Towards Deep Learning Models Resistant to Large Perturbations

Amirreza Shaeiri [1] Rozhin Nobahari [1] Mohammad Hossein Rohban [1]

Abstract

Adversarial robustness has proven to be a required property of machine learning algorithms. A key and often overlooked aspect of this problem is to try to make the adversarial noise magnitude as large as possible to enhance the benefits of the model robustness. We show that the well-established algorithm called "adversarial training" fails to train a deep neural network given a large, but reasonable, perturbation magnitude. In this paper, we propose a simple yet effective initialization of the network weights that makes learning on higher levels of noise possible. We next evaluate this idea rigorously on MNIST (ϵ up to ≈ 0.40) and CIFAR10 (ϵ up to $\approx 32/255$) datasets assuming the ℓ_∞ attack model. Additionally, in order to establish the limits of ϵ in which the learning is feasible, we study the optimal robust classifier assuming full access to the joint data and label distribution. Then, we provide some theoretical results on the adversarial accuracy for a simple multi-dimensional Bernoulli distribution, which yields some insights on the range of feasible perturbations for the MNIST dataset.

1. Introduction

Modern machine learning models achieve high accuracy on different tasks such as image classification (He et al., 2015a) and speech recognition (Graves et al., 2013; Xiong et al., 2016). However, they are not robust to adversarially chosen small perturbations of their inputs. In particular, one can make imperceptible perturbations of the input data, which can cause state-of-the-art models to misclassify their inputs with high confidence (Biggio et al., 2017; Carlini & Wagner, 2018; Carlini et al., 2016; Szegedy et al., 2014). Also, researchers have shown that even in the physical world scenarios, machine learning models are vulnerable to adversarial examples (Kurakin et al., 2016b; Li et al., 2019). The robustness properties of the models in machine learning are a huge concern, as they have been increasingly employed in applications in which safety and security are among the principal issues. Furthermore, the benefits of a robust model go beyond this; for instance, significantly improved model interpretability (Tsipras et al., 2018) and effective exclusion of brittle features in the learned robust model (Ilyas et al., 2019) are some of the model robustness benefits. The former could be observed through an input-dependent saliency map, which is usually a variant of the model gradient with respect to the input, and the fact that this map becomes sparser and semantically more relevant compared to that of a non-robust model. An example of the latter is the ability of the robust model to avoid "shortcut features" that are logically irrelevant but may be strongly correlated to the class. Learning under an adversarial noise would help in obscuring such features and make the model rely on alternative aspects of the input that are more robust for label prediction.

The phenomenon of adversarial machine learning has received significant attention in recent years and several methods have been proposed for training a robust classifier on images. However, most of these methods have been shown to be ineffective (Athalye & Carlini, 2018; Athalye et al., 2018; Carlini, 2019; Carlini & Wagner, 2016; 2017), while many others are shown to lack scalability to large networks that are expressive enough to solve problems like ImageNet (Cohen et al., 2019). To the best of our knowledge, only two training algorithms and their variants, which are called "adversarial training" (Madry et al., 2017) and "randomized smoothing" (Cohen et al., 2019), have been confirmed to be both effective and scalable. Nevertheless, adversarial training remains ineffective for large perturbations of the input (Sharma & Chen, 2018). More specifically, the authors observed that the adversarial training does not yield an accuracy better than random guessing for perturbation ℓ_∞ norm, denoted as epsilon, larger than or equal to 0.4 in the MNIST dataset. Training in the presence of large perturbations, however, is essential to advance the level of model robustness and its benefits that are gained as a result. For instance, if the model is robust to perturbations of ℓ_0 norm less than m, then only "shortcut features" that are captured in m pixels could be avoided. Otherwise, we would need a larger m to take the best advantage of the model robustness.

[1] Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. Correspondence to: Amirreza Shaeiri <shairi@ce.sharif.edu>, Rozhin Nobahari <nobahari@ce.sharif.edu>, Mohammad Hossein Rohban <rohban@sharif.edu>.

Preprint.

In addition to the mentioned robust training algorithms, some prior work was also aimed at studying adversarial machine learning from a more theoretical perspective (Attias et al., 2018; Chen et al., 2020; Cullina et al., 2018; Diochnos et al., 2019; Gilmer et al., 2018; Ilyas et al., 2019; Khim & Loh, 2018; Montasser et al., 2019; Schmidt et al., 2018; Wang et al., 2017; Yin et al., 2018; Zhai et al., 2019; Zhang et al., 2019). However, to our understanding, none of them claim to find the optimal robust classifier, assuming knowledge of joint data and label distribution, except only under a strong assumption on the hypothesis space (Ilyas et al., 2019). This helps to find a limit on the perturbation size until we could expect to get a better-than-random classifier assuming sufficiently large training set.

In section 3, we demonstrate that weight initialization of a deep neural network plays an important role in the feasibility of adversarial training of the network under a large perturbation size. A natural question that arises is:

How much can we increase the perturbation ℓ_p-norm during the training?

To answer this question, we study the optimal robust classifier, where we have the full knowledge of the input distribution given different classes. In section 5, we first prove that, in general, finding the optimal robust classifier in this setting is $\mathcal{NP}$-hard. Therefore, we focus our attention on some conditional distributions, such as the symmetric isotropic Normal and multi-dimensional Bernoulli distributions, in which finding the optimal robust classifier is tractable. Next, we discuss the limits on the perturbation size under which the optimal robust classifier has a better-than-chance adversarial accuracy.

In addition, in section 4, following (Tsipras et al., 2018) and (Kaur et al., 2019), we show that our models that are trained on larger perturbation sizes have more interpretable saliency maps and attacks, and some other notable visual properties. These results also suggest that using our proposed method to train the model adversarially on a larger perturbation size boosts the benefits that are gained in the robust models.

2. Related Work

Adversarial machine learning has been studied since two decades ago (Dalvi et al., 2004; Globerson & Roweis, 2006; Kolcz & Teo, 2009). But (Szegedy et al., 2013) and (Biggio et al., 2017) could be considered as the starting point of significant attention to this field, especially in the context of deep learning. Since then, many ideas have emerged that are intended to make the classification robust against adversarial perturbations. However, most of them have later been shown to be ineffective. Gradient masking (Papernot et al., 2016) is an example of issues that arise from such ideas, which leads to a false sense of security. In particular, obfuscated gradients (Athalye et al., 2018) could make it impossible for the gradient-based adversaries to attack the model. Subsequently, these defenses are easily broken by alternative adversaries, such as non-gradient based (Chen et al., 2017; Ilyas et al., 2018; Uesato et al., 2018), or black-box attacks (Papernot et al., 2016). This shows that evaluating the robustness of a neural network is a challenging task. In addition to these unsuccessful attempts, many others are not scalable to large networks that are expressive enough for the classification task on ImageNet and sometimes need to assume specific network architectures (Cohen et al., 2019).

"Adversarial training" is among the most established methods in the field, which was introduced in (Goodfellow et al., 2014; Kurakin et al., 2016a) and was completed later by (Madry et al., 2017) through the lens of robust optimization (Ben-Tal et al., 2009). Following this work, there have been numerous studies to improve the adversarial accuracy of adversarial training on the test set by using techniques such as domain adaptation (Song et al., 2018) or label smoothing (Shafahi et al., 2018), while some others have tried to decrease the computational cost of adversarial training (Shafahi et al., 2019; Wong et al., 2020). Researchers have also tried to apply adversarial training on more natural classes of perturbations such as translations and rotations for images (Engstrom et al., 2017), or on the mixture of perturbations (Maini et al., 2019; Tramer & Boneh, 2019). Also, (Ford et al., 2019) showed the relation between adversarial training and Gaussian data augmentation.

3. Proposed Method

Recall that in adversarial training (Madry et al., 2017), we need to solve the adversarial empirical risk minimization problem, which is a saddle point problem, that is formulated as:

$$\min_{\theta} \sum_{i=1}^{n} \max_{\delta \in S} \ell(f_\theta(x_i + \delta), y_i) \tag{1}$$

where S represents the set of feasible adversarial perturbations, and ℓ is the loss function. The inner maximization is referred to as the "adversarial loss". The setting that we study in this section is when

$$S = \{\delta \in \mathbb{R}^d : \|\delta\|_\infty \le \epsilon\}, \quad \epsilon > 0, \tag{2}$$

which is the most common perturbation set and used by (Madry et al., 2017) and is also considered as the standard benchmark in the context of adversarial examples.

To motivate the proposed method, we would begin with the result of an experiment. We observe that adversarial training on larger epsilons can not decrease the adversarial loss sufficiently, even on the training data. The training adversarial loss for different values of epsilons is provided both on MNIST (Fig. 1) and CIFAR10 (Supplementary Fig.

9 in Sec. D). The detailed experiments are provided in 3.1. This observation would raise the following question:

Is this an optimization issue in a highly non-convex non-concave min-max game; or is it just that learning is not feasible on large epsilons?

Before answering this question, one should note that optimization in deep learning has not been extensively studied from a theoretical perspective. Specifically, it is not even completely known why one could achieve near-zero training loss in standard training through randomly initialized gradient descent. This is especially surprising given the highly non-convex loss landscape. Yet the gradient descent is guaranteed to converge in this setting. However, we do not even have this convergence guarantee for adversarial training as a gradient descent-ascent algorithm on the highly non-convex non-concave min-max game (Schäfer & Anandkumar, 2019).

Nevertheless, in practice, researchers observed that the trainability of deep models is highly dependent on weight initialization, model architecture, the choice of optimizer, and a variety of other considerations (Li et al., 2017). The effect of different initializations on adversarial training has not been studied rigorously in the literature (Ben Daya et al., 2018; Simon-Gabriel et al., 2018).

Our main contribution is to propose a novel practical initialization for the adversarial training, which makes learning on larger perturbations feasible. Specifically, for adversarial training on large perturbations, we claim that the final weights of an adversarially trained network on a smaller epsilon can be used for this purpose. Surprisingly, this method can find a good solution for larger perturbations even with a few numbers of training epochs. Fig. 2 depicts the abovementioned idea by illustrating a significant decrease in the training loss for large epsilons using the proposed initialization, which was not possible using a random initialization. This is illustrated in various settings of initial and target epsilons for the MNIST and CIFAR10 datasets. The detailed experiments are provided in 3.1.

