ON THE SENSITIVITY OF ADVERSARIAL ROBUSTNESS TO INPUT DATA DISTRIBUTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks are vulnerable to small adversarial perturbations. Existing literature largely focused on understanding and mitigating the vulnerability of learned models. In this paper, we demonstrate an intriguing phenomenon about adversarial training, arguably the most popular robust training method in the literature: 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 adversarial trained model that is both trained and evaluated on the new distribution. Our discovery of such sensitivity on data distribution is based on a study which disentangles the behaviors of clean accuracy and robust accuracy of the Bayes classifier. Further empirical investigation on MNIST and CIFAR10 confirms our finding that numerous MNIST and CIFAR10 variants achieve comparable clean accuracies in various different neural nets under standard training but significantly different robustness under adversarial training. 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

Neural networks are vulnerable to adversarial examples. An otherwise highly accurate network can be greatly fooled by artificially constructed small perturbations on the original samples that don't affect human's perception (Szegedy et al., 2013; Biggio et al., 2013). Since the first discovery of adversarial examples, great progresses in constructing stronger adversarial attacks have been made in the literature, from the fast gradient sign method (Goodfellow et al., 2014), DeepFool (Moosavi-Dezfooli et al., 2016), projected gradient descent (PGD) (Kurakin et al., 2016; Madry et al., 2017), to the Carlini and Wagner (2017) optimizer-based attack, among others. In contrast, defense methods for such adversarial attacks are relatively less successful in the literature. Many defense methods were bypassed by the attackers (Carlini and Wagner, 2016; Athalye et al., 2017; 2018), because they do not increase the "intrinsic" robustness of the model. Notable defense methods include provably robust methods (Sinha et al., 2017; Raghunathan et al., 2018; Kolter and Wong, 2017), and adversarial training (Madry et al., 2017; Erraqabi et al., 2018; Huang et al., 2015; Kurakin et al., 2016). Arguably, adversarial training so far is the most popular defense method due to its simplicity, effectiveness and scalability.

This paper starts the investigation on the difficulty in achieving adversarial robustness, motivated by the empirical observations on the MNIST dataset and the CIFAR10 dataset. In particular, while MNIST and CIFAR10 have similar SOTA clean accuracies (difference is less than 3%) (Gastaldi, 2017), CIFAR10 has a much lower achievable robustness than MNIST in practice. After PGD adversarial training (Madry et al., 2017), MNIST has 89.3% accuracy under perturbation δ with $\|\delta\|_\infty \leq 0.3$ under the strongest PGD attack, but CIFAR10 has only 45.8% accuracy under a much smaller δ with $\|\delta\|_\infty \leq 8/255$. The accuracy difference is more than 40%, even without considering the difference in perturbation lengths.

We first perform theoretical analysis with the tools from the concentration properties of (probability) measures. We show the invariance of regular Bayes error to invertible transformation of data distribution and illustrate that robust error of the perfect decision boundary do not enjoy this property. We also given an example where robustness is not achievable: if input data is uniformly distributed in a

unit cube, then the prefect decision boundary in terms of clean accuracy cannot be robust under ℓ_2 attack. In addition, we show provably robust models can be trained to have only 3% provable robust error on binarized MNIST (semantically almost identical to MNIST). At the same time, this cannot be achieved on regular MNIST data. Aligning the theoretical analyses and further empirical studies, we notice the important role of the data distribution in adversarial robustness.

The second contribution of this paper is a series of carefully designed experiments on variants of MNIST and CIFAR10 datasets to study the influence of the data distribution on the achieved adversarial robustness of adversarial training, as shown in Figure 1. Our results lead to an intriguing property of adversarial training:

A semantically lossless shift on the data distribution could result in a significantly different robustness for adversarial trained models.

Such property is really undesirable and raises serious concerns about the deployment of a believed-to-be-robust model in a real product. Indeed in a standard development procedure, various models (for example different network architectures) would be prototyped and measured on the existing data. If adversarial training is employed to train the robust models, the truthfulness of the performance estimations then becomes questionable, as one would expect that future data will be slightly shifted. Note that this is different from the transferability of a fixed model that is trained on different data distribution. Even retraining the model on the new data may give you a completely different accuracy and robustness on the same new distribution. This phenomenon was previously unnoticed largely because clean accuracy of standard training is not sensitive to such shifts, as we show in later sections. We particularly illustrate the practical implications with two examples: 1) robust accuracy of PGD trained model is sensitive to gamma values of gamma corrected CIFAR10 images. This indicates that image dataset collected under different conditions will have different robustness properties; 2) both as a "harder" version of MNIST, the fashion-MNIST (Xiao et al., 2017) and edge-fashion-MNIST (an edge detection variant described in Section 4.2) exhibit completely different robust characteristics. This suggests that more care is needed on choosing the right dataset to evaluate ant match adversarial robustness in practical tasks.

The last part is an initial attempt to understand how the data distribution may influence robustness. Such knowledge may help us mitigate the aforementioned "unstable" measurement problem. We specifically look at perturbable volume, inter-class distances, and required model capacity and sample complexity. Our analysis shows that understanding this intriguing phenomenon is non-trivial.

1.1 Related Work

Existing efforts mostly focused on analyzing and improving classification models, and only a few recent papers touch on the issue of data distributions' influences on robustness. Schmidt et al. (2018) proves that it requires significantly fewer samples to achieve adversarially robust generalization on Bernoulli distributed data than on data from mixtures of Gaussian. Empirically, they also show that the lack of data could be the reason that we cannot achieve higher robustness on CIFAR10. Similarly, our focus is on adversarial robustness' sensitivity to changes in input data distribution. We show that robustness difference not only exists in binary v.s. continuous distribution, or MNIST v.s. CIFAR10, but gradual semantics-preserving transformations of data distribution can cause large changes to datasets' achievable robustness. Tsipras et al. (2018) hypothesizes that data has robust and non-robust features which poses an intrinsic tradeoff between clean accuracy and adversarial robustness. We extend this perspective and show that under standard learning settings (training algorithm, model and training set size), there are different levels of tradeoff depending on the characteristics of input data distribution. Fawzi et al. (2015) defines "distinguishability", the distance between classes, to quantify easiness of achieving robustness on a certain dataset. In later sections, we also define a similar quantity, inter-class distance, exploring the robustness differences between dataset variants. While intuitively plausible, we concretely show counter examples such that datasets with the same inter-class distance could have different robustness properties. And we believe that it is unlikely a simple distance measures can explain the robustness variation across datasets.

