On the Sensitivity of Adversarial Robustness to Input Data Distributions

Anonymous submission

Abstract

Neural networks are vulnerable to small adversarial perturbations. While existing literature largely focused on the vulnerability of learned models*, we demonstrate an intriguing phenomenon that adversarial robustness, unlike clean accuracy, is sensitive to the* input data distribution*. Even a semantics-preserving transformations on the input data distribution can cause a significantly different robustness for the adversarially trained model that is both trained and evaluated on the new distribution. We show this by constructing semantically-identical variants for MNIST and CIFAR10 respectively, and show that standardly trained models achieve similar clean accuracies on them, but adversarially trained models achieve significantly different robustness accuracies. This counter-intuitive phenomenon indicates that input data distribution alone can affect the adversarial robustness of trained neural networks, not necessarily the tasks themselves. Lastly, we discuss the practical implications on evaluating adversarial robustness, and make initial attempts to understand this complex phenomenon.*

1. Introduction

We study the relationship between adversarial robustness and the input data distribution. We focus on the adversarial training method [3], arguably the most popular defense method so far due to its simplicity, effectiveness and scalability. Our main contribution is the finding that adversarial robustness is highly sensitive to the input data distribution:

A semantically-lossless shift on the data distribution could result in a drastically different robustness for adversarially trained models.

Note that this is different from the transferability of a fixed model that is trained on one data distribution but tested on another distribution. Even retraining the model on the new data distribution may give us a completely different adversarial robustness on the same new distribution. This is also in sharp contrast to the clean accuracy of standard training, which, as we show in later sections, is insensitive to such shifts. To our best knowledge, our paper is the first work in the literature that demonstrates such sensitivity.

Such sensitivity raises the question of how to properly evaluate adversarial robustness. In particular, the sensitivity of adversarial robustness suggests that certain datasets may not be sufficiently representative when benchmarking different robust learning algorithms. It also raises serious concerns about the deployment of believed-to-be-robust training algorithm in a real product. In a standard development procedure, various models would be prototyped and measured on the existing data. However, the sensitivity of adversarial robustness makes the truthfulness of the performance estimations questionable, as one would expect future data to be slightly shifted. We illustrate the practical implications in Section 3: the robust accuracy of PGD trained model is sensitive to gamma values of gamma-corrected CIFAR10 images. This indicates that image datasets collected under different lighting conditions may have different robustness properties.

Finally, our finding opens up a new angle and provides novel insights to the adversarial vulnerability problem, complementing several recent works on the issue of data distributions' influences on robustness. [6] hypothesizes that there is an intrinsic tradeoff between clean accuracy and adversarial robustness. Our studies complement this result, showing that there are different levels of tradeoffs depending on the characteristics of input data distribution, under the same learning settings (training algorithm, model and training set size). [4] shows that different data distributions could have drastically different properties of adversarially robust generalization, theoretically on Bernoulli vs mixtures of Gaussians, and empirically on standard benchmark datasets. From the sensitivity perspective, we demonstrate that being from completely different distributions (e.g. binary vs Gaussian or MNIST vs CIFAR10) may not be the essential reason for having large robust-

ness difference. Gradual semantics-preserving transformations of data distribution can also cause large changes to datasets' achievable robustness.

2. Robustness on Datasets Variants with Different Input Distributions

In this section we carefully design a series of datasets and experiments to further study its influence. One important property of our new datasets is that they have different input data distributions $\mathbb{P}(x)$'s while keeping the true classification $\mathbb{P}(y|x)$ reasonably fixed, thus these datasets are only different in a "semantic-lossless" shift. Our experiments reveal an unexpected phenomenon that while standard learning methods manage to achieve stable clean accuracies across different data distributions under "semantic-lossless" shifts, however, adversarial training, arguably the most popular method to achieve robust models, loses this desirable property, in that its robust accuracy becomes unstable even under a "semantic-lossless" shift on the data distribution. We emphasize that different from preprocessing steps or transfer learning, here we treat the shifted data distribution as a new underlying distribution. We both train the models and test the robust accuracies on the same new distribution.

2.1. Smoothing and Saturation

In general, MNIST has a more binary distribution of pixels, while CIFAR10 has a more continuous spectrum of pixel values. We apply different levels of "smoothing" on MNIST to create more CIFAR-like datasets, and different levels of "saturation" on CIFAR10 to create more "binary" ones, as shown in Figure 1a and 1b. Note that we would like to maintain the semantic information of the original data, which means that such operations should be semantics-lossless.

Smoothing is applied on MNIST images, to make images "less binary". Given an image x_i, its smoothed version $\tilde{x_i}^{(s)}$ is generated by first applying average filter of kernel size s to x_i to generate an intermediate smooth image, and then take pixel-wise maximum between x_i and the intermediate smooth image.