To gain more insights on the proposed method, we would address these questions:

1. Does adversarial training converge when we train on larger perturbations?

2. Why the low-cost local Nash equilibrium that is found by our method cannot be reached using random initialization?

3. Why does the proposed initialization converge quickly with few numbers of training epochs?

Towards addressing all these questions, in 3.2, we employ the "loss landscape model", which gives a geometric intuition about the loss function.

In 3.3, we first evaluate our method rigorously and then we discuss another surprising observation, which is the trainability of deep models on very large perturbations (e.g. $\epsilon \geq 0.5$ on MNIST, where the pixel intensities are scaled between 0 and 1), which obviously should not be possible, because the attacker can ideally transform all the pixel intensities to a single level and as a result, we should not be able to do better than random guessing.

In the last section 3.4, we introduce "iterative adversarial training", that gradually increases the value of epsilon during the training procedure, as a possible alternative to the proposed weight initialization.

3.1. Method Evaluation

To demonstrate the effectiveness of our proposed weight initialization on the adversarial trainability of deep models on larger epsilons, we run several experiments on the MNIST (LeCun & Cortes, 2010) and CIFAR10 (Krizhevsky et al.) datasets.

3.1.1. Experimental setup

- **Model architecture**: For MNIST, we use a convolutional neural network architecture (Lecun & Bengio, 1995) that is obtained from the TensorFlow tutorial (Hope et al., 2017) and for CIFAR10, we use standard ResNet-50 architecture (He et al., 2015b).

- **Base model training**: For training the MNIST models, we scale the pixel intensities between 0 and 1. For the optimization, we use the cross-entropy loss, Adam optimizer (Kingma & Ba, 2014) with learning rate = 0.001 and batch size = 128. In standard training, we use only 5 epochs. For adversarial training, we used 30 epochs, signed Projected Gradient Descent (PGD) attack (Madry et al., 2017) with the random start, PGD learning rate = 0.05 and number of steps = epsilon$\times 2.5$ / PGD learning rate. For CIFAR10, we scale the pixel intensities between 0 and 1. For the optimization, we also use the cross-entropy loss, SGD optimizer with the learning rate schedule being 0.1 for the first 50 epochs, 0.01 for the second 50, and 0.001 for the third 50, all of them with the weight decay = 5×10^{-4}, batch size = 128 and also data augmentation. Specifically, for adversarial training, we train for 150 epochs, based on signed PGD with the random start, learning rate = $2/255$ and steps = epsilon$\times 2.5$ / PGD learning rate. Note that many of these settings are obtained from (Engstrom et al., 2019b).

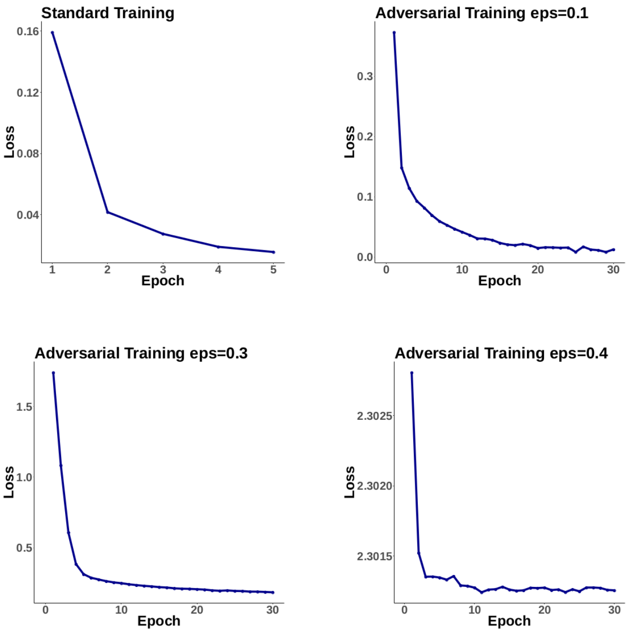


Figure 1. Training adversarial loss against the epoch number for various epsilons in the MNIST dataset. Adversarial training fails to decrease the training adversarial loss on epsilons greater than ≈ 0.35.

- **Proposed method setup**, which we call as "Extended adversarial training": For both MNIST and CIFAR10, we trained our models 5 epochs with the new larger epsilons with the same exact setting as the ones in the base model training, except that for CIFAR10, we do not use data augmentation.

3.1.2. Experiments

Experiment 1: When we try to apply adversarial training on large epsilons, the training loss does not decrease. We show the training adversarial loss across different epsilons on both MNIST (Fig. 1) and CIFAR10 (Supplementary Fig. 9 in Sec. D). We also try to adversarially train the models by adopting different choices of architectures, optimizers, learning rates, weight initialization, and different settings of PGD. These are described in more detail in the Supplementary Materials B.

Conclusion: Overall, it seems that these modifications are not playing a major role in the adversarial trainability of deep models on large epsilons.

Experiment 2: We take the weights from an adversarially trained model on epsilon = 0.3 in MNIST and epsilon = $8/255$ in CIFAR10 as the initial weights, followed by adversarial training on larger epsilons. We report the results in Fig. 2.

Conclusion: This initialization makes the adversarial learning possible on larger epsilons in both datasets.

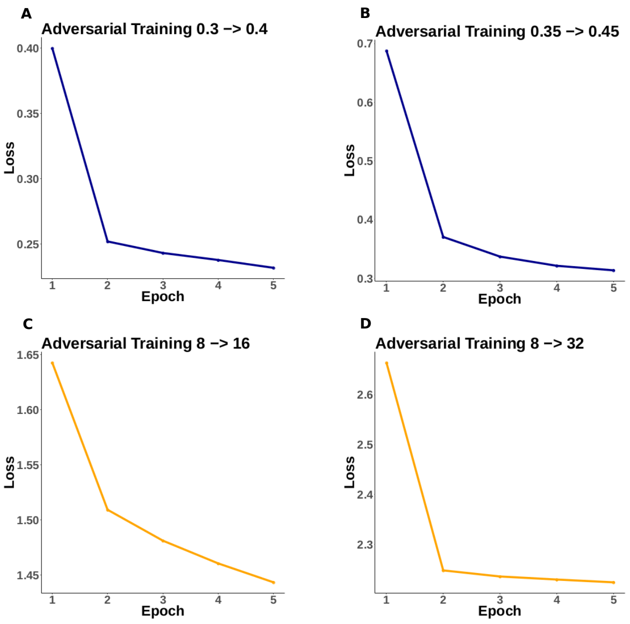


Figure 2. Using the new weight initialization based on an already adversarially trained model on a smaller epsilon makes training on larger epsilons possible. Training from epsilon = 0.3 to 0.4 (A), epsilon = 0.35 to 0.45 (B), both on MNIST. Training from epsilon = $8/255$ to $16/255$ (C), and epsilon = $8/255$ to $32/255$ (D) both on CIFAR10.

Remark 1: Initializations based on the standard trained models are also observed to be ineffective in this context, which suggests that weights of an adversarially trained model are inherently different from a standard trained model (Goodfellow et al., 2014).

3.1.3. Results

Now we compare the accuracy of our trained models with standard benchmarks in the literature (see Tables 1 and 2). We compared the results to a baseline, namely the model that is trained by the adversarial training with the commonly used epsilons in the literature. Note that one should not expect a model that is trained on a given perturbation size to resist attacks of larger magnitudes. However, this could serve as a simple baseline and give a sense of the adversarial accuracy that is gained through the proposed method. We observe that using our proposed method, we could make adversarial training on larger epsilons feasible and achieve non-trivial and significant adversarial test accuracies. We evaluate all models robustness against the signed PGD attack with 200 steps, and the other hyper-parameters stay unchanged compared to what we used for training. For the reasons that become clear later in 3.3, the PGD that is used to solve the inner maximization has a larger step size and smaller learning rate compared to the previously mentioned PGD in 3.1.1. It is worth noting that we did not try to fine-tune any of these models to improve the accuracy.

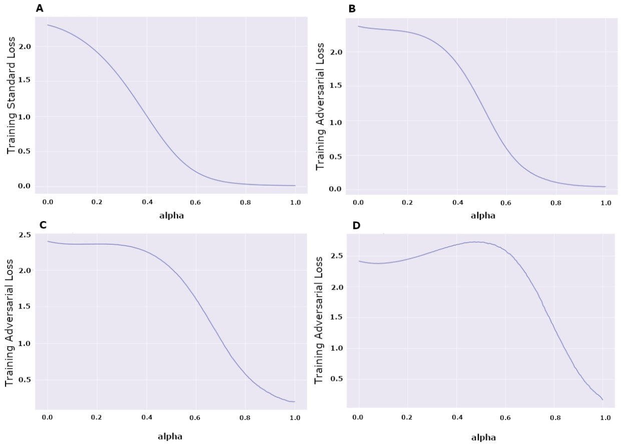


Figure 3. $\ell_\epsilon(\alpha)$, the interpolated loss function, begins to be non-convex along the direction connecting the initial random to the final trained weights. The plot is made based on 200 points along the horizontal axis. The loss is calculated on a batch of size 512 of the training data. (A) refers to the standard training, while (B), (C), and (D) correspond to epsilon = 0.2, 0.3, and 0.4, respectively.

Remark: The adversarial accuracy under the attacks with bounded ℓ_2 norm are observed to improve in the model that is trained by the proposed idea. More specifically, we consider the evaluation based on a PGD attack with epsilon $= 5$, learning rate $= 0.1$, and 125 steps. In the MNIST dataset, we obtain an adversarial test accuracy of $\approx 60\%$ for the model that is trained with epsilon $= 0.4$, compared to 30% for the model that is trained with the original adversarial training with epsilon = 0.3.

Table 1. Test accuracies of the proposed methods on MNIST compared to that of the Adversarial Training.

Model	Clean Accuracy	$\epsilon = 0.3$	$\epsilon = 0.4$
$\epsilon = 0.3 \rightarrow 0.4$	98%	94%	90%
IAT ($\epsilon = 0 \rightarrow 0.4$) (see 3.4)	98%	95%	91%
Adversarial Training ($\epsilon = 0.3$) (Madry et al., 2017)	98%	91%	15%

Table 2. Test accuracies of the proposed methods on CIFAR10 compared to that of the Adversarial Training.