1.2 Notation and Problem Setup

We specifically consider the image classification problem where the input data is inside a high dimensional unit cube. We denote the data distribution as a joint distribution $\mathbb{P}(X, Y)$, where $X \in [0, 1]^d$, d is the number of pixels, and $Y \in \{1, 2, \ldots, k\}$ is the discrete label. Note that here we assume the support of X is the whole pixel space $[0, 1]^d$. When X is a random noise (or human

perceptually unclassifiable image), one can think of $\mathbb{P}(y \mid X)$ being closed to uniform distribution on labels. In the standard setting, the samples (x_i, y_i) can be interpreted as x_i is independently sampled from the marginal distribution $\mathbb{P}(X)$, and then y_i is sampled from $\mathbb{P}(Y \mid X = x_i)$. In this paper, we discuss $\mathbb{P}(X)$'s influences on adversarial robustness, given a fixed $\mathbb{P}(y|X)$.

In our experiments, we only discuss the whitebox robustness, as it represents the "intrinsic" robustness. We use models learned by adversarially augmented training (Madry et al., 2017) (PGD training), which has the SOTA whitebox robustness. We consider bounded ℓ_∞ attack as the attack for evaluating robustness for 2 reasons: 1) PGD training can defend against ℓ_∞ relatively well, while for other attacks, how to train a robust model is still an open question; 2) in the image domain ℓ_∞ attack is the mostly widely researched attack.

2 Theoretical Analyses and Provable Cases

As mentioned in the introduction, although the SOTA clean accuracies are similar for MNIST and CIFAR10, empirical investigations suggest that robust accuracy on CIFAR10 is much more difficult to achieve, which indicates the different behaviors of clean accuracy and robust accuracy. The first result in this section is to further confirm this indication in a simple setting, where clean accuracy remains the same but robust accuracy completely changes under distribution shift. Based on results from the concentration of measure literature, we then show that under uniform distribution basically no algorithm can achieve good robustness, as long as they have high clean accuracy. Such theoretical results are in sharp contrast to our practical experience. Therefore, we further examine the performance of a verifiable defense method on binarized MNIST (pixels values rounded to 0 and 1), and the result suggests the exact opposite: provable adversarial robustness on MNIST-like dataset is achievable. Such contrast thus clearly suggests the important role of the data distribution in achieving adversarial robustness.

2.1 Disentangle Clean Accuracy and Robust Accuracy

Let $\mathcal{H}$ denote the universal set of all the measurable functions. Given a joint distribution $\mathbb{P}(\mathrm{x}, \mathrm{y})$ on the space $\mathcal{X} \times \mathcal{Y}$, we define the Bayes error $R^* = \inf_{h \in \mathcal{H}} \mathbb{E}_{\mathbb{P}(\mathrm{x},\mathrm{y})} L(\mathrm{y}; h(\mathrm{x})) = R^*(\mathbb{P}(\mathrm{x}, \mathrm{y}))$, where L is the objective function. In other words, Bayes error is the error of the best possible classifier we can have, h^*, without restriction on the function space of classifiers. We further define (adversarial) robust error $RR(h) = \mathbb{E}_{\mathbb{P}(\mathrm{x},\mathrm{y})} \max_{\|\delta\|_\infty < \epsilon} L(\mathrm{y}; h(\mathrm{x} + \delta)) = RR(\mathbb{P}(\mathrm{x}, \mathrm{y}))$. We denote $RR^* = RR(h^*)$ to be the robust error achieved by the Bayes classifier h^*. For simplicity, we assume our algorithm can always learn h^*, which reduces clean accuracy to be $(1 - \text{Bayes error})$, and robust accuracy of the Bayes classifier to be $(1 - RR^*)$.

One immediate result is that Bayes error remains the same under any distribution shift induced by an injective map $T : \mathcal{X} \to \mathcal{X}$. To see that, simply note that T^{-1} exists and $h^* \circ T^{-1}$ gives the same Bayes error for the shifted distribution. However, such invariance property does not hold for the robust error of the Bayes classifier. Furthermore, the following two examples show that Bayes error can have completely different behavior from its robust error. Although both examples have 0 Bayes error, they have completely different robust errors.

Example 1. *Assume* x *is uniformly distributed in* $[0, 1]^d$ *and* $\mathrm{y} = 1$*, for all* x *with* $\mathrm{x}^\top e_1 > 1/2$ *and* $\mathrm{y} = 0$*, for* $\mathrm{x}^\top e_1 \leq 1/2$*, where* e_1 *is the one-hot vector. We use the 0-1 loss here. Note that the Bayes error decision boundaries are given by the following hyperplane:* $HP_1 = \{x \in [0, 1]^d : x_1 = 0\}$*, and thus*

$$R^* = 0; \qquad RR^* = 2\epsilon,$$

under the budget $\|\delta\|_\infty < \epsilon$*. In this case, the robust error is tolerable and relatively robust measured by the fraction of points that are successfully attacked,* 2ϵ*.*

Moreover, consider an injective map T *which maps* $\{x : x^\top e_1 > 1/2\}$ *to* $\{x : x^\top \mathbf{1} > \frac{d}{2}\}$*, and* $\{x : x^\top e_1 \leq 1/2\}$ *to* $\{x : x^\top \mathbf{1} \leq \frac{d}{2}\}$[1]*. The Bayes error on the new distribution remains* 0*, as* T *is invertible. In contrast, the robust error is much worse. As shown in Proposition 2.1,*

$$RR^* \geq 1 - \frac{1}{4d\epsilon^2}.$$

Remark 2.1. *Note that here the robust error of the Bayes classifier will grow to 1 as the dimensionality increases, for a fixed budget* ϵ*.*

[1] Such map can be easily constructed.

Proposition 2.1 (Existence of Non-Adversarially Robust Decision Boundary)**.** *Let* x *be uniformly distributed on* $[0,1]^d$ *and* $\mathrm{y} = 1$*, for all* x *such that* $\mathrm{x}^\top \mathbf{1} > \frac{d}{2}$ *and* $\mathrm{y} = 0$ *otherwise. Consider adversarial attack under budget* $\|\delta\|_\infty < \epsilon$*. Then for zero-one loss* L*:*

$$RR^* = \mathbb{E}_{\mathbb{P}(X,Y)} \max_{\|\delta\|_\infty < \epsilon} L(Y; h^*(x+\delta)) \geq 1 - \frac{1}{4d\epsilon^2}$$

2.2 Difficulty in achieving robustness

Example 1 shows that good clean accuracy does not necessary lead to good robust accuracy. In contrast, we will show in this section that achieving a good robust accuracy is impossible given uniformly distributed data, as long as we ask for good clean accuracies. Our tool are classical results from the concentration of measure (Ledoux, 2005).