Saturation of the image x is denoted by $\widehat{x}^{(p)}$, and the procedure is defined as: $\widehat{x}^{(p)} = \text{sign}(2x - 1)\frac{|2x-1|^{\frac{2}{p}}}{2} + \frac{1}{2}$, where all the operations are pixel-wise and each element of $\widehat{x}^{(p)}$ is guaranteed to be in $[0, 1]$. Saturation is used to generate variants of the CIFAR10 dataset with less centered pixel values. For different saturation level p's, one can see from Figure 1b that $\widehat{x}^{(p)}$ is still semantically similar to x in the same classification task.

2.2. Experimental Setups

We use the smoothing and saturation to manipulate the data distributions of MNIST and CIFAR10, and show empirical results on how data distributions affects robust accuracies of neural networks trained on them. To measure the difficulty of the classification task, we perform standard neural network training and test *accuracies* on clean data. To measure the difficulty to achieve robustness, we perform ℓ_∞ projected gradient descent (PGD) based adversarial training [3] and test *robust accuracies* on adversarially perturbed data. To understand whether low robust accuracy is due to low clean accuracy or vulnerability of model, we also report *robustness w.r.t. predictions*, where the attack is used to perturb against the model's clean prediction, instead of the true label. We use LeNet5 on all the MNIST variants, and use wide residual networks [8] with widen factor 4 and depth 28 for all the CIFAR10 variants. Unless otherwise specified, PGD training on MNIST variants and CIFAR10 variants all follows the settings in [3]. PGD attacks on MNIST variants run with $\epsilon = 0.3$, step size of 0.01 and 40 iterations, and runs with $\epsilon = 8/255$, step size of $2/255$ and 10 iterations on CIFAR10 variants , same as in [3].

2.3. Sensitivity of Robust Accuracy to Data Transformations

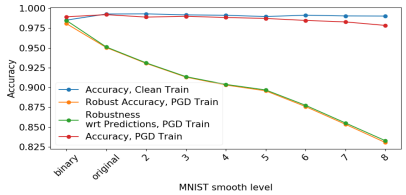
Results on MNIST variants are presented in Figure 1d. The clean accuracy of standard training is very stable across different MNIST variants. This indicates that their classification tasks have similar difficulties, if the training has no robust considerations. When performing PGD adversarial training, clean accuracy drops only slightly. However, both robust accuracy and robustness w.r.t. predictions drop significantly. This indicates that as smooth level goes up, it is significantly harder to achieve robustness. Note that for binarized MNIST with adversarial training, the clean accuracy and the robust accuracy are almost the same. Indicating that getting high robust accuracy on binarized MNIST does not conflict with achieving high clean accuracy.

CIFAR10 result tell a similar story, as reported in Figure 1e. For standard training, the clean accuracy maintains almost at the original level until saturation level 16, despite that it is already perceptually very saturated. In contrast, PGD training has a different trend. Before level 16, the robust accuracy significantly increases from 43.2% until 79.7%, while the clean test accuracy drops only in a comparatively small range, from 85.4% to 80.0%. After level 16, PGD training has almost the same clean accuracy and robust accuracy. However, robustness w.r.t. predictions still keeps increasing, which again indicates the instability of the robustness. On the other hand, if the saturation level is smaller than 2, we get worse robust accuracy after PGD training, e.g. at saturation level 1 the robust ac-

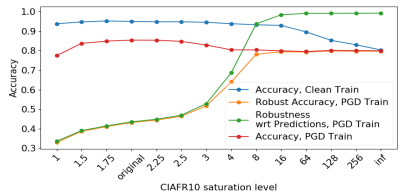
(a) MNIST variants, from left to right: binarized, original, smoothed with kernel size 2, 3, 4, 5

(b) CIFAR10 variants, from left to right, original, saturation level 4, 8, 16, 64, ∞

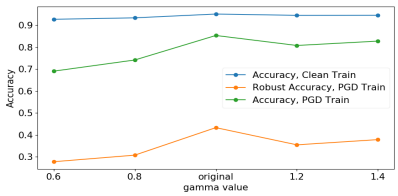
(c) Gamma mapped images from left to right 0.6, 0.8, 1.0 (original image), 1.2 , 1.4



(d) MNIST results under different smooth levels



(e) CIFAR10 results under different saturation levels



(f) Robustness results on gamma mapped CIFAR10 variant

Figure 1: Variants of MNIST and CIFAR10 datasets (a, b, c), and Accuracy, Robust Accuracy and Robustness w.r.t. Predictions on different data variants (c, d, e).

curacy is 33.0%. Simultaneously, the clean accuracy maintains almost the same.

