness and sparseness: supposed the adversarial perturbation σ satisfying $||\sigma||_\infty \leq \epsilon$, the weights of "weak" signals are on average more aggressively shrunk toward zero than during the standard training, and the rate of shrinkage is proportional to adversaries' strength (evaluated by ϵ). In other words, standard training can result in models where many weak signals have significant weights, whereas l_∞-adversarial training tends to selectively reduce the magnitude of the weights of weakly relevant or irrelevant signals and push most of these weights close to zero. In the end, the feature attribution maps generated by gradients-based feature attribution methods (Smilkov et al., 2017; Lundberg & Lee, 2017; Sundararajan et al., 2017) would be more sparse. They regard such sparseness as a merit of adversarial training as it produces more concise and human-aligned feature attributions. Different from the point proposed by Chalasani et al. (2020), we further study such sparseness and find it indicates an inequality phenomenon, which may bring unexpected threats to adversarially trained model but ignore by current works.

3 METHODOLOGY

In this section, we first propose the method used to measure the inequality degree of the test data point's feature attribution map. Then we propose two types of attacks to validate our hypothesis: such inequality may bring threats to the robust model.

3.1 MEASURING THE INEQUALITY OF A TEST DATA POINT

To characterize the inequality degree of feature attribution with a given test data point x and model f, we first need to acquire its feature attribution. Various feature attribution methods have been proposed in recent years (Smilkov et al., 2017; Lundberg & Lee, 2017; Sundararajan et al., 2017). In general, feature attribution methods rank the input features according to their purported importance in model prediction. In detail, we treat the input image x as a set of pixels $x = \{x_i, i = 1...M\}$ feature attribution methods aim to assign a corresponding effect a_i for each x_i. We denote the generated feature attribution map of x of model f as $A^f(x)$, where $A^f(x)$ is composed of a_i. Feature attribution methods attribute an effect a_i to each feature x_i, and summing the effects of all feature attributions approximates the output $f(x)$. x_i achieves the top-most score (a_i) is considered the most important feature for prediction, whereas those with the bottom-most score are considered as least important.

In this work, we do not rely on a specific feature attribution method. With a given sorted $A^f(x) = \{a_i, i = 1...M | a_i < a_{i+1}\}$ generated by a typical feature attribution method, if $f(x) \approx a_0 + \sum_{i=1}^{N} a_i, N << M$, we name such distribution of $A^f(x)$ is unequal. Namely, the prediction on x made by model f is almost decided by a very small set of features. Formally, we use Gini index (Dorfman, 1979) to measure the inequality of the distribution of a given feature attribution map. The Gini index is often used to measure wealth inequality. Formally, given a population set indexed in non-decreasing order $\Phi = \{\phi_i, i = 1...n | \phi_i \leq \phi_{i+1}\}$, Gini coefficient can be calculated as:

$$gini(\Phi) = \frac{1}{n}\left(n + 1 - 2\frac{\sum_{i=1}^{n}(n+1-i) * \phi_i}{\sum_{i=1}^{n}\phi_i}\right) \tag{4}$$

$gini(\cdot)$ ranges from 0 to 1. Here, 0 corresponds to perfect equality and 1 corresponds to perfect inequality. We define two types of inequality as follows:

- **Global inequality:** Given a feature attribution map $A^f(x) = \{a_i, i = 1...M | a_i < a_{i+1}\}$ on test data point x, we only consider the inequality degree of the global distribution of $A^f(x)$ and take no into account for other factors, the inequality degree is calculated with $gini_g(A^f(x))$ directly. The higher of $gini_g(A^f(x))$, the more unequal the distribution $A^f(x)$, the fewer features take the most prediction power. When $gini_g(A^f(x))$ is equal to 1, it indicates one feature dominates the prediction while all the other features have no contribution.
- **Regional inequality:** We also consider inequality degree together with spatial factor, whether important feature tends to cluster at specific regions. We first divide and group features into different regions, and calculate the sum of features' importance by groups, formally, $A_r^f(x) = \left\{a_{r_i}, i = 1...m | a_{r_i} < a_{r_{i+1}}\right\}$, where a_r is the sum of a_i in the defined region resultant Gini value on $A_r^f(x)$ reflects the inequality degree of different regions. The higher the value of $gini_r(A_r^f(x))$, the more important features are likely to cluster in specific regions. When $gini_r(A_r^f(x))$ is equal to 1, it indicates all features make a contribution to the prediction cluster in a specific region.

In what follows, we propose potential threats caused by such inequality (global and regional inequality). The designed attacks utilizing common corruptions (noise and occlusion) aim to reveal the unreliability of such decision pattern by l_∞-adversarially trained model.