Let $A_\epsilon := \{x \in \mathbb{R}^N | \mathrm{d}(x, A) < \epsilon\}$ denote the ϵ-neighborhood of the nonempty set A. Theorem 2.1 provides a lower bound on the mass in A_ϵ.

Theorem 2.1 (Concentration of Measure on the Unit Cube and the Unit Ball)**.** *Let* $[0,1]^d$ *denote the unit* d*-cube and* B^d *denote the Euclidean unit* d*-ball, both equipped with uniform probability distributions. Let* $\epsilon > 0$*. Then for any* $A \subset [0,1]^d$ *with* $\mathbb{P}(A) \geq 1/2$*, we have:*

$$\mathbb{P}(A_\epsilon) \geq \Phi(\epsilon\sqrt{2\pi} + \Phi^{-1}(\mathbb{P}(A))) \geq 1 - e^{-\pi\epsilon^2} \tag{1}$$

For any $B \subset B^d$*, with* $\mathbb{P}(B) \geq 1/2$*,*

$$\mathbb{P}(B_\epsilon) \geq 1 - \frac{1}{\mathbb{P}(B)}(1 - \delta_{\ell_2}(\epsilon))^{2d} \geq 1 - \frac{1}{\mathbb{P}(B)} e^{-2d(\frac{2-\sqrt{3}}{3})\epsilon^2} \tag{2}$$

where $\delta_{\ell_2}(\epsilon) = 1 - \sqrt{1 - \frac{\epsilon^2}{4}}$ *and* Φ *is the standard normal cumulative distribution function.*

Based on Theorem 2.1 we can now show that under some circumstances, *no* algorithm that achieves can perfect clean accuracy can also achieve a good robust accuracy.

Example 2 (Vulnerability Guarantee)**.** *Consider the joint distribution* $\mathbb{P}(\mathrm{x}, \mathrm{y})$*, where the input data* x *is uniformly distributed on* $[0,1]^d$ *and label* y *has 10 classes. Further assume the marginal distribution of* y *is also uniform*[2]*. Theorem 2.1 implies that under* ℓ_2 *adversarial attack with* $\epsilon = 0.5$*, at least 94 % of the samples are ether wrongly classified or can be successfully attacked for a classifier with perfect clean accuracy.*

Furthermore, if $d = 3 * 32 * 32$*, A parallel calculation for* $\mathbb{P}(\mathrm{x}, \mathrm{y})$ *on the* B^d *domain gives: under* ℓ_2 *adversarial attack with* $\epsilon = 0.09$ *, at least 97 % of the the samples are ether wrongly classified or can be successfully attacked for a classifier with perfect clean accuracy.*

On the one hand, Theorem 2.1 and Example 2 suggest that uniform distribution on a $[0,1]^d$ enjoys more robustness and is not affected by the high dimensionality. This may partially explain why MNIST is more adversarially robust than CIFAR10, as the distribution of x in CIFAR10 is "closer" to B^d than to $[0,1]^d$. On the other hand, while not completely sharp, they also suggest the intrinsic difficulty in achieving good robust accuracy.

Note that one limit of Theorem 2.1 and Example 2 is the uniform distribution assumption, which is surely not true for natural images. Indeed, although rigorously developed, Theorem 2.1 and Example 2 do *not* explain certain empirical observations. Following Kolter and Wong (2017), we train a provably[3] robust model on a binarized MNIST dataset (bMNIST) [4]. Our experiments shows that the learned model is (rigorously) guaranteed to achieve less than 3.00% robust error on bMNIST test data, while maintains 97.65% clean accuracy. Details of this experiments in described in Appendix B.2.

Such experiment clearly suggests the essential role of the data distribution in achieving good robust and clean accuracy. While it is hard to completely answer the question what geometric properties differentiate the concentration rates between the ball/the cube in high dimension, and the distribution of bMNIST. We remark that one obvious difference is the distance distributions in both spaces.

[2] but their joint distribution is not necessary uniform.

[3] "Provably" means that the robust accuracy of the model can be rigorously proved.

[4] It is created by rounding all pixel values to 0 or 1 from the original MNIST

Could the distance distributions explain the differences in achievable clean and robust accuracy? Note that the same method can only achieve 37.70% robust error on original MNIST data, and even higher error on CIFAR10, which further supports this hypothesis. In the rest of this paper, we further investigate the dependence of robust accuracy on the distribution of real data.

3 Robustness on Datasets Variants with Different Input Distributions

Section 2.2 clearly suggests that the data distribution plays an essential role in the achievable robust accuracy. 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 $\mathbb{P}(\mathrm{x})$'s while keep $\mathbb{P}(\mathrm{y}|\mathrm{x})$ 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.

3.1 Smoothing and Saturation

We now explain how the new datasets are generated under "semantic-lossless" shifts. In general, MNIST has a more binary distribution of pixels, while CIFAR10 has a more continuous spectrum of pixel values, as shown in Figure 1a and Figure 1b. To bridge the gap between this two datasets that have completely different robust accuracies, we propose two operations to modify their distribution on x: smoothing and saturation, as described below. We apply different levels of "smoothing" on MNIST to create more CIFAR-like datasets, and different levels of "saturation" on CIFAR to create more "binary" ones. Note that we would like to maintain the semantic information of the original data, which means that such operations should be semantics-lossless and not arbitrarily wide.

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. Our MNIST variants include the binarized MNIST and smoothed MNIST with different kernel sizes. As shown in Figure 1c, all MNIST variants still maintain the semantic information in MNIST, which indicates that $\mathbb{P}(\mathrm{y} \mid \tilde{x}^{(s)})$ should be similar to $\mathbb{P}(\mathrm{y} \mid x)$. It is thus reasonable to assume that y_i is approximately sampled from $\mathbb{P}(\mathrm{y} \mid \tilde{x}^{(s)})$, and as such we assign y_i as the label of $\tilde{x}^{(s)}$. Note that all the data points in the binarized MNIST are on the corners of the unit cube. For the smoothed versions, pixels on the digit boundaries are pushed off the corner of the unit cube.