Note that after saturation level 64 the standard training accuracies starts to drop significantly. This is likely due to that high degree of saturation has caused "information loss". Models trained on highly saturated CIFAR10 are quite robust and the gap between robust accuracy and robustness w.r.t. predictions is due to lower clean accuracy. In contrast, In MNIST variants, the robustness w.r.t. predictions is always almost the same as robust accuracy, indicating that drops in robust accuracy is due to adversarial vulnerability.

From these results, we can conclude that robust accuracy under PGD training is much more sensitive than clean accuracy under standard training to the differences in input data distribution. More importantly, a semantically-lossless shift on the data transformation, while not introducing any unexpected risk for the clean accuracy of standard training, can lead to large variations in robust accuracy. Such previously unnoticed sensitivity raised serious concerns in practice, as discussed in the next section.

3. Sensitivity to Image Acquisition Condition and Preprocessing

The natural images are acquired under different lighting conditions, with different cameras and different camera settings. They are usually preprocessed in different ways. All these factors could lead to mild shifts on the input distribution. Therefore, we might get very different performance measures when performing adversarial training on images taken under different conditions. In this section, we demonstrate this phenomenon on variants of CIFAR10 images under different gamma mappings. These variants are then used to represent image dataset acquired under different conditions. Gamma mapping is a simple element-wise operation that takes the original image x, and output the gamma mapped image $\tilde{x}^{(\gamma)}$ by performing $\tilde{x}^{(\gamma)} = x^{\gamma}$. Gamma mapping is commonly used to adjust the exposure of an images. We refer the readers to [5] on more details about gamma mappings. Figure 1c shows variants of the same image processed with different gamma values. Lower gamma value leads to brighter images and higher gamma values gives darker images, since pixel values range from 0 to 1. Despite the changes in brightness, the semantic information is preserved.

We perform the same experiments as in the saturated CIFAR10 variants experiment in Section 2, with results displayed in Figure 1f. Clean accuracies almost remain the same across different gamma values. However, under PGD training, both accuracy and robust accuracy varies largely under different gamma values.

These results should raise practitioners' attention on how to interpret robustness benchmark "values". For the same adversarial training setting, the robustness measure might change drastically between image datasets with different "exposures". In other words, if a training algorithm achieves good robustness on one image dataset, it doesn't necessarily achieve similar robustness on another semantically-identical but slightly varied datasets. Therefore, the actual robustness could be underestimated or overestimated significantly. This raises the questions on whether we are evaluating image classifier robustness in a reliable way, and how we choose benchmark settings that can match the real robustness requirements in practice. We defer this important open question to future research.

Table 1: PGD attack results with and without domain boundary constraints on MNIST and CIFAR10

MNIST			CIFAR10		
MNIST variants	Robust Accuracy w/ bound	Robust Accuracy w/o bound	CIFAR10 variants	Robust Accuracy w/ bound	Robust Accuracy w/o bound
binarized	98.1 %	96.1 %	saturate 1	33.0 %	32.7 %
original	95.1 %	95.1 %	original	43.2 %	43.0 %
smooth 2	93.0 %	92.9 %	saturate 4	64.0 %	64.0 %
smooth 3	91.3 %	91.5 %	saturate 8	78.1 %	78.1 %
smooth 4	90.3 %	90.6 %	saturate 16	79.4 %	79.4 %
smooth 5	89.6 %	89.9 %	saturate inf	79.7 %	79.4 %

Table 2: Different robust accuracies on datasets with same inter-class distances

Inter-class Distances	Smooth level of Smoothed MNIST	Resilience of Smoothed MNIST	Scale factor of Scaled Original MNIST	Resilience of Scaled Original MNIST	Scale factor of Scaled Binarized MNIST	Resilience of Scaled Binarized MNIST
7.12	3	91.3 %	0.970	94.6 %	0.821	98.6 %
7.01	4	90.3 %	0.955	95.5 %	0.809	98.6 %
6.85	5	89.6 %	0.932	94.9 %	0.790	98.5 %

4. Attempts to Understand the Phenomenon

4.1. On the Influence of Perturbable Volume

Saturation moves the pixel values towards 0 and 1, therefore pushing the data points to the corners of the unit cube input domain. This makes the valid perturbation space to be smaller, since the space of perturbation is the intersection between the ϵ-ℓ_∞ ball and the input domain. Due to high dimensionality, the volume of "perturbable region" changes drastically across different saturation levels. For example, the average log perturbable volume [1] of original CIFAR10 images are -12354, and the average log perturbable volume of ∞-saturated CIFAR10 is -15342, which means that the perturbable volume differs by a factor of $2^{2990} = 2^{(-12352-(-15342))}$. If the differences in perturbable volume is a key factor on the robustness' sensitivity, then by allowing the attack to go beyond the domain boundary [2], the robust accuracies across different saturation levels should behave similarly again, or at least significantly differ from the case of box constrained attacks. We performed PGD attack allowing the perturbation to be outside of the data domain boundary, and compare the robust accuracy to what we get for normal PGD attack within domain boundary. We found that the expected difference is not observed, in Table 1, which serves as evidence that differences in perturbable volume are not causing the differences in robustness on the tested MNIST and CIFAR10 variants.