Model	Clean Accuracy	$\epsilon = \frac{8}{255}$	$\epsilon = \frac{16}{255}$	$\epsilon = \frac{32}{255}$
$\epsilon = 8/255 \rightarrow 16/255$	75%	56%	32%	10%
$\epsilon = 8/255 \rightarrow 32/255$	33%	28%	24%	16%
Adversarial Training ($\epsilon = 8/255$) (Engstrom et al., 2019b)	87%	53%	18%	1%

3.2. Insights on the proposed method

In this section, we address the three questions that were raised earlier in this section. In the deep learning literature, researchers have used loss function visualization to gain insights about the training of deep models, which could not be theoretically explained. Several methods have been proposed for this purpose (Goodfellow et al., 2015; Li et al., 2017). We further use some visualization to address these questions.

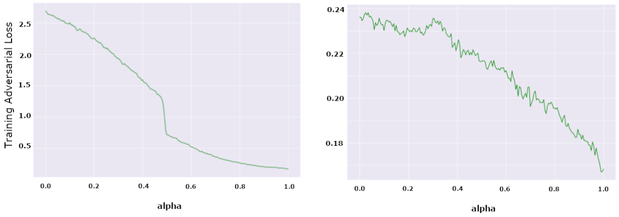


Figure 4. Left: When initialized on weights of an already adversarially trained model with epsilon = 0.3, $\ell_\epsilon(\alpha)$ could be better approximated by a convex function. Right: $\ell_\epsilon(\alpha)$ based on the initial epsilon stays low as we progress from the initial to the final weights, which implies that the training is focused on low-cost solutions for the initial epsilon (weights that are robust against the initial epsilon) along this line segment.

Visualization 1: To assess the convergence of adversarial training, we used two different plots. Specifically, the difference between training adversarial loss, and the ℓ_2 distance of weights in two consecutive epochs empirically indicate the convergence. The plots are shown in the Supplementary Materials E.

Conclusion: It seems that according to both mentioned plots, the adversarial training empirically has reached a local Nash equilibrium. This result is not unexpected as it has previously been shown that adversarial training would converge irrespective of the epsilon value in a simple model setting (Gao et al., 2019).

Visualization 2: (Goodfellow et al., 2015) proposed a method for loss landscape visualization. Specifically, the adversarial loss function denoted as $\ell_\epsilon(\theta)$, is considered for the weights θ lying on the line segment that is connecting θ_0 to θ_1, where θ_0 is the initial network weights and θ_1 is the final trained weights. $\ell_\epsilon(\theta)$ could be parametrized by α, where $\theta = (1-\alpha)\theta_0 + \alpha\theta_1$, and $0 \leq \alpha \leq 1$.

Conclusion:

- We assume that the network initial weights are random. $\ell_\epsilon(\alpha)$ is convex for small perturbations but begins to be non-convex as epsilon grows (Fig. 3). Therefore, the weights associated with low adversarial loss may not be easily reachable from a random initialization through gradient-based optimization for a large epsilon.

- Now, assume that the initial weights are obtained from an adversarially trained model. The $\ell_\epsilon(\alpha)$ seems to be well approximated by a convex function (Fig. 4; Left). Therefore, solutions with low training adversarial loss become reachable from the initial weights.

Visualization 3: Mode connectivity (Dräxler et al., 2018; Freeman & Bruna, 2016; Garipov et al., 2018) is an unexpected phenomenon in the loss landscape of deep nets. There have been some efforts in the literature to give a theoretical explanation of this effect (Kuditipudi et al., 2019). We made a similar plot to the visualization 2, with the difference that training adversarial loss is calculated according to the initial epsilon (Fig. 4; Right).

Conclusion: The given plot suggests that the proposed initialization makes the adversarial training focused on a small weight subspace. Because the training adversarial loss, when evaluated based on the initial epsilon, does not increase, and even decreases slightly. Therefore, we are exploring the weights that are robust against the initial epsilon as opposed to the entire weight space. This suggests "mode connectivity" in our training and therefore, we could expect the optimization to converge quickly.

3.3. Evaluating adversarial robustness

As already mentioned, evaluating against the adversarial attacks has proven to be extremely tricky. Here, we are inspired by the latest recommendations on evaluating adversarial robustness (Carlini et al., 2019). Note that our threat model is simple, and is indeed similar to the one that is used in the evaluation of adversarial training. Motivated by the recently recommended checklist for evaluation of adversarial robustness, we applied various sanity checks such as black-box attacks (Papernot et al., 2016), gradient-free (Chen et al., 2017), brute force attacks, and a novel semi-black-box attack to make the threat model broader. We found out that the proposed defenses pass all these checklists on MNIST. The details of the evaluation setup and the results for these experiments are explained in the Supplementary Materials C.

Building upon the mentioned recommendation list, we further evaluated our models based on more challenging tests. These are designed to increase our confidence in the claims that we make about the model robustness that is achieved by the proposed method.

We first decrease the learning rate of PGD to 0.01 and increased the number of steps to 1000 to find a stronger attack. In the model that is trained based on our proposed initialization with target epsilon = 0.5, this led to adversarial accuracy of 30%, as opposed to 80% that is obtained when the same PGD as in the training is used for the test-time attack. However, the mentioned stronger attack does not significantly affect the adversarial accuracy of the model with target epsilon = 0.4. To make the attack even stronger we used the actual gradient, as opposed to the signed gradient in PGD (Madry et al., 2017), with the learning rate = 5×10^4 and step number $= 2000$. We observed that the adversarial accuracy dropped to 14% - 20% for the former model, but this again did not affect adversarial accuracy of the model with target epsilon = 0.4. Notably, we could not decrease validation adversarial accuracy of the model with the target epsilon = 0.4 although we tried out various PGD settings.

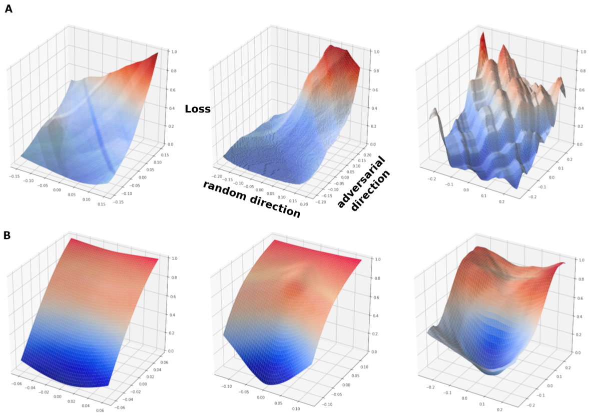


Figure 5. The loss function is visualized along the random and adversarial directions on a test data point. (A): The plots for the MNIST data, for epsilon = 0.3, 0.4, and 0.5 from left to right. The models do not show significant signs of gradient obfuscation, which is a sudden jump or non-smoothness of the loss, for epsilon up to around 0.4. (B): In CIFAR10, the plots are made for models that are trained with epsilons $16/255$, $32/255$, $64/255$ from left to right. We do not observe gradient obfuscation up until $64/255$. For MNIST, we used 10^4 points on the horizontal plane and we considered half of the corresponding epsilon along both directions. For CIFAR10, we used 50^2 points and considered epsilon along both directions. Note that, *in the MNIST dataset*, the "clean" loss landscape is not observed on all test samples even in the original adversarial training. However, for the majority of such samples, we get the desired loss landscape.

One could use the loss function visualization along the adversarial and random direction in the input space to assess the gradient obfuscation. This plot suggests that the model with the target epsilon = 0.4 does not exhibit gradient obfuscation, while the model with epsilon = 0.5 shows signs of gradient obfuscation (Fig. 5).

To go further, we used the PGD with learning rate = 0.005 and 200 steps for the inner maximization in adversarial training from epsilon = 0.3 to target epsilon = 0.5. We noted that the adversarial training loss increased in the final model from 0.6 to 1.2, and test adversarial accuracy based on the same PGD decreased significantly from 80% to 30%. Surprisingly, unlike the model that is trained with a weaker PGD, we could successfully attack the trained model with the stronger PGD even with the weaker attack that was introduced in 3.1.1.

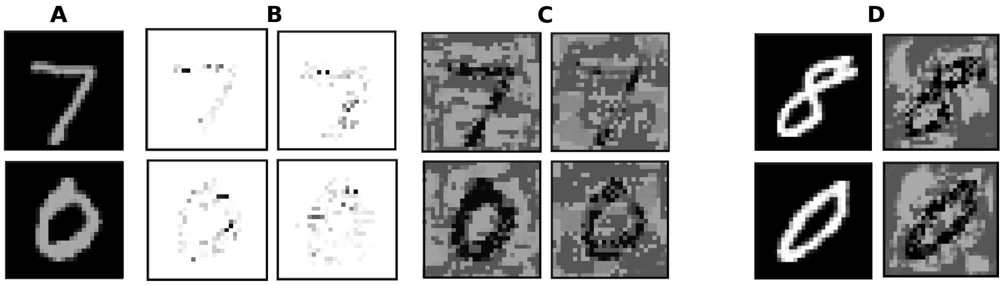


Figure 6. (A): Two test samples from MNIST. (B) The loss gradient (saliency map) with respect to the perturbed input (of epsilon = 0.3) that is obtained for the model trained on epsilon = 0.3 (Right) and the model that is trained on epsilon from 0.3 to 0.4 (Left). (C) The perturbation (of size epsilon = 0.45) for the corresponding models in (B). (D) Left: the original image. Right: the perturbations that are obtained from the model trained on epsilon from 0.3 to 0.5, based on the attack obtained from the actual gradient that is mentioned in Sec. 3.3. We note that the attack that is based on the actual gradient exhibits better interpretability even in the model trained with the original adversarial training. Recall that this model shows obvious signs of gradient obfuscation. Note that we did not apply random restart in any of these plots, and the images are scaled for visualization purposes. Overall, training on larger perturbations leads to stronger interpretability of both the model and the attacks.

This is analogous to the remedy that is used to avoid gradient obfuscation in FGSM by increasing the number of steps, which led to emergence of the PGD attack. Indeed, the first attempt to train the model using a PGD with a small number of steps has overfitted to this weak attack. We believe that by making the inner maximization more accurate, one could not even be able to train a model at epsilon = 0.5.