3.2 ATTACK ALGORITHMS

We propose two simple attacks to validate potential vulnerabilities caused by such inequality phenomena: 1) inductive noise attack. 2) inductive occlusion attack.

3.2.1 INDUCTIVE NOISE ATTACK

We design two types of noise to evaluate the models' stability.

- **Noise (Type I):** Given a image x, we perturb images with Gaussian noise $\sigma \in \mathcal{N}(0,1)$ by masking important features with M. Formally:

$$x' = x + M * \sigma, \quad where \quad M_i = \begin{cases} 0, a_i < a_{tre} \\ 1, a_i \geq a_{tre} \end{cases} \tag{5}$$

  where a_{tre} represents the threshold that below the value will be kept as original images, and we perturb x_i that $a_i \geq a_{tre}$ by Gaussian noise.

- **Noise (Type II):** Regarding the second type of noise attack, we directly replace important features with Gaussian noise, formally $x' = \overline{M} * x + M * \sigma$, where $\overline{M}$ represents reverse mask of M that used to keep original images. In this attack, we replace important features totally compared to Noise type I, this type of noise disturb images more severely.

Regarding inductive noise attack, if the model's decision pattern is extremely unequal, the performance will be highly influenced when important features are corrupted by noise.

3.2.2 INDUCTIVE OCCLUSION ATTACK

We set max count N and max radius R to mask important regions with occlusions gradually. The order of perturbing regions is depended on the order of $A_r^f(x)$, where feature of higher a_{r_i} is firstly perturbed with occlusion with size $r \in \{1...R\}$. The number of occlusions is constrained in $n \in \{1...N\}$ where n represents the number of regions to occlude, and r represents the size of occlusion. We also consider occlusion with different colors to reflect potential real-world occlusion (e.g., black, grey, white). The inductive occlusion attack algorithm is designed as follows:

Algorithm 1 Inductive Occlusion Attack

Require: Test data point (x, y), Model f, Regional Attribution map $A_r^f(x)$, Max count and radius N, R, Perturb color c.
Ensure: $f(x) = y$ ▷ Ensure the test data x is correctly classified by model f.
  $n \leftarrow 1, r \leftarrow 1, x' = x$
  for $n = 1$ to N **do**
    for $r = 1$ to R **do**
      $M \leftarrow$ get_perturb_ mask($A_r^f(x)$, n, r) ▷ A function to acquire the perturbation mask.
      $x' = \overline{M} * x + M * c$ ▷ Perturb x by mask M with color c.
      If $f(x') \neq y$:**break**
    end for
  end for
  return x'

Note the intention of this work is not to propose strong adversarial attacks against adversarially trained models. Although either noise or occlusion is beyond the threat model considered in l_∞ adversarial training, our intention is to reveal the vulnerabilities caused by such inequality phenomena. We augment that the inequality decision pattern of l_∞-trained adversarial models make themselves more fragile under some designed corruptions.

4 EXPERIMENTS

The experiments are organized as follows. We first outline the experimental setup. We then evaluate the inequality degree (by Gini) of different models trained with different adversarial strengths. Then we evaluate how much the models could be affected by inductive noise. Afterward, we analyze the models' performance under inductive occlusions. Finally, we perform an ablation study about the selection of feature attribution methods.

4.1 EXPERIMENTAL SETTINGS

Dataset and models. Regarding evaluating inequality degree by gini coefficient of models, we use ResNet18 (He et al., 2016), ResNet50, WideResNet50 (Zagoruyko & Komodakis, 2016) trained on ImageNet (Deng et al., 2009) and ResNet18, DenseNet (Huang et al., 2017) trained on Cifar10 (Krizhevsky et al., 2009). We use l_∞-adversarially trained model provided by Microsoft [1]. Regarding feature attribution methods, we consider methods including: Input X Gradients (Shrikumar et al., 2016), Integrated Gradients (Sundararajan et al., 2017), Shapley Value (Lundberg & Lee, 2017) and SmoothGrad (Smilkov et al., 2017). Considering space and time efficiency, we mainly present results based on Integrated Gradients and perform an ablation study on the other feature attribution methods. Regarding feature attribution methods, we use implementation by Captum[2]. For each experiment, we randomly selected 1000 correctly classified images from ImageNet for evaluation.

Metrics. For all the tests about the models' performance, we use error rate (%) as the metric to evaluate the model's performance under corruptions (e.g., noise and occlusions), which is the proportion of misclassified test images among the total number of test images defined as $\frac{1}{N}\sum_{n=1}^{N}[f(x) \neq f(x')]$, where x represents clean test images, and x' represents test images corrupted by noise and occlusions.