[1] Definition of "log perturbable volume" and other detailed analysis of perturbable volume are given in Appendix C.1.

[2] So we have a controlled and constant perturbable volume across all cases, where the volume is that of the ϵ-ℓ_∞ ball

4.2. On the Influence of Inter-Class Distance

When saturation pushes data points towards data domain boundaries, the distances between data points increase too. Therefore, the margin, the distance from data point to the decision boundary, could also increase. We use the "inter-class distance" as an approximation. Inter-class distance [3] characterizes the distances between each class to rest of classes in each dataset. Intuitively, if the distances between classes are larger, then it should be easier to achieve robustness. We also observed (in Appendix C.2.1 Figure 2) that inter-class distances are positively correlated with robust accuracy. However, we also find counter examples where datasets having the same inter-class distance exhibit different robust accuracies. Specifically, We construct scaled variants of original MNIST and binarized MNIST, such that their inter-class distances are the same as smooth-3, smooth-4, smooth-5 MNIST. The scaling operation is defined as $\tilde{x}^{(\alpha)} = \alpha(x - 0.5) + 0.5$, where α is the scaling coefficient. When $\alpha < 1$. each dimension of x is pushed towards the center with the same rate. Table 2 shows the results. We can see that although having the same interclass distances, the smoothed MNIST is still less robust than the their correspondents of scaled binarized MNIST and original MNIST. This indicates the complexity of the problem, such that a simple measure like inter-class distance cannot fully characterize robustness property of datasets, at least on the variants of MNIST.

[3] The calculation of "inter-class distance" and other detailed analyses are delayed to Appendix C.2.1 and Fig 2. Also note that our inter-class distance is similar to the "distinguishability" in [1], which also measures the distance between classes to quantify easiness of achieving robustness on a certain dataset.

References

[1] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Analysis of classifiers' robustness to adversarial perturbations. *arXiv preprint arXiv:1502.02590*, 2015.

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

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

[4] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Mądry. Adversarially robust generalization requires more data. *arXiv preprint arXiv:1804.11285*, 2018.

[5] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[6] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

[7] David Warde-Farley and Ian Goodfellow. 11 adversarial perturbations of deep neural networks. page 311, 2016.

[8] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

A. Detailed Settings for Training

A.1. Detailed settings of adversarial training

The LeNet5 (widen factor 1) is composed of 32-channel conv filter + ReLU + size 2 max pooling + 64-channel conv filter + ReLU + size 2 max pooling + fc layer with 1024 units + ReLU + fc layer with 10 output classes. We do not preprocess MNIST images before feeding into the model.

For training LeNet5 on MNIST variants, we use the Adam optimizer with an initial learning rate of 0.0001 and train for 100000 steps with batch size 50.

We use the WideResNet-28-4 as described in [8] for our experiments, where 28 is the depth and 4 is the widen factor. We use "per image standardization" [4] to preprocess CIFAR10 images, following [3].

For training WideResNet on CIFAR10 variants, we use stochastic gradient descent with momentum 0.9 and weight decay 0.0002. We train 80000 steps in total with batch size 128. The learning rate is set to 0.1 at step 0, 0.01 at step 40000, and 0.001 at step 60000.

We performed manual hyperparameter search for our initial experiment and do not observe improvements over the above settings. Therefore we used these settings throughout the all the experiments in the paper unless otherwise indicated.

[4] https://www.tensorflow.org/api_docs/python/tf/image/per_image_standardization

B. Detailed Experimental Results

We listed exact numbers of experiments involved in the main body in Table 3, 4, 5 and 6.

C. Detailed Analyses

C.1. Detailed Analysis of Effects of Data Domain Boundary

One natural hypothesis about the reason of achieving better robustness could be that it is the effect of the boundaries. Indeed, if the data distribution is closer to the data domain boundary, the valid perturbation space, the ϵ-ℓ_∞ ball may be restricted since it will intersect with the boundary. We then test the correlation between "how close the data distribution is to the boundary" and its achievable robustness, by examining the volume of the allowed perturbed box across different datasets.