Note that for CIFAR10, as the adversarial accuracy is already low for large epsilons, we do not need to evaluate the model on the stronger attack. We included the gradient obfuscation plots for CIFAR10 in Fig. 5, which shows a similar trend to those of MNIST.

3.4. Iterative adversarial training (IAT)

Inspired by previously suggested “extended model training”, we propose a new method, which we call “Iterative Adversarial Training” (IAT). In IAT, the epsilon is gradually increased from 0 to the target epsilon with a specific schedule across epochs during the training.

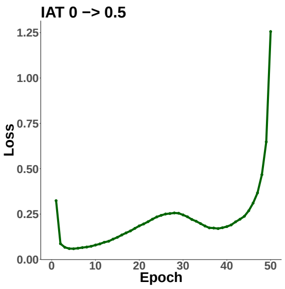


Figure 7. Training adversarial loss against the number of epochs for the proposed iterative adversarial training (IAT) algorithm on MNIST.

For the MNIST dataset, we used the same model as described earlier. The only difference here is that we increase epsilon by 0.01 in each epoch of adversarial training. In addition, we could identify the stopping point for epsilon by plotting the training adversarial loss against epochs and stop as soon as a significant sudden increase is observed (Fig. 7).

For the CIFAR10 dataset, we tested several schedules (e.g. linear and exponential schedules) with different settings, but unfortunately, the trained models either did not show a good adversarial accuracy or got overfitted.

4. Benefits of Adversarial Robustness

The interpretability of the machine learning models has emerged as an essential property of the model in many real-world applications. These, include areas where an explanation on the model prediction is required for either validation or providing overlooked insights about the input-output association to a human user. In addition to the recent efforts to understand the internal logic of these models (Montavon et al., 2015; Selvaraju et al., 2016), it has been observed that interpretability comes as an unexpected benefit of adversarial robustness. For instance, the saliency map that highlights influential portions of the input becomes sparser and more meaningful in the robust models (Tsipras et al., 2018).

The last layer of a deep neural network can be thought of as the representation that is learned by the model. This representation could be divided into a set of robust and non-robust features (Ilyas et al., 2019). Robust features are the ones that are highly correlated to the class label even in the presence of adversarial perturbations. These features, therefore, have a close relation to the human perception and changes in these features can affect the meaning of data for a human being. In contrast, non-robust features show a brittle correlation to the class label and therefore are not aligned with the human perception. During the training process, a network has a bias towards learning non-robust features (Engstrom et al., 2019a; Ilyas et al., 2019). Adversarial training is a natural remedy that prevents the network from learning and relying on non-robust features. By training a network on lager perturbations, we narrow the set of non-robust features that classifier can learn and as a result, the loss gradient with respect to the input relies on more robust features and become more interpretable. Therefore, we would expect more relevant saliency maps for such a model.

Next, we assess this property in our learned models and compare them to models that are robust to smaller perturbations (Fig 6). Specifically, we observe that the saliency map that is obtained from the loss gradient with respect to the input, evaluated on the perturbed data, in the model trained on larger epsilon, becomes more compact and concentrated around the foreground. Relevant to this, we also plotted the adversarial perturbation on the model that is trained on the MNIST dataset with epsilon = 0.4 and compared it to a network that is trained on epsilon = 0.3. The PGD attack is based on epsilon = 0.45. We observe that the attack on both models has aimed to obscure the digit, but the attack from the model with higher epsilon would erase the digit more precisely. It has been reported that the attack on adversarially trained models lacks interpretability (Schott et al., 2018; Sharma & Chen, 2018), however, our model appears to yield more interpretable attacks compared to the baseline.

5. Theoretical Results

We first briefly recap the definition of classification error rate and classification adversarial error rate and then, we define 1. optimal classifier, 2. $(p, \epsilon)-$optimal robust classifier, 3. optimal classification error rate and 4. $(p, \epsilon)-$optimal classification error rate with respect to the given distribution. Next, we discuss such optimal classifiers and their corresponding error rates under two specific distributions.

5.1. Basics and definitions

In the subsequent definitions, we let $L = \{1, 2, \ldots, c\}$ be the set of labels. We also, let $D : \mathbb{R}^d \times L \to \mathbb{R}$ denote the joint feature and label distribution.

Definition 1. *The error rate of a classifier $h : \mathbb{R}^d \to L$ on the distribution D is defined as:*

$$R(h, D) := \mathbb{E}_{x,y \,\sim\, D} \left\{\mathbb{I}(h(x) \neq y)\right\}, \tag{3}$$

where $\mathbb{I}(.)$ is the indicator function that takes 1 if its input is a true logical statement, and 0 otherwise.

Definition 2. *The classifier $h : \mathbb{R}^d \to L$ is said to be an optimal classifier on D if for any other classifier $h' : \mathbb{R}^d \to L$, $R(h, D) \leq R(h', D)$. $R(h, D)$ is also called the optimal Bayes classification error rate on D.*

Definition 3. *Let $x \in \mathbb{R}^d$ be a point in the input space. Let $p, \epsilon \in \mathbb{R}^+$. Then, the perturbation set $B_p^\epsilon(x)$ is defined as:*

$$B_p^\epsilon(x) := \left\{x' \in \mathbb{R}^d \ : \ \|x - x'\|_p \leq \epsilon\right\}, \tag{4}$$

where $\|.\|_p$ is the ℓ_p norm of a vector.

Definition 4. *Let $B_p^\epsilon(x)$ be a perturbation set. Then, adversarial error rate of a classifier $h : \mathbb{R}^d \to L$ on the perturbation set $B_p^\epsilon(x)$ and the distribution D is defined as:*

$$R_{p,\epsilon}^{(adv)}(h, D) := \mathbb{E}_{x,y \,\sim\, D} \left\{\mathbb{I}(\exists \delta \in B_p^\epsilon : h(x + \delta) \neq y)\right\}. \tag{5}$$

Definition 5. *The classifier $h : \mathbb{R}^d \to L$ is said to be a (p, ϵ)-optimal robust classifier on D if for any other classifier $h' : \mathbb{R}^d \to L$, $R_{p,\epsilon}^{(adv)}(h, D) \leq R_{p,\epsilon}^{(adv)}(h', D)$. $R_{p,\epsilon}^{(adv)}(h, D)$ is also called the (p, ϵ)-optimal classification error rate on D.*

As the optimal and the optimal robust classifiers tend to solve completely different problems by the definition, they can be different functions for a given distribution. One can easily show that they can be different functions in the general case. Consider a very simple distribution on $X \subset \mathbb{R}^d$ and $L = \{1, -1\}$, which contains just two points x_1 and x_2, $D(x_1, 1) = D(x_2, -1) = 1/2$. Assume that $\|x_1 - x_2\|_p \leq \epsilon$. The optimal classifier outputs 1 and -1, when the input is x_1 and x_2, respectively. However, the (p, ϵ)-optimal robust classifier outputs a constant, either 1 or -1.

5.2. Optimal classifiers

In the standard setting, we can find the optimal classifier for a given distribution by assuming full access to the joint data and label distribution efficiently, specifically, through applying the Bayes optimal classifier theorem (Devroye et al., 1997). However, the problem of finding the $(p, \epsilon)-$optimal robust classifier in this setting is computationally hard in general.

Theorem 1. *The problem of finding the optimal robust classifier given the joint data and label distribution $D(x, y)$ for the perturbations as large as ϵ in ℓ_p norm, is an $\mathcal{NP}$-hard problem.*

It was shown that adversarial robustness might come at the cost of higher training time (Madry et al., 2017), requiring more data (Schmidt et al., 2018), and losing the standard accuracy (Tsipras et al., 2018). In addition, we show that finding the optimal robust classifier is an $\mathcal{NP}$-hard problem. Note that this computational difficulty is different from the ones that arise in finding the global minimum of a non-convex function. Because by assuming an infinite amount of training data, instead of minimizing the non-convex empirical loss, we can well approximate the data distribution and therefore, we can efficiently find an optimal Bayes classifier. However, in this case, finding an optimal robust classifier is still computational prohibitive in the general case.

We note that for several distributions, one could efficiently find the optimal robust classifier. These include the Gaussian and mixture of Bernoulli distributions.

5.3. The Isotropic Gaussian model

Definition 6. *Isotropic Gaussian class conditional distributions: We first sample $y \in \{-1, 1\}$ uniformly and then sample a data point in $\mathbb{R}^d$ from a multivariate Gaussian distribution $N(\mu_y, \sigma^2 I)$, where without loss of generality $\mu_1 = -\mu_{-1}$.*

Theorem 2. *Let $(x|y) \sim N(\mu_y, \sigma^2 I)$, and $y \in \{-1, 1\}$ with equal probabilities, then assuming that the $(2, \epsilon)$-optimal robust classifier has a continuous decision boundary, $f(x) = 2\mathbb{I}(w^\top x \geq 0) - 1$ is $(2, \epsilon)$-robust , where $w = \mu_1 - \mu_{-1}$. Furthermore, the optimal Bayes and optimal robust classification error rates are $\Phi\left(\frac{-\|\mu_1 - \mu_{-1}\|_2}{2\sigma}\right)$, and $\Phi\left(\frac{-\|\mu_1 - \mu_{-1}\|_2 + 2\epsilon}{2\sigma}\right)$, respectively, where Φ is the cumulative distribution function for the standard normal distribution.*

5.4. The multi-dimensional Bernoulli model

Definition 7. *Uniform mixture of two multivariate Bernoulli distributions, Bernoulli$(\theta^{(1)}, \theta^{(-1)}, t)$: Let $\theta_i^{(1)}$ and $\theta_i^{(-1)} \in \{-1, 1\}$ for all $1 \leq i \leq d$, and $t \in \mathbb{R}^+$. We first randomly sample the class label y uniformly from $\{-1, 1\}$. Then, we sample each dimension of x according to a Bernoulli distribution:*

$$x_i = \begin{cases} \theta_i^{(y)}, & \text{with probability } 1/2 + t \\ -\theta_i^{(y)}, & \text{with probability } 1/2 - t \end{cases} \tag{6}$$