4.2 INEQUALITY TEST

In this section, we first evaluate the inequality degree (both global and regional inequality) of l_∞-adversarially trained models and standard trained models with different architectures (ResNet18, ResNet50, WideResNet, DenseNet) trained on ImageNet and Cifar10. We also evaluate the inequality degree of different models adversarially trained with increasing l_∞-norms (ϵ = 1, 2, 4, 8). Regarding the evaluation on Gini, We applied the Gini index to the sorted absolute value of the flattened feature attribution maps. The results are presented in table 1. As shown in Table 1, on Cifar10,

Table 1: **Gini index across different models.** We evaluate the Gini coefficient of different models trained with different ϵ on ImageNet and Cifar10.

Dataset	Model	Std. trained	ϵ = 1.0	ϵ = 2.0	ϵ = 4.0	ϵ = 8.0
	Global Inequality					
Cifar10	**ResNet18**	0.58 ± 0.05	0.65 ± 0.05	0.67 ± 0.06	0.69 ± 0.06	0.73 ± 0.06
	DenseNet	0.57 ± 0.04	0.66 ± 0.06	0.67 ± 0.06	0.69 ± 0.06	0.72 ± 0.07
	Regional Inequality					
	ResNet18	0.79 ± 0.02	0.87 ± 0.04	0.87 ± 0.04	0.88 ± 0.04	0.88 ± 0.04
	DenseNet	0.79 ± 0.02	0.85 ± 0.04	0.86 ± 0.04	0.87 ± 0.04	0.88 ± 0.03
	Global Inequality					
ImageNet	**ResNet18**	0.60 ± 0.04	0.69 ± 0.06	0.79 ± 0.04	0.92 ± 0.01	0.95 ± 0.01
	ResNet50	0.62 ± 0.04	0.75 ± 0.05	0.86 ± 0.03	0.92 ± 0.02	0.94 ± 0.01
	WideResNet	0.62 ± 0.05	0.74 ± 0.05	0.79 ± 0.04	0.88 ± 0.03	0.94 ± 0.01
	Regional Inequality					
	ResNet18	0.80 ± 0.02	0.83 ± 0.04	0.88 ± 0.03	0.95 ± 0.01	0.97 ± 0.01
	ResNet50	0.84 ± 0.02	0.91 ± 0.05	0.95 ± 0.02	0.96 ± 0.01	0.97 ± 0.01
	WideResNet	0.81 ± 0.03	0.86 ± 0.03	0.88 ± 0.03	0.93 ± 0.03	0.97 ± 0.02

the global inequality degree of the standard trained model with various architectures is around 0.58. And the l_∞-adversarially trained model at most has a gini value around 0.73 when $\epsilon = 8$. However, on ImageNet, the inequality phenomena are much more severe. Especially for an adversarially trained Resnet50 ($\epsilon = 8$), the gini value is as high as 0.94, which indicates that individual features almost decide the prediction. Experiments on Cifar10 and ImageNet illustrate that l_∞-adversarially trained model tends to use fewer features to support the prediction with the increasing of the adversarial strength (ϵ). We also test the inequality degree of different classes on ImageNet; classes

[1] https://github.com/microsoft/robust-models-transfer

[2] https://github.com/pytorch/captum

related to animal tends to have higher inequality, e.g., class 'Bustard' has the highest inequality of 0.950. Classes related to scene or stuff tend to have lower inequality, e.g., class 'Web site' has the lowest inequality of 0.890 (See more results in Appendix A.1).

We visualize the features' attribution of given images returned by the standard and adversarially trained ResNet50 in Figure 2. As shown in Figure 2, when the model is adversarially trained with

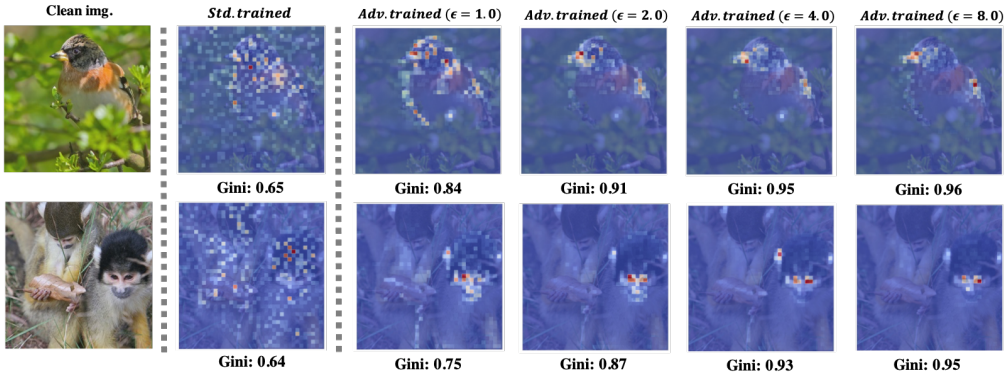

Figure 2: **Feature attributions of different models.** We visualize feature attributions generated by $l_\infty-$adversarially trained models (adversarially trained by adversaries of different ϵ), the larger of ϵ, the fewer features that model relies on for prediction.