The intersection of the data domain, unit cube $[0,1]^d$, with the allowed perturbation space, ϵ-ℓ_∞ ball $[x_i-\epsilon, x_i+\epsilon]^d$, is the hyperrectangle $[\max\{x_i-\epsilon,0\}, \min\{x_i+\epsilon,1\}]^d$, where $i=1,\cdots,d$ are the indexes over input dimensions. The size of the available perturbation space at x and ϵ is defined by the volume of this hyperrectangle:

$$\mathrm{Vol}(x,\epsilon)=\prod_{i=1}^{d}(\min\{x_i+\epsilon_i,1\}-\max\{x_i-\epsilon_i,0\})$$

In high dimensional space, when ϵ is fixed, this volume varies greatly based on the location of x. For example, if x is on one of the corners of the unit cube, $\mathrm{Vol}(x_{corner},\epsilon)=\epsilon^d$. If each dimension of x is at least ϵ away from all the data boundaries, then the volume of the hyperrectangle is $\mathrm{Vol}(x_{inside},\epsilon)=(2\epsilon)^d$. Therefore there can be 2^d times difference of perturbable space between different data points. As shown in the average log perturbable volumes Table 7, we can see that different variations of datasets has significantly different perturbable volumes, with the same trend with previously described. It is notable that for the original CIFAR10 datasets has log volume -12354, which is very close to the -12270. The different of 84 bits indicates on average, the perturbation space is 2^{84} smaller than the full ϵ-ℓ_∞ ball if there is no intersection with the data domain boundary. Volume differences between different saturation or smooth level can be interpreted in the similar way. Note that for CIFAR10 images with large saturation, although they appear similar to human, they actually have very large differences in terms of perturbable volumes.

If the perturbable volume hypothesis holds, then we should observe significantly lower accuracy under PGD

Table 3: Performance and Robustness of models trained on MNIST variants.

	Standard Training	PGD Training		
MNIST variants	Test Acc	Test Acc	Robust Accuracy $\epsilon = 0.3$	Robustness w.r.t. Predictions $\epsilon = 0.3$
binarized	98.5 %	98.9 %	98.1 %	98.5 %
original	99.3 %	99.2 %	95.1 %	95.1 %
smooth 2	99.3 %	98.9 %	93.0 %	93.1 %
smooth 3	99.2 %	99.0 %	91.3 %	91.4 %
smooth 4	99.1 %	98.8 %	90.3 %	90.4 %
smooth 5	99.0 %	98.7 %	89.6 %	89.7 %
smooth 6	99.1 %	98.5 %	87.6 %	87.7 %
smooth 7	99.0 %	98.3 %	85.4 %	85.5 %
smooth 8	99.0 %	97.9 %	83.1 %	83.3 %

Table 4: Performance and Robustness of models trained on CIFAR10 variants.

	Standard Training	PGD Training		
CIFAR10 variants	Test Acc	Test Acc	Robust Accuracy $\epsilon = 8/255$	Robustness w.r.t. Predictions $\epsilon = 8/255$
saturate 1	93.8 %	77.5 %	33.0 %	33.6 %
saturate 1.5	94.7 %	83.7 %	38.7 %	39.1 %
saturate 1.75	95.2 %	84.9 %	41.1 %	41.5 %
original	95.0 %	85.4 %	43.2 %	43.6 %
saturate 2.25	94.8 %	85.4 %	44.4 %	44.9 %
saturate 2.5	94.8 %	84.8 %	46.4 %	47.0 %
saturate 3	94.5 %	82.9 %	51.7 %	52.9 %
saturate 4	93.8 %	80.4 %	64.0 %	68.7 %
saturate 8	93.3 %	80.4 %	78.1 %	93.8 %
saturate 16	92.9 %	79.9 %	79.4 %	98.4 %
saturate 64	89.6 %	79.5 %	79.3 %	99.1 %
saturate 128	85.3 %	80.2 %	79.9 %	99.1 %
saturate 256	83.0 %	80.0 %	79.7 %	99.2 %
saturate inf	80.3 %	80.0 %	79.7 %	99.2 %

attack if we allow perturbation outside of data domain boundary. Since this greatly increases the perturbable volume. We measure the accuracy under PGD attack with and without considering data domain boundary for both MNIST and CIFAR10 variants. The results are shown in Table 1. "With considering boundary" corresponds to regular PGD attacks. We can see that allowing PGD to perturb out of bound do not reduce accuracy under attack. This means that PGD is not able to use the significantly larger additional volumes even for binarized MNIST or highly saturated CIFAR10, whose data points are on or very close to the corner. In some cases, allowing perturbation outside of domain boundary makes the attack slightly less effective. This might be due to that data domain boundary constrained the perturbation to be in an "easier" region. This might seem surprising considering the huge difference in perturbable volumes, these results conform with empirical results in previous research [2, 7] that adversarial examples appears in certain directions instead of being distributed in small pockets across space. Therefore, the perturbable volume hypothesis is rejected.