Therefore, the conditional distribution of x given y becomes:

$$D(x|y) = (1/2 + t)^{d - \|x - \theta^{(y)}\|_0} (1/2 - t)^{\|x - \theta^{(y)}\|_0} \tag{7}$$

Theorem 3. *Let Bernoulli $(\theta^{(1)}, \theta^{(-1)}, t)$ be a Bernoulli model. Let $(x, y) \sim$ Bernoulli$(\theta^{(1)}, \theta^{(-1)}, t)$. Then, 1. the optimal Bayes' classifier $h^\star : \{-1, 1\}^d \rightarrow \{-1, 1\}$ on this distribution is of the form:*

$$h^\star(x) = \begin{cases} 1, & \|x - \theta^{(1)}\|_0 \leq \|x - \theta^{(-1)}\|_0 \\ -1, & \text{otherwise}, \end{cases} \tag{8}$$

and 2. if k is even, the optimal classification error rate is:

$$\sum_{i=0}^{k/2} \binom{k}{i} (1/2 + t)^i (1/2 - t)^{k-i} - 1/2 \binom{k}{k/2} (1/2 + t)^{k/2} (1/2 - t)^{k/2} \tag{9}$$

Otherwise, if k is odd, the classification optimal error rate would be:

$$\sum_{i=0}^{\lfloor k/2 \rfloor} \binom{k}{i} (1/2 + t)^i (1/2 - t)^{k-i}, \tag{10}$$

where $k := \|\theta^{(1)} - \theta^{(-1)}\|_0$, which is the number of dimensions that $\theta^{(1)}$ and $\theta^{(-1)}$ do not agree. We call these dimensions as the "effective dimensions".

Assumption 1. *We assume that for $\forall u, v \in \{-1, 1\}^d$ and $\|u' - \theta'^{(1)}\|_0 = \|v' - \theta'^{(1)}\|_0$, the $(0, \epsilon)$-optimal robust classifier would give the same label to u and v. Note that, this is a rational assumption because of the symmetry of the problem.*

Theorem 4. *Let Bernoulli $(\theta^{(1)}, \theta^{(-1)}, t)$ be a Bernoulli model. Let $(x, y) \sim$ Bernoulli$(\theta^{(1)}, \theta^{(-1)}, t)$. Let $B_0^\epsilon(x)$ be the perturbation set. Then, under the assumption 1, 1. the $(0, \epsilon)$-optimal robust classifier $h^\star : \{-1, 1\}^d \rightarrow \{-1, 1\}$ on this distribution is of the form:*

$$h^{\star(adv)}(x) = \begin{cases} 1, & \|x' - \theta'^{(1)}\|_0 \leq s \\ -1, & \text{otherwise}, \end{cases} \tag{11}$$

where $s \in \{0, 1, ..., k\}$ is a threshold, and x', and $\theta'^{(1)}$ are the restrictions of x, and $\theta^{(1)}$ to the effective dimensions, defined in theorem 3, respectively, and 2. the classification $(0, \epsilon)$-optimal error rate would be:

$$R_{0,\epsilon}^{(adv)}(h, D) = 1/2 \sum_{i=0}^{s+\epsilon} \binom{k}{i} (1/2 + t)^i (1/2 - t)^{k-i} + 1/2 \sum_{i=0}^{k-s+\epsilon-1} \binom{k}{i} (1/2 + t)^i (1/2 - t)^{k-i} \tag{12}$$

Theorem 5. *Let Bernoulli $(\theta^{(1)}, \theta^{(-1)}, t)$ be a Bernoulli model. Let $(x, y) \sim$ Bernoulli$(\theta^{(1)}, \theta^{(-1)}, t)$. Let $B_\infty^\epsilon(x)$ be a perturbation set. Then, 1. the (∞, ϵ)-optimal robust classifier $h^{\star(adv)} : \{-1, 1\}^d \rightarrow \{-1, 1\}$ on this distribution for $\epsilon < 1$ is of the form:*

$$h^{\star(adv)}(x) = \begin{cases} 1, & \|T(x) - \theta^{(1)}\|_0 \leq \|T(x) - \theta^{(-1)}\|_0 \\ -1, & \text{otherwise}, \end{cases} \tag{13}$$

where $T(x) = (sign(x_1), \ldots, sign(x_d))$ and $sign(r)$ is 1 if $r \geq 0$ and is -1 otherwise, and 2. the classification

(∞, ϵ)*-optimal error rate would be:*

$$R^{(adv)}_{\infty,\epsilon}(h, D) = \begin{cases} R(h^\star, D), & \epsilon < 1 \\ 1/2, & \textit{otherwise}, \end{cases} \tag{14}$$

where $h^\star$ *is the optimal Baye's classification rule.*

Remark 1: Note that the effective dimension, k, rather than the real dimension of the input, appears in the equations for the optimal classifiers.

Remark 2: For the case of isotropic Gaussian distribution, for the ℓ_2 attack model, maximum allowed adversarial perturbation, before getting the trivial random chance adversarial accuracy is half of the ℓ_2 distance between the centers of the two Gaussians. For the mixture of multi-dimensional Bernoulli distribution, however, for the ℓ_∞ attack model, such a limit is $\epsilon = 1$, which translates to $\epsilon = 0.5$ for the mixture elements being in $\{0, 1\}$. For the ℓ_0 attack, one has to use the final equation of the proof to find the mentioned limit for epsilon, which is plotted in the Supplementary Materials G.

Detailed proofs of the theorems in this section are provided in Supplementary Materials A.

It is notable that, there is some work in the literature (Bubeck et al., 2018; Garg et al., 2019; Gourdeau et al., 2019; Mahloujifar & Mahmoody, 2018), which discussed the computational hardness of adversarial robust learning. However, none of them considered the optimal case.

6. Conclusion and future work

We demonstrated that the weight initialization plays an important role in the adversarial learnability of deep models on larger perturbations. Specifically, weights from an already robust model on a smaller perturbation size could be an effective initialization to achieve this goal. In spite of the promises in the proposed idea, some directions remain unstudied. We next provide a list of possible future directions to extend this work:

- In this work, we demonstrated the importance of weight initialization in the feasibility of adversarial training on large perturbations. However, more theoretical explanation is still essential to better understand this phenomenon from an optimization perspective, e.g. explaining the geometry of loss landscape.

- We suggest that the idea behind this method may be the key to make learning possible on large variance Gaussian noise data augmentation. This can be used to improve methods such as the randomized smoothing (Cohen et al., 2019; Salman et al., 2019) as well.

- We showed that finding the optimal robust classifier is an $\mathcal{NP}$-hard problem in general. We also showed that it can be computed for some specific distributions. However, it would be useful to understand what general properties of the data distribution make finding the optimal robust classifier efficiently possible.

Code

You can find our code in: `https://github.com/rohban-lab/Shaeiri_submitted_2020`

Acknowledgement

We would like to thank Soroosh Baselizadeh, Hossein Yousefi Moghaddam, and Zeinab Golgooni for their insightful comments and reviews of this work.

References

Athalye, A. and Carlini, N. On the robustness of the CVPR 2018 white-box adversarial example defenses. *CoRR*, abs/1804.03286, 2018. URL http://arxiv.org/abs/1804.03286.

Athalye, A., Carlini, N., and Wagner, D. A. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *CoRR*, abs/1802.00420, 2018. URL http://arxiv.org/abs/1802.00420.

Attias, I., Kontorovich, A., and Mansour, Y. Improved generalization bounds for robust learning, 2018.

Ben Daya, I., Shaifee, M., Karg, M., Scharfenderger, C., and Wong, A. On robustness of deep neural networks: A comprehensive study on the effect of architecture and weight initialization to susceptibility and transferability of adversarial attacks. *Journal of Computational Vision and Imaging Systems*, 4(1):3, Dec. 2018. doi: 10.15353/jcvis.v4i1.329. URL https://openjournals.uwaterloo.ca/index.php/vsl/article/view/329.

Ben-Tal, A., Ghaoui, L. E., and Nemirovski, A. *Robust Optimization*, volume 28 of *Princeton Series in Applied Mathematics*. Princeton University Press, 2009. ISBN 978-1-4008-3105-0.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Srndic, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. *CoRR*, abs/1708.06131, 2017. URL http://arxiv.org/abs/1708.06131.

Bubeck, S., Price, E., and Razenshteyn, I. Adversarial examples from computational constraints, 2018.

Carlini, N. Is ami (attacks meet interpretability) robust to adversarial examples? *CoRR*, abs/1902.02322, 2019. URL http://arxiv.org/abs/1902.02322.

Carlini, N. and Wagner, D. A. Defensive distillation is not robust to adversarial examples. *CoRR*, abs/1607.04311, 2016. URL http://arxiv.org/abs/1607.04311.

Carlini, N. and Wagner, D. A. Magnet and "efficient defenses against adversarial attacks" are not robust to adversarial examples. *CoRR*, abs/1711.08478, 2017. URL http://arxiv.org/abs/1711.08478.

Carlini, N. and Wagner, D. A. Audio adversarial examples: Targeted attacks on speech-to-text. *CoRR*, abs/1801.01944, 2018. URL http://arxiv.org/abs/1801.01944.

Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D. A., and Zhou, W. Hidden voice commands. In *USENIX Security Symposium*, 2016.

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I. J., Madry, A., and Kurakin, A. On evaluating adversarial robustness. *CoRR*, abs/1902.06705, 2019. URL http://arxiv.org/abs/1902.06705.

Chen, L., Min, Y., Zhang, M., and Karbasi, A. More data can expand the generalization gap between adversarially robust and standard models, 2020.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. Zoo. *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security - AISec '17*, 2017. doi: 10.1145/3128572.3140448. URL http://dx.doi.org/10.1145/3128572.3140448.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified adversarial robustness via randomized smoothing. *CoRR*, abs/1902.02918, 2019. URL http://arxiv.org/abs/1902.02918.

Cullina, D., Bhagoji, A. N., and Mittal, P. Pac-learning in the presence of evasion adversaries, 2018.

Dalvi, N., Domingos, P., Sumit, M., and Verma, S. D. Adversarial classification. In *In Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining*, pp. 99–108. ACM Press, 2004.

Devroye, L., Gyorfi, L., and Lugosi, G. *A probabilistic theory of pattern recognition*. Springer, 1997.