weak adversarial strength ($\epsilon = 1$), the model has similar feature attribution to human observers (e.g., the head). However, when the adversarial strength increases, the model gradually assigns higher importance to fewer features (e.g., the eye). To summarize, adversarially trained model tends to focus and rely on fewer features, and these important features tend to gather in specific regions (see more visualization results in Appendix A.4).

4.3 Evaluation under Inductive noise Attack

In this part, we compare the performance of standard- and adversarially- trained ResNet50 under random and inductive noise. We set noise with different scales, including subpixels of 500, 1000, 5000, 10000, and 20000. We present the results in Figure 3.

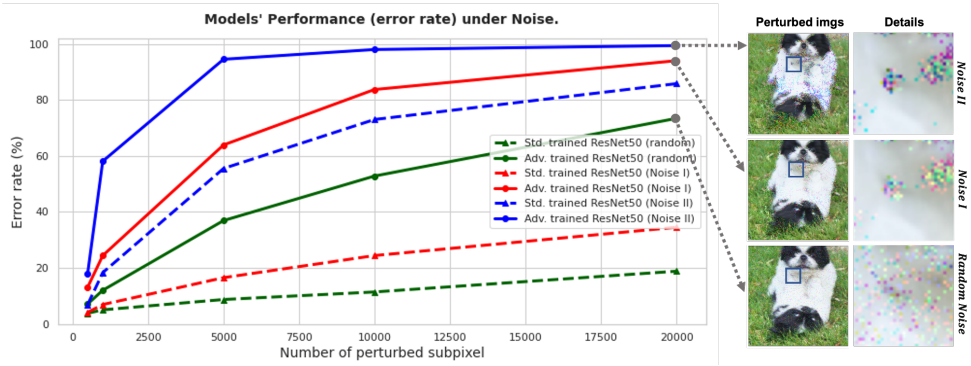

Figure 3: **Evaluation under noise.** We plot the error rate of standard- and adversarially-trained models on images perturbed with increasing number of noise.

Under random noise, the adversarially trained model could be affected at 73.4%, but the standard trained model was only misclassified at 18.8%. Under Noise-I, the adversarially trained model is fooled with 94.0%, while the standard trained model is only fooled at 34.5%. Under Noise-II, even when we control the amount of noise under a small threshold (e.g., 1000 subpixels), more than 50% of predictions made by the adversarially trained model is affected. When we enlarge the

threshold to 20000, the adversarially trained model (ϵ=8) is almost fooled with a 100% success rate. In summary, compared to the standard trained model, l_∞-adversarial model relies on fewer features to make a decision; such a decision pattern results in unstable prediction under noise.

4.4 Evaluation under Inductive Occlusion Attack

In this part, we perform an inductive occlusion attack and evaluate the standard trained and l_∞-adversarially trained ResNet50s' performance. We set two group experiments with different thresholds.

In the first group of experiments, we generate occlusions with a max count of 5 and a max radius of each occlusion of 10. Under such a setting, the adversarially trained model is fooled at a 71.7% error rate, but the standard trained model' predictions are only affected by 31.6%. When we enlarge the threshold and set max count as 10 and radius as 20, both $\epsilon = 4$ and $\epsilon = 8$ adversarially trained model can be fooled with 100% success rate while only 41.2% attack success rate regarding the standard trained model. We visualize corresponding results (see Figure 4). As the figure illustrates, even under the same threshold, l_∞-adversarially trained model with larger ϵ could be easily attacked by smaller occlusions. E.g., in Figure 4, the standard trained model can recognize 'Bulbul' well with the head part totally occluded, but the adversarially trained model fails to recognize the 'Bulbul' even if only the beak of the bulbul is occluded.

Table 2: **Models' performance (Error rate %) under occlusions.** We evaluate the models' performance by gradually occluded important areas with patches of different sizes and colors.