C.2. Detailed Analyses of Inter-class Distance

C.2.1 Calculation of Inter-class Distance

We calculate the inter-class distance as follows. Let $D = \{x_i\}$ denote the set of all the input data points, $D_c = \{x_i | y_i = c\}$ denote the set of all the data points in class c, and $D_{\neg c} = \{x_i | y_i \neq c\}$ denote all the data points not in class c. Our goal is to calculate $d(D_c, D_{\neg c})$ for all the classes, where $d(D_c, D_{\neg c})$ approximates the margin between class c and the rest. To estimate $d(D_c, D_{\neg c})$, we first compute the margin for each data point x in class c. To do that, we calculate the average $\|x - x_j\|_2$, where $x_j \in D_{\neg c}$ is one of x's 10% nearest neighbors in $D_{\neg c}$. Lastly, the inter-class distance of class c, $d(D_c, D_{\neg c})$, is then calculated as the average of smallest 10% $d(x, D_{\neg c})$ for $x \in D_c$.

Note that we choose ℓ_2 distance for inter-class distance, instead of using the ℓ_∞ which measures the robustness. This is because ℓ_∞-distance between data examples is essentially the max over the per pixel differences, which is always very close to 1. Therefore the ℓ_∞-distance between data examples is not really representative / distinguishable.

Figure 2 shows the inter-class distances (averaged over all classes) calculated on MNIST and CIFAR10 variants. The binarized MNIST has a significantly

Table 5: Performance and robustness of different sized LeNet5 models on MNIST variants

Standard Training, Accuracy	Training Set						Test Set					
Widen factor	0.125	0.25	0.5	1	2	4	0.125	0.25	0.5	1	2	4
binarized	99.9%	100.0%	100.0%	99.6%	100.0%	100.0%	98.7%	99.0%	99.2%	98.5%	99.4%	99.2%
original	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	98.8%	99.2%	99.2%	99.3%	99.4%	99.3%
smooth 2	99.9%	100.0%	100.0%	100.0%	100.0%	100.0%	98.8%	99.0%	99.1%	99.3%	99.3%	99.4%
smooth 3	99.9%	99.9%	100.0%	100.0%	100.0%	100.0%	98.8%	98.8%	99.2%	99.2%	99.1%	99.3%
smooth 4	99.9%	100.0%	100.0%	100.0%	100.0%	100.0%	98.7%	99.0%	99.0%	99.1%	99.4%	99.4%
smooth 5	99.8%	100.0%	100.0%	100.0%	100.0%	100.0%	98.5%	99.0%	99.2%	99.0%	99.3%	99.3%
smooth 6	99.8%	100.0%	100.0%	100.0%	100.0%	100.0%	98.4%	98.9%	99.0%	99.1%	99.2%	99.3%
smooth 7	99.8%	99.9%	100.0%	100.0%	100.0%	100.0%	98.5%	98.8%	99.0%	99.0%	99.3%	99.3%
smooth 8	99.7%	100.0%	100.0%	100.0%	100.0%	100.0%	98.4%	98.9%	98.9%	99.0%	99.2%	99.0%

PGD Training, Accuracy	Training Set						Test Set					
Widen factor	0.125	0.25	0.5	1	2	4	0.125	0.25	0.5	1	2	4
binarized	97.8%	99.6%	100.0%	100.0%	100.0%	100.0%	97.4%	98.3%	98.8%	98.9%	99.0%	99.2%
original	97.0%	98.4%	99.8%	100.0%	100.0%	100.0%	97.0%	98.2%	98.9%	99.2%	99.1%	99.2%
smooth 2	96.1%	98.1%	99.0%	99.9%	100.0%	100.0%	96.1%	97.8%	98.5%	98.9%	99.0%	99.0%
smooth 3	96.3%	97.8%	98.9%	99.7%	99.9%	100.0%	96.5%	97.6%	98.6%	99.0%	99.1%	99.1%
smooth 4	95.3%	97.3%	98.5%	99.5%	99.8%	99.9%	95.4%	97.2%	98.1%	98.8%	99.0%	99.0%
smooth 5	94.9%	96.5%	98.0%	99.3%	99.6%	99.8%	95.0%	96.5%	97.9%	98.7%	98.9%	98.9%
smooth 6	93.2%	95.6%	97.4%	99.0%	99.5%	99.7%	93.5%	95.7%	97.1%	98.5%	98.7%	98.7%
smooth 7	91.9%	95.0%	97.5%	98.7%	99.2%	99.4%	92.4%	95.2%	97.2%	98.3%	98.5%	98.7%
smooth 8	89.4%	94.2%	96.5%	98.4%	99.0%	99.3%	89.7%	94.4%	96.4%	97.9%	98.2%	98.4%