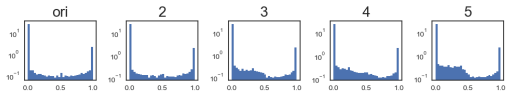
Saturation of the image x_i is denoted by $\widehat{x_i}^{(p)}$, and the procedure is defined as below: for each pixel of x, v, its corresponding pixel of $\widehat{x}^{(p)}$ is defined as

$$\mathrm{sign}(2v-1)\frac{|2v-1|^{\frac{2}{p}}}{2}+\frac{1}{2}\in[0,1].$$

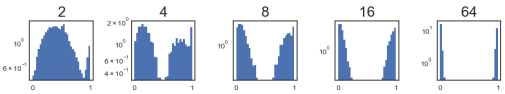
Saturation is used to generate variants of the CIFAR10 dataset with less centered pixel values. For different saturation level p's, one can see from Figure 1d that $\widehat{x}^{(p)}$ is still semantically similar to x in the same classification task. Similarly we assign y_i as the label of $\widehat{x}^{(p)}$. One immediate property about $\widehat{x}^{(p)}$ is that it pushes x to the corners of the data domain where the pixel values are either 0 or 1 when $p \geq 2$, and pull the data to the center of 0.5 when $p \leq 2$. When $p = 2$ it does not change the image, and when $p = \infty$ it becomes binarization.

3.2 Experimental Setups

In this section we use the smoothing and saturation operations 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. Since we are only concerned with the intrinsic robustness of neural networks models, we do not consider methods like preprocessing that tries to remove perturbations or randomizing inputs. We perform standard neural network training on clean data to



(a) Pixel value histogram (log scale in y) of MNIST variants, from left to right: original, smoothed with kernel size 2, 3, 4, 5



(b) Pixel value histogram (log scale in y) of CIFAR10 variants, from left to right: original, saturation level 4, 8, 16, 64

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

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

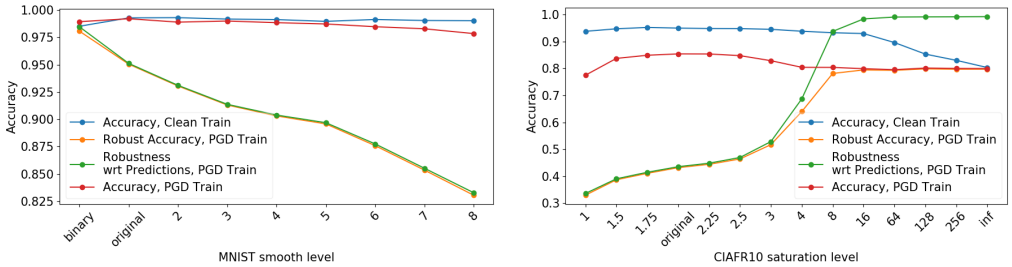
Figure 1: Variants of smoothed MNIST and saturated CIFAR10 datasets.

measure the difficulty of the classification task, and projected gradient descent (PGD) based adversarial training (Madry et al., 2017) to measure the difficulty to achieve robustness.

By default, we use LeNet5 on all the MNIST variants, and use wide residual networks (Zagoruyko and Komodakis, 2016) with widen factor 4 for all the CIFAR10 variants. Unless otherwise specified, PGD training on MNIST variants and CIFAR10 variants all follows the settings in Madry et al. (2017). Details of network structures and training hyperparameters can be found in Appendix B.

We evaluate the classification performance using the test *accuracy* of standardly trained models on clean unperturbed examples, and the robustness using the *robust accuracy* of PGD trained model, which is the accuracy on adversarially perturbed examples. Although not directly indicating robustness, we report the clean accuracy on PGD trained models to indicate the tradeoff between being accurate and robust. To understand whether low robust accuracy is due to low clean accuracy or vulnerability of model, we also report *robustness wrt predictions*, where the attack is used to perturb against the model's clean prediction, instead of the true label. We use ℓ_∞ untargeted PGD attacks (Madry et al., 2017) as our adversary, since it is the strongest attack in general based on our experiments. Unless otherwise specified, PGD attack on MNIST variants runs 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 Madry et al. (2017).

3.3 Sensitivity of Robust Accuracy to Data Transformations



(a) MNIST results under different smooth levels

(b) CIFAR10 results under different saturation levels

Figure 2: Accuracy, Robust Accuracy and Robustness wrt Predictions on different data variants

Results on MNIST variants are presented in Figure 2a [5]. According to the results, 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 wrt 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 face a conflict against achieving high clean accuracy. This result conforms with results of provably robust model having high robustness on binarized MNIST described in Section 2.

[5] Exact numbers are listed in Table 2 in Appendix C.

Different results on CIFAR10 variants are reported in Figure 2b. [6], which tells a similar story. 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 accuracy 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" of the images. 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 wrt 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 does not introduce 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.

4 Practical Implications

Given adversarial robustness' sensitivity to input distribution, we further demonstrate two practical implications: 1) Robust accuracy could be sensitive to image acquisition condition and preprocessing. This leads to unreliable benchmarks in practice; 2) When introducing new dataset for benchmarking adversarial robustness, we need to carefully choose datasets with the right characteristics.

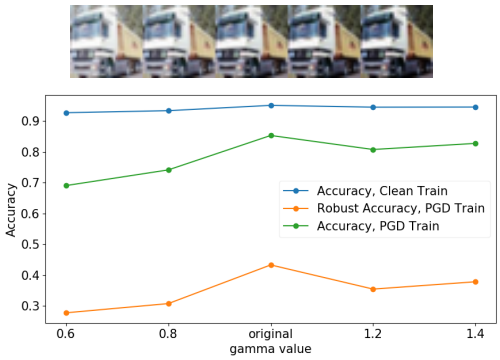
4.1 Robust Accuracy is Sensitive to Gamma Correction

The natural images are acquired under different lighting conditions, with different cameras and different camera settings. They are also preprocessed in different ways. All these factors could lead to mild shift on input distribution. Therefore, we might get very different performance measures when performing adversarial training on images taken under different conditions. In this section, we use gamma mapping to transform CIFAR10 images to different variants. 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 Szeliski (2010) on more details about gamma mappings. Figure 3a 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.

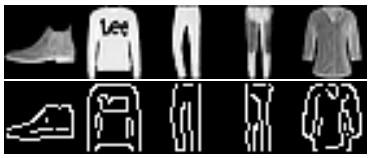
We perform the same experiments as in the saturated CIFAR10 variants experiment in Section 3. The results are displayed in Figure 3a. Accuracies on clean data almost remain the same across different gamma values. However, under PGD training, both accuracy and robust accuracy varies largely following 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". Note that this is not to use a trained model on slightly varied datasets. The discrepancies of robust performance measures is between models trained on slightly varied datasets separately. In other words, if an training algorithm achieves good robustness on one image datasets, it doesn't necessarily achieve similar robustness on another semantically-identical but slightly varied datasets. Therefore, the actual robustness could either be significantly underestimated or overestimated.