Diochnos, D. I., Mahloujifar, S., and Mahmoody, M. Lower bounds for adversarially robust pac learning, 2019.

Dräxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. A. Essentially no barriers in neural network energy landscape. In *ICML*, 2018.

Engstrom, L., Tsipras, D., Schmidt, L., and Madry, A. A rotation and a translation suffice: Fooling cnns with simple transformations. *CoRR*, abs/1712.02779, 2017. URL http://arxiv.org/abs/1712.02779.

Engstrom, L., Ilyas, A., Madry, A., Santurkar, S., Tran, B., and Tsipras, D. A discussion of 'adversarial examples are not bugs, they are features': Discussion and author responses. *Distill*, 2019a. doi: 10.23915/distill.00019.7. https://distill.pub/2019/advex-bugs-discussion/original-authors.

Engstrom, L., Ilyas, A., Santurkar, S., and Tsipras, D. Robustness (python library), 2019b. URL https://github.com/MadryLab/robustness.

Ford, N., Gilmer, J., Carlini, N., and Cubuk, D. Adversarial examples are a natural consequence of test error in noise, 2019.

Freeman, C. D. and Bruna, J. Topology and geometry of half-rectified network optimization, 2016.

Gao, R., Cai, T., Li, H., Wang, L., Hsieh, C.-J., and Lee, J. D. Convergence of adversarial training in overparametrized neural networks, 2019.

Garg, S., Jha, S., Mahloujifar, S., and Mahmoody, M. Adversarially robust learning could leverage computational hardness, 2019.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *NeurIPS*, 2018.

Gilmer, J., Metz, L., Faghri, F., Schoenholz, S. S., Raghu, M., Wattenberg, M., and Goodfellow, I. J. Adversarial spheres. *CoRR*, abs/1801.02774, 2018. URL http://arxiv.org/abs/1801.02774.

Globerson, A. and Roweis, S. T. Nightmare at test time: robust learning by feature deletion. In *Proceedings of the 23rd International Conference on Machine Learning*, volume 148 of *ACM International Conference Proceeding Series*, pp. 353–360. ACM, 2006. doi: 10.1145/1143844.1143889.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL http://proceedings.mlr.press/v9/glorot10a.html.

Goodfellow, I., Vinyals, O., and Saxe, A. Qualitatively characterizing neural network optimization problems. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6544.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

Gourdeau, P., Kanade, V., Kwiatkowska, M., and Worrell, J. On the hardness of robust classification, 2019.

Gowal, S., Uesato, J., Qin, C., Huang, P.-S., Mann, T., and Kohli, P. An alternative surrogate loss for pgd-based adversarial testing, 2019.

Graves, A., Mohamed, A., and Hinton, G. E. Speech recognition with deep recurrent neural networks. *CoRR*, abs/1303.5778, 2013. URL http://arxiv.org/abs/1303.5778.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015a. URL http://arxiv.org/abs/1502.01852.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015b.

Hope, T., Resheff, Y. S., and Lieder, I. *Learning TensorFlow: A Guide to Building Deep Learning Systems*. O'Reilly Media, Inc., 1st edition, 2017. ISBN 1491978511.

Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information, 2018.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Adversarial examples are not bugs, they are features, 2019. URL http://arxiv.org/abs/1905.02175. cite arxiv:1905.02175.

Kaur, S., Cohen, J., and Lipton, Z. C. Are perceptually-aligned gradients a general property of robust classifiers? *ArXiv*, abs/1910.08640, 2019.

Khim, J. and Loh, P.-L. Adversarial risk bounds via function transformation, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2014.

Kolcz, A. and Teo, C. H. Feature weighting for improved classifier robustness. In *CEAS 2009*, 2009.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

Kuditipudi, R., Wang, X., Lee, H., Zhang, Y., Li, Z., Hu, W., Arora, S., and Ge, R. Explaining landscape connectivity of low-cost solutions for multilayer nets. *CoRR*, abs/1906.06247, 2019. URL http://arxiv.org/abs/1906.06247.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial machine learning at scale, 2016a.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016b. URL http://arxiv.org/abs/1607.02533.

Lecun, Y. and Bengio, Y. *Convolutional networks for images, speech, and time-series*. MIT Press, 1995.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Li, H., Xu, Z., Taylor, G., and Goldstein, T. Visualizing the loss landscape of neural nets. In *NeurIPS*, 2017.

Li, J., Qu, S., Li, X., Szurley, J., Kolter, J. Z., and Metze, F. Adversarial music: Real world audio adversary against wake-word detection system. In *Advances in Neural Information Processing Systems 32*, pp. 11908–11918. Curran Associates, Inc., 2019.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2017.

Mahloujifar, S. and Mahmoody, M. Can adversarially robust learning leverage computational hardness?, 2018.

Maini, P., Wong, E., and Kolter, J. Z. Adversarial robustness against the union of multiple perturbation models, 2019.

Montasser, O., Hanneke, S., and Srebro, N. Vc classes are adversarially robustly learnable, but only improperly, 2019.

Montavon, G., Bach, S., Binder, A., Samek, W., and Müller, K. Explaining nonlinear classification decisions with deep taylor decomposition. *CoRR*, abs/1512.02479, 2015. URL http://arxiv.org/abs/1512.02479.

Nicolae, M.-I., Sinn, M., Tran, M. N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., Molloy, I., and Edwards, B. Adversarial robustness toolbox v1.2.0. *CoRR*, 1807.01069, 2018. URL https://arxiv.org/pdf/1807.01069.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning, 2016.

Salman, H., Yang, G., Li, J., Zhang, P., Zhang, H., Razenshteyn, I., and Bubeck, S. Provably robust deep learning via adversarially trained smoothed classifiers, 2019.

Schäfer, F. and Anandkumar, A. Competitive gradient descent. In *NeurIPS*, 2019.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. Adversarially robust generalization requires more data, 2018.

Schott, L., Rauber, J., Bethge, M., and Brendel, W. Towards the first adversarially robust neural network model on mnist, 2018.

Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., and Batra, D. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR*, abs/1610.02391, 2016. URL http://arxiv.org/abs/1610.02391.

Shafahi, A., Ghiasi, A., Huang, F., and Goldstein, T. Label smoothing and logit squeezing: A replacement for adversarial training? *ArXiv*, abs/1910.11585, 2018.

Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J. P., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. Adversarial training for free! *CoRR*, abs/1904.12843, 2019. URL http://arxiv.org/abs/1904.12843.

Sharma, Y. and Chen, P. Attacking the madry defense model with l_1-based adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=Sy8WeUJPf.

Simon-Gabriel, C.-J., Ollivier, Y., Bottou, L., Schölkopf, B., and Lopez-Paz, D. First-order adversarial vulnerability of neural networks and input dimension, 2018.

Song, C., He, K., Wang, L., and Hopcroft, J. E. Improving the generalization of adversarial training with domain adaptation. *CoRR*, abs/1810.00740, 2018. URL http://arxiv.org/abs/1810.00740.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6199.

Tramer, F. and Boneh, D. Adversarial training and robustness for multiple perturbations. In *Advances in Neural Information Processing Systems 32*, pp. 5858–5868. Curran Associates, Inc., 2019.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy, 2018. URL http://arxiv.org/abs/1805.12152. cite arxiv:1805.12152.

Uesato, J., O'Donoghue, B., van den Oord, A., and Kohli, P. Adversarial risk and the dangers of evaluating against weak attacks, 2018.

Wang, Y., Jha, S., and Chaudhuri, K. Analyzing the robustness of nearest neighbors to adversarial examples. *ArXiv*, abs/1706.03922, 2017.

Wong, E., Rice, L., and Kolter, J. Z. Fast is better than free: Revisiting adversarial training. *ArXiv*, abs/2001.03994, 2020.

Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., and Zweig, G. Achieving human parity in conversational speech recognition. *CoRR*, abs/1610.05256, 2016. URL http://arxiv.org/abs/1610.05256.

Yin, D., Ramchandran, K., and Bartlett, P. Rademacher complexity for adversarially robust generalization, 2018.

Zhai, R., Cai, T., He, D., Dan, C., He, K., Hopcroft, J. E., and Wang, L. Adversarially robust generalization just requires more unlabeled data. *CoRR*, abs/1906.00555, 2019. URL http://arxiv.org/abs/1906.00555.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy, 2019.

Appendices

A. Proofs

Definition 1. *$P_{D,p,\epsilon}$: The general problem of finding a (p,ϵ)-optimal robust classifier on the distribution D, under the perturbation set $B_p^\epsilon(x)$. An algorithmic solution to this problem would take as input D, p and ϵ and outputs a classifier $h : \mathbb{R}^d \to L$ that is (p,ϵ)-optimal robust.*

Definition 2. *$Q_{G(V,E)}$: The problem of finding the maximum distance-3 independent set in an unweighted undirected graph $G(V,E)$. We call a subset of nodes $S \subseteq V$ a distance$-d$ independent set if $\forall u_1, u_2 \in S, u_1 \neq u_2 : dist_G(u_1,u_2) \geq d$, where $dist_G(u_1,u_2)$ is length of the shortest path between u_1 and u_2 on the graph G. The goal of this problem is to find a maximum cardinality distance-3 independent set of graph $G(V,E)$.*

Lemma 1. *Every instance of $Q_{G(V,E)}$ can be reduced to an instance of $P_{D,p,\epsilon}$ in polynomial time.*

Proof. Let $|V| = k$ and consider an order $e_1, \dots, e_{\binom{k}{2}}$ of all possible edges on $G(V,E)$, where each possible edge is of the form $e_i = \{v_j, v_l\}$ and connects 2 nodes $v_j, v_l \ \in V, j \neq l$. We first construct a set of k points in $d = \binom{k}{2}$ dimensional space and then $D(x,y)$ be supported on these points. Let M be $k \times d$ matrix, which is defined as:

$$M_{i,j} := \begin{cases} 0, & v_i \notin e_j \\ \lambda, & e_j = \{v_i, v_l\} \text{ and } i > l \\ -\lambda, & e_j = \{v_i, v_l\} \text{ and } i \leq l, \end{cases} \tag{15}$$

and λ is defined for any given $p, \epsilon \in \mathbb{R},\ p, \epsilon > 0$ or $p = \infty$ as follows:

$$\lambda := \begin{cases} \frac{\epsilon}{\sqrt[p]{2k-4}}, & p \neq \infty \\ \epsilon, & p = \infty. \end{cases} \tag{16}$$

Note that the ℓ_p distance between every pair of rows in M is greater than ϵ. Now, based on M, we define a new matrix M', where if the edge $e_i \in E$ and $e_i = (v_j, v_l)$, we set $M'_{j,i} = |M_{j,i}|$ and $M'_{l,i} = |M_{l,i}|$. Note that in the matrix M', two rows whose corresponding vertices are not connected still remain greater than ϵ apart with respect to the ℓ_p norm. However, two rows that are connected in the graph $G(V,E)$ would now have an ℓ_p distance of exactly ϵ.