Model	Std.	ϵ = 1.0	ϵ = 2.0	ϵ = 4.0	ϵ = 8.0
Max cnt N = 5, R = 10					
Occlusion-G	23.5%	31.6%	38.4%	32.4%	54.0%
Occlusion-W	28.3%	48.4%	57.5%	61.3%	71.7%
Occlusion-B	31.6%	51.5%	53.3%	48.9%	64.6%
Max cnt N = 10, R = 20					
Occlusion-G	30.1%	48.2%	56.3%	100.0%	100.0%
Occlusion-W	40.1%	59.1%	73.3%	100.0%	100.0%
Occlusion-B	41.2%	70.2%	72.2%	100.0%	100.0%

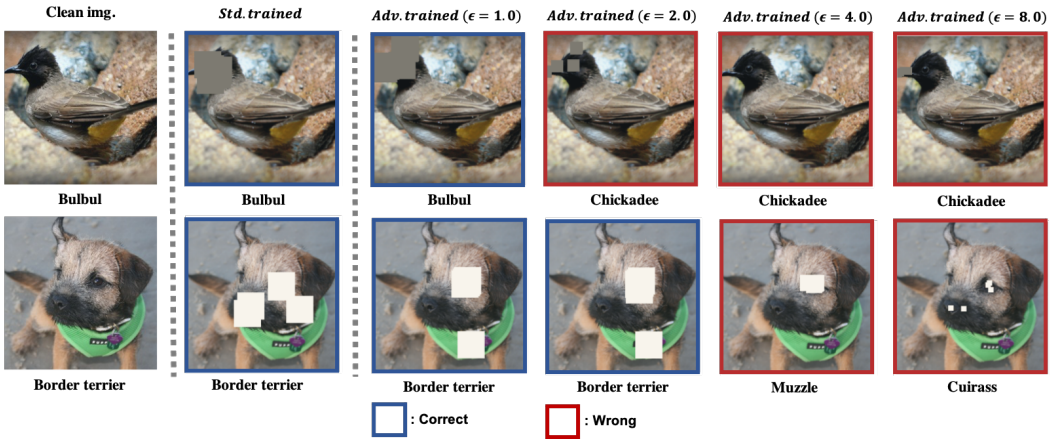

Figure 4: **Visualization of occluded images.** We visualize images occluded with different patches of different sizes and the corresponding predictions made by standard and l_∞-adversarially trained models. Compared to a standard trained model, the adversarially trained model is fragile when occlusion covers the area of important features.

Compared with adversarial perturbation, occlusion has a more practical meaning that occlusion frequently occurs in the real world. We hope the result in this part will draw more attention to the practicality of adversarially trained models when deployed in the real world.

4.5 Ablation Study

We consider four attribution methods: Input X Gradient (Shrikumar et al., 2016), SmoothGrad (Smilkov et al., 2017), Gradient Shapley Value (GradShap for short) (Lundberg & Lee, 2017) and Integrated Gradients (Sundararajan et al., 2017). More details and parameters setting can be seen in Appendix A.2. We perform an ablation study to evaluate the effect of selection on the feature attribution methods (see Table 3).

Table 3: **Ablation study on selection.** We evaluate our hypothesis with different feature attribution methods.

Attribution Method	Model	Gini	Noise I	Noise II	Occlusion-B	Occlusion-G	Occlusion-W
Input X Gradient	Std. trained	0.63	16.1%	45.0%	24.7%	16.8%	23.5%
	Adv. trained	0.93	60.9%	90.4%	63.3%	51.2%	63.2%
GradShap	Std. trained	0.62	19.8%	54.7%	31.5%	24.1%	29.9%
	Adv. trained	0.93	62.3%	93.7 %	64.8%	53.3%	71.6%
SmoothGrad	Std. trained	0.75	63.5%	45.0%	32.3%	25.6%	30.3%
	Adv. trained	0.98	82.5%	98.3%	61.6%	49.7%	60.8%
Integrated Gradients	Std. trained	0.62	16.5%	55.5%	31.6%	23.5%	28.3%
	Adv. trained	0.94	63.9 %	95.5%	64.6%	54.0%	71.7%

Among various attribution methods, SmoothGrad produces more spare feature attribution maps and thus results in a higher Gini value. Regarding evaluation under noise, SmoothGrad improves inductive noise attack's success rate. Regarding evaluation under occlusion, Integrated Gradients produce better attack results.

In conclusion, the selection of attribution methods slightly affects our experimental results (e.g., attacks' success rate) but does not change our augment: the distribution of features' attribution by the adversarially trained model is more unequal; such inequality makes the adversarially trained model more vulnerable to inductive noise and occlusions.