[6] Exact numbers are listed in Table 3 in Appendix C.



(a) Top: Gamma mapped images from left to right 0.6, 0.8, 1.0 (original image), 1.2 , 1.4; Bottom: Robustness results on gamma mapped CIFAR10 variant



fMNIST: accuracy, standard training, 92.7%
accuracy, PGD training, 81.2%
robust accuracy, PGD training, 65.3%
efMNIST: accuracy, standard training, 88.3%
accuracy, PGD training, 87.2%
robust accuracy, PGD training, 86.6%

(b) Top: Examples of fashion-MNIST images and edge-fashion-MNIST; bottom: Robustness results on fMNIST and efMNIST

Figure 3: Illustrations on Practical Implications

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. This is an important open question and we defer it to future research.

4.2 Choice of Datasets for Evaluating Robustness

As discussed, evaluating robustness on a suitable dataset is important. Here we use fashion-MNIST (fMNIST) (Xiao et al., 2017) and edge-fashion-MNIST (efMNIST) as examples to analyze characteristics of "harder" datasets. The edge-fashion MNIST is generated by running Canny edge detector (Canny, 1986) with $\sigma = 1$ on the fashion MNIST images. Figure 3b shows examples of fMNIST and efMNIST. We performed the same standard training and PGD training experiments on both fMNIST and efMNIST as we did on MNIST. Figure 3b shows the results. We can see that fMNIST exhibit similar behavior to CIFAR10, where the test accuracy is significantly affected by PGD training and the gap between robust accuracy and accuracy is large. On the other hand, efMNIST is closer to the binarized MNIST: the accuracy is affected very little by PGD training, along with an insignificant difference between robust accuracy and accuracy.

Both fMNIST and efMNIST can be seen as a "harder" MNIST, but they are harder in different ways. One one hand, since efMNIST results from the edge detection run on fMNIST, it contains less information. It is therefore harder to achieve higher accuracy on efMNIST than on fMNIST, where richer semantics is accessible. However, fMNIST's richer semantics makes it better resembles natural images' pixel value distribution, which could lead to increased difficulty in achieving adversarial robustness. efMNIST, on the other hand, can be viewed as a set of "more complex binary symbols" compared to MNIST or binarized MNIST. It is harder to classify these more complex symbols. However, it is easy to achieve high robustness due to the binary pixel value distribution.

To sum up, when introducing new dataset for adversarial robustness, we should not only look for a "harder" one, but we also need to consider whether the dataset is "harder in the right way".

5 Attempts to Understand the Phenomenon

In this section, we make initial attempts to understand the sensitivity of adversarial robustness. We use saturated CIFAR10 as the running example. These analyses also apply to smoothed MNIST. Saturation pushes pixel values towards 0 or 1, thus pushing data points towards the corner of unit cube. This immediately leads to: 1) the "perturbable volume" decreases; 2) distances between data examples increases. Intuitively, both could be related to the increasd robustness. We analyze them and show that although they are correlated with robustness change, none of them can fully explain the observed phenomena. We then further examine the possibility of increasing robust accuracy on less robust datasets by having larger models and more data.

5.1 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 dimen-

Table 1: 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 %

sionality, the volume of "perturbable region" changes drastically across different saturation levels. For example, the average log perturbable volume [7] 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 [8], then 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, which serves as evidence that differences in perturbable volume are not causing the differences in robustness.

5.2 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. Since the margin is difficult to measure, we use the "inter-class distance" as an approximate measure. Inter-class distance [9] 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 D.2.1 Figure 5) 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 1 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.

5.3 On the Required Model Capacity and Sample Complexity

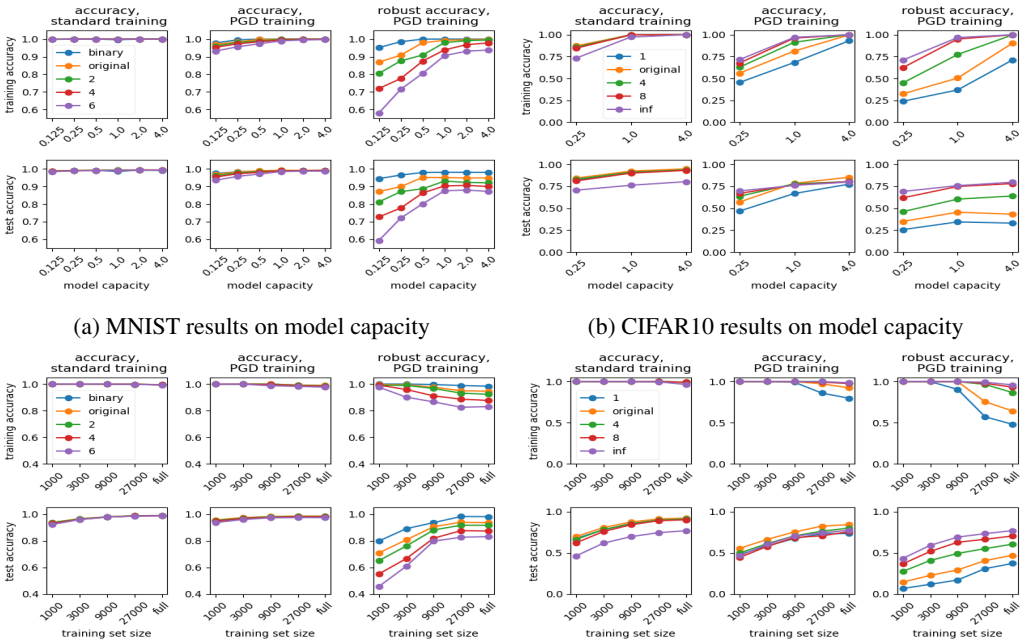
In practice, it is unclear how far robust accuracy of PGD trained model is from adversarial Bayes error RR^* for the given data distribution. In the case RR^* is not yet achieved, this is a non-exhaustive list that we can improve upon: 1) better training/learning algorithm; 2) increase model capacity; 3) train on more data. Finding a better learning algorithm is beyond the scope of this paper. Here we inspect 2) and 3) to see if it is possible to improve robustness by having larger model and more data. For model capacity, we use differently sized LeNet5 by multiplying the number of channels at each layer with different widen factors. These factors include 0.125, 0.25, 0.5, 1, 2, 4. On CIFAR10 variants, we use WideResNet with widen factors 0.25, 1 and 4. For sample complexity, we follow the practice in Section 3 except that we use a weight decay value of 0.002 to prevent overfitting. For both MNIST and CIFAR10, we test on 1000, 3000, 9000, 27000 and entire training set. Both model capacity and sample complexity results are shown in Figure 4.