Next, let $L = \{1, 2, \dots, k\}$ be the set of labels. We define discrete probability distributions $D : \mathbb{R}^d \times L \to \mathbb{R}$ on points defined as rows of M'.

$$D(x = M'_i, y = j) = 1/k \times \mathbb{I}(i = j) \tag{17}$$

To find a (p,ϵ)-optimal robust classifier on the mentioned D, which is supported on a finite set of points $X := \{M'_1, \dots, M'_k\}$, we have to solve the following optimization problem:

$$\begin{aligned} &\underset{h:\mathbb{R}^d \to L}{\arg\min}\, R_{p,\epsilon}^{(adv)}(h,p) \\ &= \underset{h:\mathbb{R}^d \to L}{\arg\min}\, \mathbb{E}_{x,y\, \sim D} \left\{ \mathbb{I}(\exists \delta \in B_p^\epsilon : h(x+\delta) \neq y) \right\} \\ &= \underset{h:\mathbb{R}^d \to L}{\arg\min} \sum_{x_i \in X,\ 1 \leq y_j \leq k} \mathbb{I}\left(\exists \delta \in B_p^\epsilon : h(x+\delta) \neq y_j\right) D(x_i, y_j) \\ &= \underset{h:\mathbb{R}^d \to L}{\arg\max} \sum_{x_i \in X,\ 1 \leq y_j \leq k} \mathbb{I}\left(\forall \delta \in B_p^\epsilon : h(x_i+\delta) = y_j\right) D(x_i, y_j) \end{aligned} \tag{18}$$

By our construction of D, this would be equivalent to:

$$\begin{aligned} &\underset{h:\mathbb{R}^d \to L}{\arg\max} \sum_{x_i \in X} \mathbb{I}\left(\forall \delta \in B_p^\epsilon : h(x_i+\delta) = y_i\right) \\ &= \underset{h:\mathbb{R}^d \to L}{\arg\max} \sum_{x_i \in X} \mathbb{I}\left(\forall \delta \in B_p^\epsilon : h(x_i+\delta) = i\right) \end{aligned} \tag{19}$$

Note that by our construction of X, there is an edge between v_i and v_j iff $\|x_j - x_i\|_p \leq \epsilon$. Hence, this optimization is equivalent to:

$$\underset{h:V\to L}{\arg\max} \sum_{v_i\in V} \mathbb{I}\left(\forall u \in \text{neighbors}(v_i) : h(u) = i \text{ and } h(v_i) = i\right), \tag{20}$$

where neighbors(v) is the set of all nodes connected to the v in the graph G. Note that for the i-th term in the sum to be one, (1) v_i should be labeled as i by h, and (2) all neighbors of v_i in the graph $G(V, E)$ should be labeled as i. Therefore, for both the i-th and j-th term in the sum to be one, they should have a graph distance of at least 3. Also, if v_i and v_j have a graph distance of at least 3 then, they can both be one in the sum. As a result, the last optimization would be equivalent to:

$$\underset{U\subseteq V}{\arg\max} \sum_{u\in U} \mathbb{I}(\forall v \in U,\ v \neq u : dist_G(u, v) \geq 3). \tag{21}$$

Therefore, the last optimization would equivalently find the set of vertices $U \subseteq V$ with maximum cardinality such that, all selected vertices have a distance of at least 3 in the graph. As a result, the problem of finding a (p, ϵ)-optimal robust classifier on D would yield a solution to the maximum distance-3 independent set for the input graph $G(V, E)$. □

Theorem 1. *The problem of finding the (p, ϵ)-optimal robust classifier on the distribution D, $P_{D,p,\epsilon}$, is an $\mathcal{NP}$-hard problem.*

Proof. As shown in Lemma 1, the maximum distance-3 independent set problem can be reduced to $P_{D,p,\epsilon}$ in polynomial time. The former is a well-known $\mathcal{NP}$-hard problem, which implies that $P_{D,p,\epsilon}$ would be $\mathcal{NP}$-hard too. □

Remark 1. *For the special case of $p = 0$ in Lemma 1, one can consider $\epsilon = \binom{k}{2} - 1 = d - 1$ and define the $k \times d$ matrix M as: $M_{i,j} := (i - 1) \times d + j$. Note that the ℓ_0 distance between every pair of rows in M is exactly d. Now, based on M, we can again define a new matrix M', where if the edge $e_i \in E$ and $e_i = (v_j, v_l)$, we set $M'_{j,i} = \min(M_{j,i}, M_{l,i})$ and $M'_{l,i} = \min(M_{j,i}, M_{l,i})$. Note that in the matrix M', two rows whose corresponding vertices are not connected still remain exactly d apart with respect to the ℓ_0 norm. However, two rows that are connected in the graph $G(V, E)$ would now have an ℓ_0 distance of exactly $d - 1$.*

Remark 2. *The most general case of the perturbation set can be defined by using a relation $Re \subseteq \mathbb{R}^d \times \mathbb{R}^d$. Let $x \in \mathbb{R}^d$ then, the perturbation set is defined as: $B(x) := \left\{x' \in \mathbb{R}^d\ :\ (x, x') \in Re\right\}$. In this general case making a relation Re using graph $G(V, E)$, and specifying coordinates of the points is trivial.*

Theorem 2. *Let $(x|y) \sim N(\mu_y, \sigma^2 I)$, and $y \in \{-1, 1\}$ with equal probabilities, then assuming that the $(2, \epsilon)$-optimal robust classifier has a continuous decision boundary, $f(x) = 2\mathbb{I}(w^\top x \geq 0) - 1$ is $(2, \epsilon)$-robust , where $w = \mu_1 - \mu_{-1}$. Furthermore, the optimal Bayes and optimal robust classification error rates are $\Phi\left(\frac{-\|\mu_1-\mu_{-1}\|_2}{2\sigma}\right)$, and $\Phi\left(\frac{-\|\mu_1-\mu_{-1}\|_2+2\epsilon}{2\sigma}\right)$, respectively, where Φ is the cumulative distribution function for the standard normal distribution.*

Proof. A binary classifier can be expressed as $h : \mathbb{R}^d \to \{-1, 1\}$. We aim to find the optimal robust classifier $h^{\star(adv)}$:

$$h^{\star(adv)} = \underset{h}{\arg\min}\, \mathbb{E}_{X,Y}\left\{\max_{\|\delta\|_2\leq\epsilon} |h(X + \delta) - Y|\right\}, \tag{22}$$

where Y is the true label of the data point X.

Let f_1 and f_{-1} denote the conditional densities of X given Y, $N(\mu_1, \sigma^2 I)$ and $N(\mu_{-1}, \sigma^2 I)$, accordingly. Then, we have:

$$\mathbb{P}(Y = i|X = x) = \frac{f_i(x)}{f_1(x) + f_{-1}(x)}, \qquad i \in \{1, -1\}, \tag{23}$$

and we get:

$$\begin{aligned}
&\mathbb{E}_{X,Y}\left\{\max_{\|\delta\|_2\leq\epsilon}|h(X+\delta)-Y|\right\} \\
&=\mathbb{E}_X\Bigg\{\frac{f_1(x)}{f_1(x)+f_{-1}(x)}\max_{\|\delta\|_2\leq\epsilon}|h(x+\delta)-1|+ \\
&\qquad\qquad\frac{f_{-1}(x)}{f_1(x)+f_{-1}(x)}\max_{\|\delta\|_2\leq\epsilon}|h(x+\delta)+1|\Bigg\} \\
&=\frac{1}{2}\int_{\mathbb{R}^d}(f_1(x)\max_{\|\delta\|_2\leq\epsilon}|h(x+\delta)-1|\;+\;f_{-1}(x)\max_{\|\delta\|_2\leq\epsilon}|h(x+\delta)+1|)dx
\end{aligned} \tag{24}$$

Definition 3. *Decision Boundary of classifier h is defined as*

$$DB(h):=\{x\mid\exists C>0\;\forall 0<\epsilon\leq C,\;\;\exists u\in\mathbb{R}^d,\;\;u\neq 0,\;\;\|u\|_2\leq\epsilon,\;h(x+u)\neq h(x)\} \tag{25}$$

In what follows, we focus only on the candidate classifiers that have a continuous decision boundary.

Definition 4. *$\beta_\epsilon(x)$ is defined as the set of points that are within ℓ_2 ball of radius ϵ around x, excluding the outer boundary. Also, the critical region of h, denoted as $CR(h)$, is defined as the set of points that can be misclassified by adding a perturbation of length no more than ϵ.*

$$CR(h):=\{x\mid\exists\delta\in\mathbb{R}^d,\;\;\|\delta\|_2\leq\epsilon,\;\;h(x)\neq h(x+\delta)\} \tag{26}$$

Lemma 2. $CR(h)=\bigcup_{x\in DB(h)}\beta_\epsilon(x)$

Proof. Suppose that $x\in CR(h)$. Therefore, there exists δ such that $\|\delta\|_2\leq\epsilon$ and $h(x)\neq h(x+\delta)$. Since the label of the data points, x and $x+\delta$ are different and as we assumed that $DB(h)$ is continuous, there must exist $k\in(0,1)$ such that $x+k\delta\in DB(h)$. Note that

$$\|x+k\delta-x\|_2=\|k\delta\|_2=k\|\delta\|_2<\|\delta\|_2\leq\epsilon\implies x+k\delta\in\beta_\epsilon(x),$$

and according to Definition 4, $CR(h)\subseteq\bigcup_{x\in DB(h)}\beta_\epsilon(x)$.