5 Discussion and Conclusion

In this work, we study the inequality phenomena occur in l_∞-adversarial training. Different from the perspective that previous work regards sparse feature representation learned by the adversarially trained model as an advantage, we find the feature representation of l_∞ robust model could be too sparse than we expect. We perform extensive experiments to evaluate such phenomena: l_∞ robust model assigns a few features with extremely high importance. Thus, a few features dominate the prediction. Such extreme inequality threatens the l_∞ adversarially trained model. We also design attacks (by utilizing noise and occlusion) to validate our augment that adversarially trained models could be more vulnerable. To conclude, an attacker can easily fool the l_∞-trained model by altering important features with either noise or occlusion easily. We suggest that both noise and occlusion are common in a real-world scenario. Robustness against either noise or occlusion is more essential and crucial than robustness against adversarial examples. Besides, regarding occlusion attack, as the region where adversarially trained model regarded as important similar to human use, the such attack could be easily realizable in the real world. Our work reveals the limitation and vulnerability of the current l_∞-adversarially trained model.

We also propose a strategy to release such inequality phenomena during l_∞-adversarial training. We combine Cutout (DeVries & Taylor, 2017) with adversarial training and force the model learning features from different regions by cutting out part of training images at each iteration during the training (see the result in Appendix A.3). The strategy slightly releases the inequality degree of adversairally trained model. More effective strategies releasing such extreme inequality of l_∞-adversarial training could be a crucial and promising direction as future work. Besides safety and security issues, this work also provides an intuitive understanding of adversarial models' feature representation. We hope our work can motivate new research into the characteristics of adversarial training and open up further challenges for reliable and practical adversarial training.

ETHICS STATEMENT

In this paper, we identify inequality phenomena occur during l_∞-adversarial training, that l_∞-adversarially trained model tends to use few features to make the decision. We give a systematical evaluation of such inequality phenomena across different datasets and models with different architectures. We further identified unrealized threats caused by such decision patterns and validated our hypothesis by designing corresponding attacks. Our findings provide a new perspective on inspecting adversarial training. Our goal is to understand current adversarial training's weaknesses and make DNNs truly robust and reliable. We did not use crowdsourcing and did not conduct research with human subjects in our experiments. We cited the creators when using existing assets (e.g., code, data, models).

REPRODUCIBILITY STATEMENT

We present the settings of hyper-parameters and how they were chosen in the experiment section. We repeat experiments multiple times with different random seeds and show the corresponding standard deviation in the tables. We plan to open the source code to reproduce the main experimental results later.

REFERENCES

Naveed Akhtar, Jian Liu, and Ajmal Mian. Defense against universal adversarial perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3389–3398, 2018.

Zeyuan Allen-Zhu and Yuanzhi Li. Feature purification: How adversarial training performs robust deep learning. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 977–988. IEEE, 2022.

Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33:16048–16059, 2020.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *ACM Workshop on Artificial Intelligence and Security*, pp. 3–14. ACM, 2017.

Prasad Chalasani, Jiefeng Chen, Amrita Roy Chowdhury, Xi Wu, and Somesh Jha. Concise explanations of neural networks using adversarial training. In *International Conference on Machine Learning*, pp. 1383–1391. PMLR, 2020.

Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: Elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Jiequan Cui, Shu Liu, Liwei Wang, and Jiaya Jia. Learnable boundary guided adversarial training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15721–15730, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Zhun Deng, Linjun Zhang, Kailas Vodrahalli, Kenji Kawaguchi, and James Y Zou. Adversarial training helps transfer learning via better representations. *Advances in Neural Information Processing Systems*, 34:25179–25191, 2021.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Robert Dorfman. A formula for the gini coefficient. *The review of economics and statistics*, pp. 146–149, 1979.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer vision and Pattern Recognition*, pp. 770–778, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Neural Information Processing Systems*, pp. 125–136, 2019.

Xiaojun Jia, Yong Zhang, Baoyuan Wu, Ke Ma, Jue Wang, and Xiaochun Cao. Las-at: Adversarial training with learnable attack strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13398–13408, 2022.

Hoki Kim, Woojin Lee, and Jaewook Lee. Understanding catastrophic overfitting in single-step adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8119–8127, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *International Conference on Learning Representations*, 2016.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018a.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018b.

Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pp. 135–147, 2017.

Chaithanya Kumar Mummadi, Thomas Brox, and Jan Hendrik Metzen. Defending against universal perturbations with shared adversarial training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4928–4937, 2019.

Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8571–8580, 2018.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018.

Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32, 2019.

Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, et al. Towards efficient and effective adversarial training. *Advances in Neural Information Processing Systems*, 34:11821–11833, 2021.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2013.