For MNIST, both training and test accuracies of clean training are invariant to model sizes, even we only use a model with widen factor 0.125. In slight contrast, both the training and test accuracy

[7] Definition of "log perturbable volume" and other detailed analysis of perturbable volume are given in Appendix D.1 and Table 7.

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

[9] The calculation of "inter-class distance" and other detailed analyses are delayed to Appendix D.2.1 and Figure 5.



(a) MNIST results on model capacity

(b) CIFAR10 results on model capacity

(c) MNIST results on training set size

(d) CIFAR10 results on training set size

Figure 4: Model capacity and training set size's influences on accuracy and robust accuracy. In each subfigure, the top row contains accuracy and robust accuracy measured on training set, the bottom row contains results measured on test set.

of PGD training increase as the model capacity increases, but it plateaus after widen factor 1 at an almost 100% accuracy. For robust accuracy, training robust accuracy kept increasing as model gets larger until the value is close to 100%. However, test robust accuracy stops increasing after widen factor 1, additional model capacity leads to larger (robust) generalization gap. When we vary the size of training set, the model can always fit the training set well to almost 100% clean training accuracy under standard training. The clean test accuracy grows as the training set size get larger. Training set size has more significant impact on robust accuracies of PGD trained models. For most MNIST variants except for binarized MNIST, training robust accuracy gradually drops, and test robust accuracy gradually increases as the training set size increases. This shows that when training set size is small, PGD training overfits to the training set. As training set gets larger, the generalization gap becomes smaller. Both training and test robust accuracies plateau after training set size reaches 27000. Indicating that increasing the training set size might not help in this setting. In conclusion, for MNIST variants, increasing training set size and model capacity does not seem to help beyond a certain point. Therefore, it is not obvious on how to improve robustness on MNIST variants with higher smoothing levels.

CIFAR10 variants exhibit similar trends in general. One notable difference is that for PGD training, the training robust accuracy does not plateau as model size increases. However the test robust accuracy plateaus after wide factor 1. Also when training set size increases, the training robust accuracy drops and test robust accuracy increases with no plateau present. These together suggest that having more training data and training a larger model could potentially improve the robust accuracies on CIFAR10 variants. One interesting phenomenon is that binarized MNIST and ∞-saturated CIFAR10 has different sample complexity property, despite both being "cornered" datasets. This indicates that the existence of complex interaction between the classification tasks and input data distribution.

6 Conclusion

Our theoretical analyses shows the significance of input data distribution in adversarial robustness, which further motivates our systematic experiments on MNIST and CIFAR10 variants. We discover that, counter-intuitively, robustness of adversarial trained models are sensitive to semantically-preserving transformations on data. We then discuss the difficulties on evaluating robust learning algorithms. Finally, we make initial attempts to understand this sensitivity, and believe deeper understanding requires non-trivial future efforts.

REFERENCES

Athalye, A., Carlini, N., and Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*.

Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. (2017). Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*.

Ball, K. (1997). An elementary introduction to modern convex geometry. *Flavors of geometry*, 31:1–58.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. (2013). Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698.

Carlini, N. and Wagner, D. (2016). Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*.

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

Erraqabi, A., Baratin, A., Bengio, Y., and Lacoste-Julien, S. (2018). A3t: Adversarially augmented adversarial training. *arXiv preprint arXiv:1801.04055*.

Fawzi, A., Fawzi, O., and Frossard, P. (2015). Analysis of classifiers' robustness to adversarial perturbations. *arXiv preprint arXiv:1502.02590*.

Gastaldi, X. (2017). Shake-shake regularization. *arXiv preprint arXiv:1705.07485*.

Gilmer, J., Metz, L., Faghri, F., Schoenholz, S. S., Raghu, M., Wattenberg, M., and Goodfellow, I. (2018). Adversarial spheres. *arXiv preprint arXiv:1801.02774*.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Huang, R., Xu, B., Schuurmans, D., and Szepesvári, C. (2015). Learning with a strong adversary. *arXiv preprint arXiv:1511.03034*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kolter, J. Z. and Wong, E. (2017). Provable defenses against adversarial examples via the convex outer adversarial polytope. *arXiv preprint arXiv:1711.00851*.

Kurakin, A., Goodfellow, I., and Bengio, S. (2016). Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*.

Ledoux, M. (2005). *The concentration of measure phenomenon*. Number 89. American Mathematical Soc.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582.

Raghunathan, A., Steinhardt, J., and Liang, P. (2018). Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Mądry, A. (2018). Adversarially robust generalization requires more data. *arXiv preprint arXiv:1804.11285*.

Sinha, A., Namkoong, H., and Duchi, J. (2017). Certifiable distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. (2018). There is no free lunch in adversarial robustness (but there are unexpected benefits). *arXiv preprint arXiv:1805.12152*.

Warde-Farley, D. and Goodfellow, I. (2016). 11 adversarial perturbations of deep neural networks. page 311.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xu, H., Caramanis, C., and Mannor, S. (2009). Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, 10(Jul):1485–1510.

Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Appendix

A PROOFS

A.1 PROOF FOR PROPOSITION 2.1

Proof. The argument is well-known in concentration of measure. We provide here for the sake of completeness and adapt it to the context. The hyperplane $HP_2 = \{x \in [0,1]^d : X^\top \mathbf{1} = \frac{d}{2}\}$ defines the decision boundary. We first compute the orthogonal distance of a given point $y = (y_1, y_2, \cdots, y_d) = (x_1 + \delta_1, x_2 + \delta_2, \cdots, x_d + \delta_d)$ to HP_2. The point y is the perturbed point within budget $\|\delta\|_\infty < \epsilon$. The vector $\mathbf{1}$ is orthogonal to HP_2. Pick any point $x \in HP_2$, the orthogonal distance from y to HP_2 is:

$$\begin{aligned}
\ell_2(y, HP_2) = \|Proj_{\mathbf{1}}(y - x)\| &= \|\mathbf{1}\frac{(y-x)^\top \mathbf{1}}{\mathbf{1}^\top \mathbf{1}}\| \\
&= |\frac{y^\top \mathbf{1} - x^\top \mathbf{1}}{\|\mathbf{1}\|}| \\
&= |\frac{\delta^\top \mathbf{1}}{\|\mathbf{1}\|}| \\
&= |\frac{y^\top \mathbf{1} - \frac{d}{2}}{\sqrt{d}}|
\end{aligned}$$

The last two equations show that the ℓ_2 distance under an ℓ_∞ attack can grow at the rate of $\epsilon\sqrt{d}$, for this particular hyperplane HP_2.