PGD Training, Robust Accuracy	Training Set						Test Set					
Widen factor	0.125	0.25	0.5	1	2	4	0.125	0.25	0.5	1	2	4
binarized	95.2%	98.5%	100.0%	100.0%	100.0%	100.0%	94.5%	96.5%	98.0%	98.1%	98.0%	98.0%
original	86.9%	90.8%	97.9%	99.3%	99.6%	99.8%	87.1%	89.9%	95.2%	95.1%	94.8%	94.9%
smooth 2	80.5%	87.6%	90.9%	98.0%	99.1%	99.5%	81.2%	87.0%	88.7%	93.0%	92.3%	92.1%
smooth 3	75.2%	82.0%	90.3%	95.5%	97.8%	98.7%	75.7%	81.5%	88.5%	91.3%	91.6%	90.8%
smooth 4	71.9%	77.6%	87.5%	93.9%	96.8%	97.9%	72.7%	77.7%	86.3%	90.3%	90.6%	90.0%
smooth 5	65.7%	77.1%	85.7%	92.5%	94.6%	95.0%	66.2%	77.1%	85.1%	89.6%	89.8%	88.4%
smooth 6	58.0%	71.5%	80.5%	90.6%	93.1%	93.8%	59.3%	72.0%	80.2%	87.6%	88.0%	87.2%
smooth 7	61.7%	74.2%	83.3%	87.6%	90.5%	92.6%	62.8%	75.3%	83.0%	85.4%	86.7%	87.8%
smooth 8	70.3%	72.4%	80.3%	85.3%	90.5%	88.7%	71.7%	73.2%	80.3%	83.1%	86.9%	83.8%

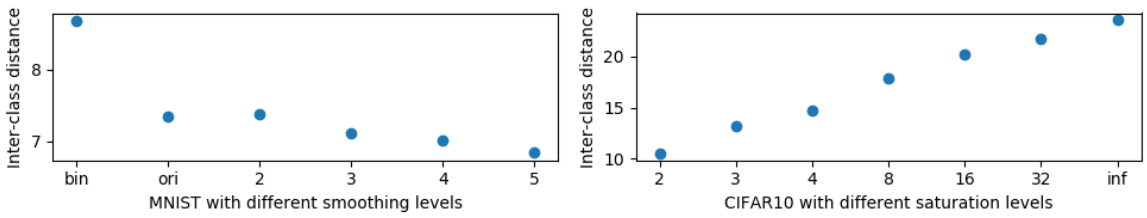
larger inter-class distance. As smoothing kernel size increases, the distance also decrease slightly. On CIFAR10 variants, as the saturation level gets higher, the inter-class distance increases monotonically. We also directly plot inter-class distance vs robust accuracy on MNIST and CIFAR10 variants. In general, inter-class distance shows a strong positive correlation with robust accuracy under these transformations. With one exception that original MNIST has smaller inter-class distance, but is sightly more robust than smooth-2 MNIST. This, together with the counter examples we gave in Table 2, suggests that inter-class distance cannot fully explain the robust variation across different dataset variants.

Table 6: Performance and robustness of different sized Wide ResNet models on CIFAR10 variants