Francisco Utrera, Evan Kravitz, N Benjamin Erichson, Rajiv Khanna, and Michael W Mahoney. Adversarially-trained deep nets transfer better: Illustration on image classification. In *International Conference on Learning Representations*, 2020.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pp. 5286–5295. PMLR, 2018.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

A APPENDIX

A.1 EQUALITY DEGREE OF DIFFERENT CLASSES

We test inequality degree of feature attributions' distribution of 50000 samples from 1000 classes in ImageNet. We present results in Table 4.

Table 4: **Global inequality degree of different classes ($l_\in$-Adv. traind, $\epsilon = 8$)**

Top-5	Class	Bustard	Manhole cover	Oystercatcher	Redshank	Pomeranian
	Gini	0.950	0.949	0.949	0.949	0.949
Bottom-5	Class	Web site	Slot	Grocery store	Grille	Comic book
	Gini	0.890	0.900	0.901	0.901	0.905

Classes with top-5 inequality (gini value) are: Bustard, Manhole cover, Oystercatcher, Redshank and Pomeranian. And their Gini values are 0.950, 0.949, 0.949, 0.949 and 0.949. Classes with bottom-5 inequality (gini value) are: Web site, Slot, Grocery store, Grille and Comic book. Their corresponding gini values are: 0.890, 0.900, 0.901, 0.901 and 0.905.

Regarding regional inequality, we present results in Table 5. Classes with high regional inequality

Table 5: **Regional inequality degree of different classes ($l_\in$-Adv. traind, $\epsilon = 8$)**

Top-5	Class	Redshank	American coot	Oystercatcher	Bustard	Gazelle
	Gini	0.976	0.975	0.975	0.974	0.974
Bottom-5	Class	Web site	Grocery store	Confectionery	Slot	Grille
	Gini	0.929	0.932	0.934	0.935	0.936

are similar to classes with high global inequality. Specifically, classes of the top 5 (regional inequality) are Redshank, American coot, Oystercatcher, Bustard and Gazelle. And the $gini_r(.)$ of these classes are: 0.976, 0.975, 0.975, 0.974 and 0.974. Classes in the bottom 5 (regional inequality) are Web site, Grocery store, Confectionery, Slot, and Grille. And the $gini_r(.)$ of these classes are: 0.929, 0.932, 0.934, 0.935 and 0.936.

A.2 DETAILS ABOUT FEATURE ATTRIBUTION METHOD

We use attribution methods including Input X Gradient, Smooth Gradient (short for SmoothGrad), Gradient Shapley Value (short for GradShap) and Integrated Gradients.

Input X Gradient: The Input X Gradient multiplies input with the gradient with respect to each input feature. It is a baseline approach for computing the attribution.

GradShap: Shapley Values aims compute each feature's attribution based on cooperative game theory. GradShap approximates Shapley Values by computing the expectations of gradients by randomly sampling from the distribution of $baselines$. It adds noise to each input sample $n_{samples}$ times, selects a random baseline from the baselines' distribution, and a random point along the path between the baseline and the input. Then it computes the gradient of outputs with respect to those selected random points. In our evaluation, we set $n_{samples} = 20$ for experiments with ImageNet and $n_{samples} = 10$ for experiments with Cifar10. We set $baseline = 0$ for all the experiments.

Integrated Gradients: Integrated Gradients is an axiomatic model which assigns an importance score to each input feature by approximating the integral of gradients of the model's output with respect to the inputs along the path (straight line) from given $baseline$ to inputs. Previous work points out Integrated Gradients method is sensitive to the choice of path. To reduce such sensitivity, Integrated Gradients are usually repeated for n_{step} steps. For all the experiments, we set $n_{step} = 20$, and $baseline \in \mathcal{N}(0, 1)$.

SmoothGrad: SmoothGrad adds gaussian noise to each input in the batch $n_{samples}$ times, then applies the given attribution algorithm to each of the samples. It returns the mean of the sampled attributions. SmoothGrad returns a sparser feature attribution map than other methods. In our experiment, we set $n_{samples} = 20$

A.3 How to release inequality phenomenon

We also try to propose a strategy to release the inequality phenomenon in adversarial training. Intuitively, we hope adversarially trained models learns to find robust features from the whole image rather than focus on a specific robust feature. Towards this end, we incorporate Cutout with adversarial training that Cutout enables the model to learn features from multiple spatial spaces. We evaluate our strategy on Cifar10. The results are presented in Table 1.