Now we take the expectation over $[0,1]^d$, and note that expectation of the uniform distribution over a product space $[0,1]^d$ is the same as taking expectation on each dimension (Fubini's theorem), picking each random variable coordinatewise uniformly from $[0,1]$.

$$\begin{aligned}
\mathbb{E}[\ell_2^2(y, HP_2)] = \mathbb{E}[(\frac{y^\top \mathbf{1} - \frac{d}{2}}{d})^2] &= \frac{1}{d}\mathbb{E}[(\sum_{i=1}^d y_i - \frac{d}{2})^2] \\
&= \frac{1}{d}\mathbb{V}[\sum_{i=1}^d y_i] = \frac{1}{d}\sum_{i=1}^d \mathbb{V}[y_i] = \frac{1}{4}
\end{aligned}$$

Then we apply Markov's inequality, for all real number $t > 0$:

$$\mathbb{P}(\ell_2(x, H) \geq \sqrt{t}) = \mathbb{P}(\ell_2(x, H)^2 \geq t) \leq \frac{1}{4t}$$

Finally, we observe that the longest (in terms of ℓ_2 norm) such ϵ ℓ_∞ attacks vector to HP_2 are parallel to the normal vector $\mathbf{1}$ to HP_2. They have ℓ_2 distance $\epsilon\sqrt{d}$. The set these attacks cover is characterized by $\{x \in [0,1]^d : \ell_\infty(x, H) \leq \epsilon\}$ = $\{x \in [0,1]^d : \ell_2(x, H) \leq \epsilon\sqrt{d}\}$.

Let $t = \epsilon^2 d$, we have:

$$\mathbb{P}(\ell_2(x, H) \geq \sqrt{t}) = \mathbb{P}(\ell_2(x, H)^2 \geq t) \leq \frac{1}{4t} = \frac{1}{4\epsilon^2 d}$$

In the case of zero-one loss, $RR^* = \mathbb{P}(\ell_2(x, H) \leq \epsilon\sqrt{d}) \geq 1 - \frac{1}{4\epsilon^2 d}$. □

A.2 PROOF FOR THEOREM 2.1

Proof. **(First Inequality for Cube)** The proof here follows that of Ledoux (2005), but we track of the tight constants so as to give tighter adversarial robustness calculations.

Let Φ be one dimensional standard normal cumulative distribution function and let μ_d denote d dimensional Gaussian measures. Consider the map $T : \mathbb{R}^d \longrightarrow (0,1)^d$:

$$T(x_1, \cdots, x_d) = (\Phi(x_1), \cdots, \Phi(x_d))$$

T pushes forward μ_d defined on $\mathbb{R}^d$ into a probability measure $\mathbb{P}$ on $(0,1)^d$:

$$\mathbb{P}(A) = \mu_d(T^{-1}(A))$$

for $A \subset (0,1)^d$. Next we have the following Gaussian isoperimetric/concentration inequality (Ledoux, 2005):

$$\mu_d(B_\epsilon) \geq \Phi(\Phi^{-1}(\mu_d(B)) + \epsilon)$$

for all $B \subset \mathbb{R}^d$ measureable.

Now for $A \subset (0,1)^d$, we have:

$$\mathbb{P}(A_\epsilon) = \mu_d(T^{-1}(A_\epsilon)) \geq \mu_d(T^{-1}(A)_{\epsilon\sqrt{2\pi}}) \geq \Phi(\Phi^{-1}(\mu_d(T^{-1}(A)) + \sqrt{2\pi}\epsilon))$$

where the first inequality follows from that T has Lipschitz constant $\frac{1}{\sqrt{2\pi}}$, and thus T^{-1} has Lipschitz constant $\sqrt{2\pi}$; and the second one follows from Gaussian isoperimetric inequality.

When $\mathbb{P}(A) \geq 1/2$,

$$\Phi(\Phi^{-1}(\mu_d(T^{-1}(A)) + \sqrt{2\pi}\epsilon)) \geq \Phi(\Phi^{-1}(\sqrt{2\pi}\epsilon))$$

Additionally, the inequality $\Phi(x) \geq 1 - e^{\frac{x^2}{2}}$ implies the last inequality in the theorem.

(Second Inequality for Ball)

We first define the notion of modulus of convexity for a normed space, in this case ℓ_2:

$$\begin{aligned}\delta_{\ell_2}(\epsilon) &= \inf\{1 - \|\frac{x+y}{2}\| : \|x\| = \|y\| = 1, \|x-y\| \geq \epsilon\} \\ &= 1 - \sqrt{1 - \frac{\epsilon^2}{4}}\end{aligned}$$

The important property about $\delta_{\ell_2}(\epsilon)$ is that there is a constant C such that:

$$\delta_{\ell_2}(\epsilon) \geq C\epsilon^2$$

By elementary algebraic calculuation, We can take $C = \frac{2-\sqrt{3}}{3}$.

By Equation (2.25) in (Ledoux, 2005),

$$\mathbb{P}(A_\epsilon) \geq 1 - \frac{1}{\mathbb{P}(B)}(1 - \delta_{\ell_2}(\epsilon))^{2d} \geq 1 - \frac{1}{\mathbb{P}(A)}e^{-2d\delta_{\ell_2}(\epsilon)} = 1 - \frac{1}{\mathbb{P}(A)}e^{-2d(\frac{2-\sqrt{3}}{3})\epsilon^2}$$

□

B Detailed Settings for Training

B.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 Zagoruyko and Komodakis (2016) for our experiments, where 28 is the depth and 4 is the widen factor. We use "per image standardization" [10] to preprocess CIFAR10 images, following Madry et al. (2017).

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

Table 2: 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 wrt 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 3: Performance and Robustness of models trained on CIFAR10 variants.

	Standard Training	PGD Training		
MNIST Variants	Test Acc	Test Acc	Robust Accuracy $\epsilon = 0.3$	Robustness wrt Predictions $\epsilon = 0.3$
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 %

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.

B.2 LP robust model described in Section 2

For the linear programming based provably robust model (Kolter and Wong, 2017) (LP-robust model). We trained a ConvNet identical to the one in the original paper. It has 2 convolutional layers, with 16 and 32 channels, each with a stride of 2; and 2 fully connected layers, the first one maps the flattened convolution features to hidden dimension 100, the second maps to 10 logit units. We use ReLUs as the nonlinear activation and there is no max pooling in the network.