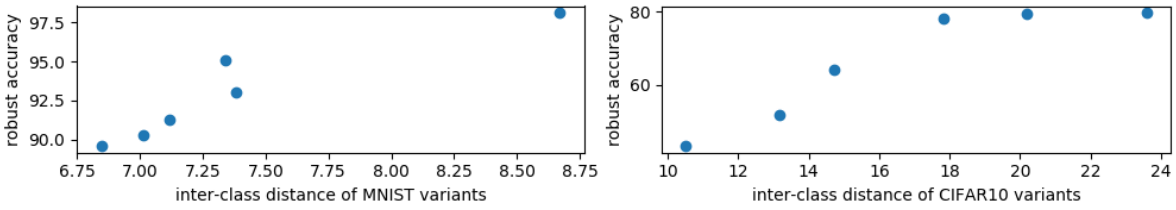
	Standard Training, Accuracy					
	Training Set			Test Set		
Widen factor	0.25	1	4	0.25	1	4
saturate 1	85.5%	99.9%	100.0%	82.4%	91.1%	93.8%
saturate 1.5	87.0%	99.9%	100.0%	84.2%	92.1%	94.7%
saturate 1.75	87.4%	99.9%	100.0%	84.5%	93.0%	95.2%
original	87.2%	99.9%	100.0%	84.4%	92.5%	95.0%
saturate 2.25	87.3%	99.9%	100.0%	84.5%	92.5%	94.8%
saturate 2.5	86.4%	99.9%	100.0%	83.7%	92.3%	94.8%
saturate 3	86.2%	99.9%	100.0%	84.0%	92.2%	94.5%
saturate 4	85.8%	99.9%	100.0%	83.1%	91.1%	93.8%
saturate 8	84.6%	99.8%	100.0%	81.2%	90.1%	93.3%
saturate 16	83.5%	99.7%	100.0%	81.0%	89.4%	92.9%
saturate 64	80.5%	99.4%	100.0%	79.2%	86.9%	89.6%
saturate 128	77.1%	98.7%	100.0%	74.6%	83.0%	85.3%
saturate 256	73.7%	97.6%	100.0%	70.7%	76.5%	83.0%
saturate inf	73.2%	97.3%	99.9%	70.6%	76.3%	80.3%
	PGD Training, Accuracy					
	Training Set			Test Set		
Widen factor	0.25	1	4	0.25	1	4
saturate 1	45.4%	68.3%	93.1%	46.8%	66.9%	77.5%
saturate 1.5	52.1%	76.5%	98.0%	53.3%	74.1%	83.7%
saturate 1.75	53.8%	79.5%	99.2%	55.3%	77.0%	84.9%
original	56.1%	81.4%	99.7%	57.1%	78.4%	85.4%
saturate 2.25	56.8%	82.7%	99.9%	58.1%	78.8%	85.4%
saturate 2.5	57.6%	83.9%	100.0%	58.3%	79.1%	84.8%
saturate 3	60.0%	86.3%	100.0%	60.8%	79.5%	82.9%
saturate 4	62.8%	91.3%	100.0%	63.7%	77.9%	80.4%
saturate 8	67.7%	96.1%	100.0%	67.0%	76.6%	80.4%
saturate 16	67.2%	96.1%	99.9%	66.0%	76.4%	79.9%
saturate 64	70.0%	96.5%	99.9%	68.6%	75.8%	79.5%
saturate 128	71.4%	96.4%	99.9%	68.9%	76.6%	80.2%
saturate 256	68.6%	96.9%	99.9%	65.7%	76.6%	80.0%
saturate inf	71.5%	96.9%	99.9%	69.7%	76.1%	80.0%
	PGD Training, Robust Accuracy					
	Training Set			Test Set		
Widen factor	0.25	1	4	0.25	1	4
saturate 1	24.0%	36.9%	71.1%	25.6%	34.4%	33.0%
saturate 1.5	29.0%	44.4%	81.3%	31.6%	40.7%	38.7%
saturate 1.75	30.9%	47.8%	86.0%	32.7%	44.0%	41.1%
original	32.4%	50.4%	90.3%	35.0%	45.5%	43.2%
saturate 2.25	33.9%	52.9%	93.4%	36.1%	47.3%	44.4%
saturate 2.5	35.5%	55.4%	96.0%	37.5%	49.1%	46.4%
saturate 3	38.4%	61.5%	98.9%	40.6%	52.5%	51.7%
saturate 4	44.9%	77.4%	99.7%	46.1%	60.4%	64.0%
saturate 8	62.3%	95.0%	99.8%	61.9%	74.9%	78.1%
saturate 16	66.0%	95.5%	99.9%	65.0%	75.5%	79.4%
saturate 64	69.1%	96.3%	99.9%	67.6%	75.5%	79.3%
saturate 128	70.7%	96.2%	99.9%	68.2%	76.2%	79.9%
saturate 256	68.0%	96.7%	99.9%	65.2%	76.3%	79.7%
saturate inf	70.9%	96.7%	99.9%	69.2%	75.8%	79.7%

Table 7: Perturbable volumes of different variants of MNIST and CIFAR10. Values shown in table are the average log value (in bits) of volumes of test data. For MNIST, $\epsilon = 0.3$, for CIFAR10 $\epsilon = 8/255$.

MNIST (valid range -1361 to -577)				CIFAR10 (valid range -15342 to -12270)							
binary	original	3	5	original	4	8	16	64	256	512	inf
-1361	-1297	-1265	-1234	-12354	-12394	-12477	-12657	-13620	-14747	-15028	-15342



(a) Processing levels vs inter-class distances



(b) Inter-class distance vs robust accuracy

Figure 2: Inter-class distance's influence on robust accuracy on different MNIST and CIFAR10 variants