Model	Gini	Gini-R	Clean Acc.	Adv. Acc.	Noise I	Noise II	Occlusion-B	Occlusion-G	Occlusion-W
Adv. trained	0.73	0.88	82.10 %	48.50 %	40.31 %	1.58%	100.00%	100.00%	100.00%
Adv. trained+Cutout	0.70	0.88	81.40 %	47.20 %	40.78%	1.72%	93.39%	29.98%	55.06 %

Table 6: **l_∞-adversarial training with and without Cutout.**

As the table indicates, the Cutout strategy can slightly release the inequality of l_∞-adversarial training. But at a price, both clean accuracy and adversarial accuracy slightly decrease. Regarding performance under noise and occlusion, the strategy does not improve the adversarial trained model's performance under noise but improves its performance under occlusions. A more effective strategy to release inequality phenomena is highly required.

A.4 MORE VISUALIZATION RESULTS

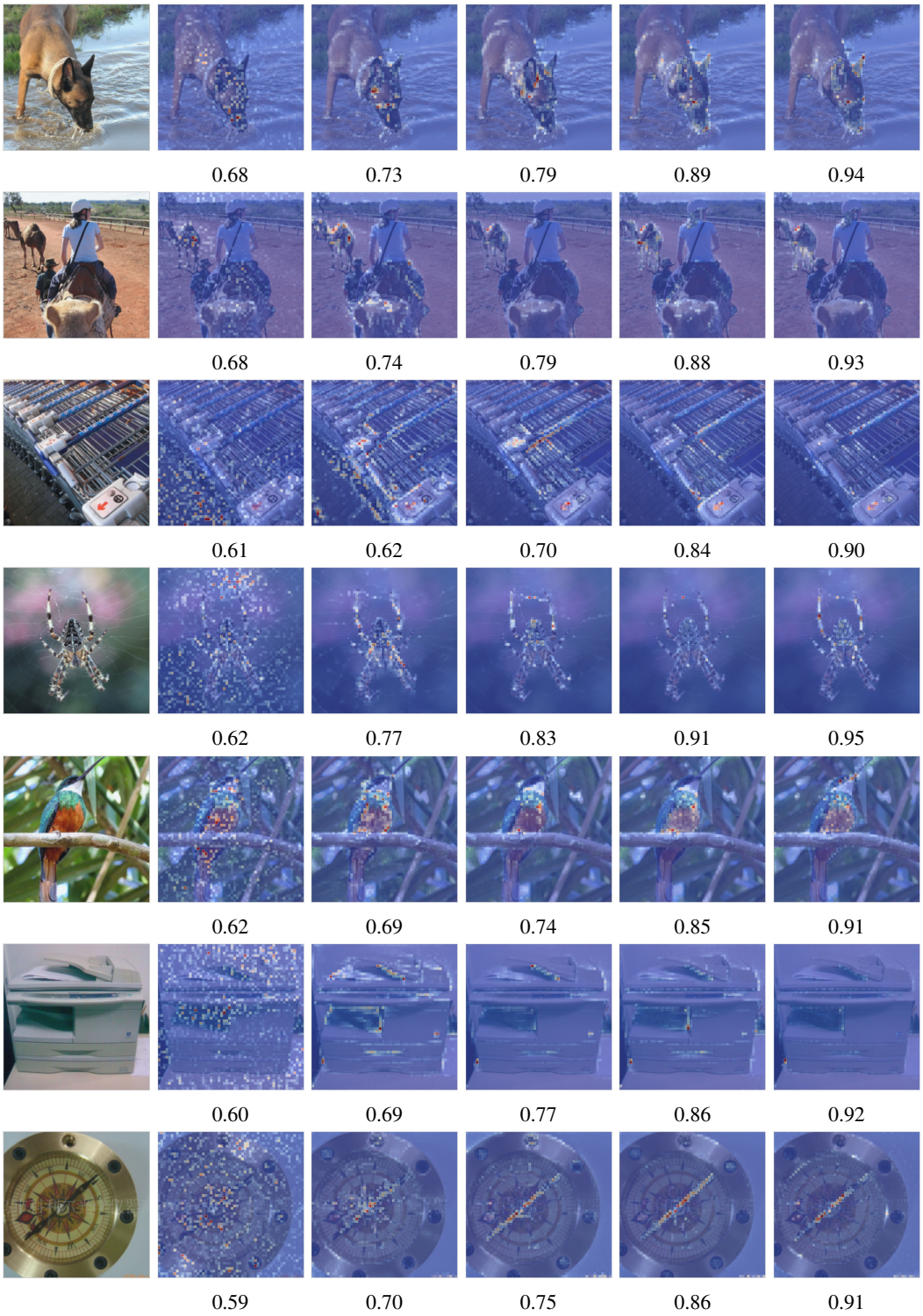

Figure 5: Visualization of feature attributions generated by standard- and adversarially trained model with different ϵ.