We train for 100 epochs with batch size 50. The first 50 epochs are warm start epochs where epsilon increases from 0.01 to 0.3 linearly. We use Adam optimizer (Kingma and Ba, 2014) with a constant learning rate of 0.001.

C Detailed Experimental Results

We listed exact numbers of experiments involved in the main body in Table 2, 3, 4 and 5.

Table 4: 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%

D Detailed Analyses

D.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 6, 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

Table 5: 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 6: 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

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 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 7. "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

Table 7: 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 %

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 (Goodfellow et al., 2014; Warde-Farley and Goodfellow, 2016) that adversarial examples appears in certain directions instead of being distributed in small pockets across space. Therefore, the perturbable volume hypothesis is rejected.

D.2 DETAILED ANALYSES OF INTER-CLASS DISTANCE

D.2.1 CALCULATION OF INTER-CLASS DISTANCE

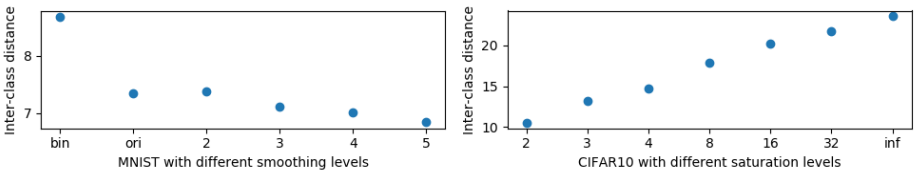
We calculate inter-class distance as follows. Let $D = \{x_i\}$ denote the set of all the input data points, $D_c = \{x_i | y_i = c\}$ denote the set of all the data points in class c, and $D_{\neg c} = \{x_i | y_i \neq c\}$ denote all the data points not in class c. Our goal is to calculate $d(D_c, D_{\neg c})$ 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 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.

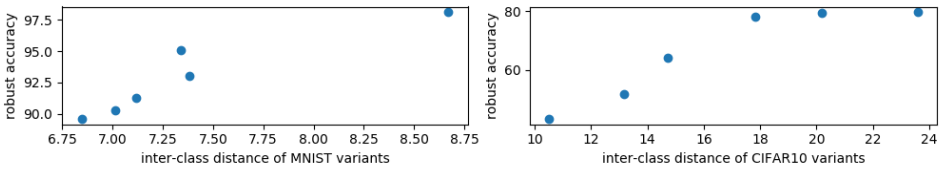
Figure 5 shows the inter-class distances (averaged over all classes) calculated on MNIST and CIFAR10 variants. The binarized MNIST has a significantly larger inter-class distance. As smoothing kernel size increases, the distance also decrease slightly. On CIFAR10 variants, as the saturation level gets 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 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 1, suggests that inter-class distance cannot fully explain the robust variation across different dataset variants.

D.2.2 INTER-CLASS DISTANCE COULD POTENTIALLY INFLUENCE REQUIRED MODEL CAPACITY

We attempt to understand the relation between the inter-class distance of a dataset and its achievable robustness in this section. We first illustrate our intuition in a synthetic experiment, where a ReLU network is trained to perfectly separate 2 concentric spheres (Gilmer et al., 2018), as shown in Figure 6. Here the inter-class distance is the width of the ring between two spheres. In such example, adversarial training is actually closely related to the inter-class distance of the data. In fact, in the simple setting where the classifier is linear, it has been shown in Xu et al. (2009) that adversarial training, as a particular form of robust optimization, is equivalent to maximizing the classification margins. Following this intuition, one can easily see that the effect of adversarial training is to push two spheres close to each other, and requires the network to perfectly separate the new spheres with much smaller inter-class.



(a) Processing levels vs inter-class distances



(b) Inter-class distance vs robust accuracy

Figure 5: Inter-class distance's influence on robust accuracy on different MNIST and CIFAR10 variants

Intuitively, when the inter-class distance is large, i.e. the gap between two spheres are large, reasonable model should be able to achieve good standard accuracy. We have also observed such phenomenon on original MNIST and saturated CIFAR (say level 16). As the inter-class gets smaller, although the model capacity could still be enough for the standard training, it may no longer be enough for adversarial training, upon which we would observe that although the test accuracies stay similar, accuracies under adversarial attack significantly would drop. We have also seen similar behavior on smooth MNIST data and smaller level of saturated CIFAR data. Finally, when the inter-class distance is so small such that even a high clean test accuracy may be difficult to achieve.

Considering robust accuracy as the clean accuracy with smaller gap between the spheres, the next theorem provides a theoretical guarantee in relating together the difficulty of attaining good accuracy under attack and the model capacity (Ball, 1997), verifying our intuition above. Note that one way to measure the capacity of a ReLU network is by counting the number of its induced piece-wise linear region, which is closely related to the number of facets of its decision boundary.

Theorem D.1. *Let $d(K, L)$ between symmetric convex bodies K and L denote the least positive d for which there is a linear image $\tilde{L}$ of L such that $\tilde{L} \subset K \subset d\tilde{L}$. Let K be a (symmetric) polytope in $\mathbb{R}^n$ with $d(K, B_2^n) = d$. Then K has at least $e^{n/(2d^2)}$ facets. On the other hand, for each n, there is a polytope with $4n$ facets whose distance from the ball is at most 2.*

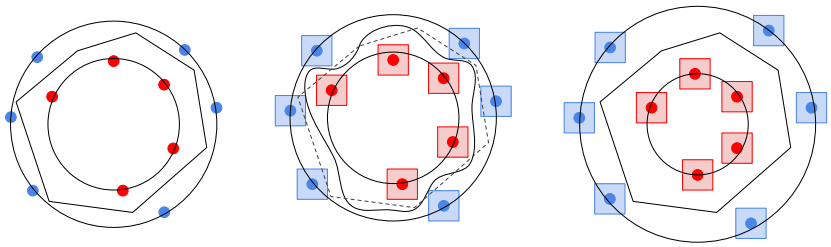


Figure 6: Inter-class distance and required model capacity. Left: when distance is small, a small capacity polytope classifier could separate original data; middle: when distance is small, linear classifier not able to separate data points "robustly", but a more complex nonlinear classifier could; right:when distance is large, a a small capacity polytope classifier can separate data "robustly".

The above analysis is partially supported by our experiments on model capacity in Section 5.3. However, as we've shown in Section 5.2, the nature of the problem is complex and more conclusive statements requires further research.