Now suppose that $x\in\bigcup_{x\in DB(h)}\beta_\epsilon(x)$. Then, there exists $x_0\in DB(h)$ such that $x\in\beta_\epsilon(x_0)$. But note that

$$x\in\beta_\epsilon(x_0)\implies\|x-x_0\|_2<\epsilon$$

$$x_0\in DB(h)\implies\exists u:\|u\|_2\leq\epsilon-\|x-x_0\|_2, h(x_0+u)\neq h(x_0)$$

As a result, $\|x-(x_0+u)\|_2\leq\|x-x_0\|_2+\|u\|_2\leq\epsilon$.

Note that as $\|x-x_0\|_2<\epsilon$, $h(x)\neq h(x+(x_0-x))=h(x_0)$, if $h(x_0)\neq h(x)$, $x\in CR(h)$. Otherwise,

$$h(x_0)=h(x),\quad h(x_0+u)\neq h(x_0)\implies h(x_0+u)\neq h(x),$$

$$\|x-(x_0+u)\|_2\leq\epsilon\implies x\in CR(h).$$

So, in both cases, $\bigcup_{x\in DB(h)}\beta_\epsilon(x)\subseteq CR(h)$. □

According to the Definition 4, for all $x\in CR(h)$, there exist two points like $x_0=x$, x_1 within ϵ distance of x, such that $h(x_0)=1$, $h(x_1)=-1$. Therefore, $\max_{\|\delta\|_2\leq\epsilon}|h(x+\delta)-1|=\max_{\|\delta\|_2\leq\epsilon}|h(x+\delta)+1|=2$. As a result, we have

$$\frac{1}{2}\int_{CR(h)}(f_1(x)\max_{\|\delta\|_2\leq\epsilon}|h(x+\delta)-1|+f_{-1}(x)\max_{\|\delta\|_2\leq\epsilon}|h(x+\delta)+1|)dx=\int_{CR(h)}(f_1(x)+f_{-1}(x))dx. \tag{27}$$

Definition 5. *Safe Region of the class i, denoted as $SR_i(h)$, is defined as the set of points that are labeled as i by the classifier h, and cannot be misclassified when a perturbation of size less than or equal to ϵ is added.*

According to the definition above, for all $x_0 \in SR_i(h)$, we get

$$\forall \delta \ \|\delta\|_2 \leq \epsilon \quad h(x_0) = h(x_0 + \delta) = i.$$

As a result, for such x_0, $\max_{\|\delta\|_2 \leq \epsilon} |h(x_0 + \delta) - 1| = |i - 1|$ and $\max_{\|\delta\|_2 \leq \epsilon} |h(x_0 + \delta) + 1| = |i + 1|$. Therefore, the value of

$$\frac{1}{2} \int_{SR_i(h)} f_1(x) \max_{\|\delta\|_2 \leq \epsilon} |h(x + \delta) - 1| \ + \ f_{-1}(x) \max_{\|\delta\|_2 \leq \epsilon} |h(x + \delta) + 1| \, dx$$

for $i = 1, -1$ is

$$\int_{SR_1(h)} f_{-1}(x)dx, \qquad \text{and} \qquad \int_{SR_{-1}(h)} f_1(x)dx, \tag{28}$$

respectively.

As $\mathbb{R}^d = CR(h) \bigcup SR_1(h) \bigcup SR_{-1}(h)$, and according to Eqs. 24, 27, and 28, we have

$$\begin{aligned}\mathbb{E}_{X,Y} \left\{ \max_{\|\delta\|_2 \leq \epsilon} |h(x + \delta) - y| \right\} &= \int_{CR(h)} (f_1(x) + f_{-1}(x))dx \ + \ \int_{SR_1(h)} f_{-1}(x)dx \\ &\quad + \int_{SR_{-1}(h)} f_1(x)dx = 2 - \int_{SR_1(h)} f_1(x)dx - \int_{SR_{-1}(h)} f_{-1}(x)dx\end{aligned} \tag{29}$$

So the solution to optimal ϵ robust classifier in Eq. 26, is equivalent to the solution of

$$\arg\max_h \int_{SR_1(h)} f_1(x)dx + \int_{SR_{-1}(h)} f_{-1}(x)dx \tag{30}$$

Let E be a unit vector parallel to the line passing means of the normal distributions. That is, $E := \frac{\mu_1 - \mu_{-1}}{\|\mu_1 - \mu_{-1}\|_2}$. Define $SR'_i(h) = \{x - i\epsilon E \mid x \in SR_i(h)\}$ and f'_i as the density of the normal distributions with mean of $\mu_1 - i\epsilon E$ and covariance $\sigma^2 I$, for $i = 1, -1$. Then, we have

$$\begin{aligned}&\int_{SR_1(h)} f_1(x)dx + \int_{SR_{-1}(h)} f_{-1}(x)dx \\ &\quad = \int_{SR'_1(h)} f_1(x + \epsilon E)dx + \int_{SR'_{-1}(h)} f_{-1}(x - \epsilon E)dx \\ &\quad = \int_{SR'_1(h)} f'_1(x)dx + \int_{SR'_{-1}(h)} f'_{-1}(x)dx\end{aligned} \tag{31}$$

Lemma 3. $SR'_1(h) \bigcap SR'_{-1}(h) = \emptyset$.

Proof. Otherwise, there exists $x \in SR'_1(h), x \in SR'_{-1}(h)$, such that

$$\begin{aligned} x - \epsilon E \in SR_{-1}(h) &\overset{\|\epsilon E\|_2 = \epsilon}{\Longrightarrow} h(x - \epsilon E) = h(x) = -1 \\ x + \epsilon E \in SR_1(h) &\overset{\|\epsilon E\|_2 = \epsilon}{\Longrightarrow} h(x + \epsilon E) = h(x) = 1,\end{aligned}$$

which is a contradiction. □

So every $x \in \mathbb{R}^d$, belongs to at most one of $SR'_1(h)$ and $SR'_{-1}(h)$ and so, $SR'_{-1}(h) \subseteq \mathbb{R}^d \backslash SR'_1(h)$. This implies that

$$\int_{SR'_1(h)} f'_1(x)dx + \int_{SR_{-1}(h)} f'_{-1}(x)dx \leq \int_{SR'_1(h)} f'_1(x)dx + \int_{\mathbb{R}^d \backslash SR'_1(h)} f'_{-1}(x)dx \tag{32}$$

Now, let $DB^\star$ be the hyperplane perpendicular to E, $h^\star$ be the classifier whose decision boundary is $DB^\star$, and $h^\star(\mu_1) = 1, h^\star(\mu_{-1}) = -1$. Then

$$\forall x \text{ s.t. } h^\star(x) = i, \quad \|x - \mu_i\|_2 \leq \|x - \mu_{-i}\|_2$$

As f'_1 and f'_{-1} are densities of the normal distributions with the same covariance, for ϵ less than half of the distance between μ_1 and μ_{-1}, if $\|x-\mu_1\|_2 \leq \|x-\mu_{-1}\|_2$, then $f'_1(x) \geq f'_{-1}(x)$. Further, if $\|x-\mu_1\|_2 \geq \|x-\mu_{-1}\|_2$, then $f'_1(x) \leq f'_{-1}(x)$.

Now define $C_i = \{x | h^\star(x) = i\}$. By definition, $C_1 \bigcup C_{-1} = \mathbb{R}^d$. Then, $\forall x \in C_i, \quad f'_i(x) \geq f'_{-i}(x)$. For any classifier, we have

$$\begin{aligned}
&\int_{SR'_1(h)} f'_1(x)dx + \int_{\mathbb{R}^d \setminus SR'_1(h)} f'_{-1}(x)dx \\
&\quad = \int_{SR'_1(h) \bigcap C_1} f'_1(x)dx + \int_{SR'_1(h) \bigcap C_{-1}} f'_1(x)dx \\
&\quad + \int_{\mathbb{R}^d \setminus SR'_1(h) \bigcap C_1} f'_{-1}(x)dx + \int_{\mathbb{R}^d \setminus SR'_1(h) \bigcap C_{-1}} f'_{-1}(x)dx \\
&\quad \leq \int_{SR'_1(h) \bigcap C_1} f'_1(x)dx + \int_{SR'_1(h) \bigcap C_{-1}} f'_{-1}(x)dx \\
&\quad + \int_{\mathbb{R}^d \setminus SR'_1(h) \bigcap C_1} f'_1(x)dx + \int_{\mathbb{R}^d \setminus SR'_1(h) \bigcap C_{-1}} f'_{-1}(x)dx \\
&\quad = \int_{C_1} f'_1(x)dx + \int_{C_{-1}} f'_{-1}(x)dx
\end{aligned} \tag{33}$$

Finally, note that for the classifier $h^\star$, the RHS and LHS of the above inequality are equal. Therefore, $h^\star$ is an optimal ϵ robust classifier and the optimal robust classification error would clearly be $\Phi\left(\frac{-\|\mu_1-\mu_{-1}\|_2+2\epsilon}{2\sigma}\right)$.

□

Remark 3. *(Ilyas et al., 2019) has studied the problem of the optimal robust classifier on a binary classification task where data is driven from Gaussian distributions. However, they have a strong assumption on the hypothesis space. Specifically, they focused on linear classifiers. Here, we do not assume anything about the hypothesis space other than continuity of the decision boundary.*

Fact 1. *Assume that $\theta^{(1)}$ and $\theta^{(-1)}$ are different in $k \leq d$ dimensions. We call k the "effective dimensionality" and we get $k = \|\theta^{(1)} - \theta^{(-1)}\|_0$. We can restrict all x, $\theta^{(1)}$, and $\theta^{(-1)}$ only to the k effective dimensions. We call x', $\theta'^{(1)}$, and $\theta'^{(-1)}$ the restriction of these vectors to the "effective dimensions". Similarly, we call x'', $\theta''^{(1)}$, and $\theta''^{(-1)}$ the restriction of these vectors to "non-effective dimensions".*
Note that $\theta'^{(1)} = -\theta'^{(-1)}$ and $\theta''^{(1)} = \theta''^{(-1)}$, and

$$\|x' - \theta'^{(1)}\|_0 + \|x' - \theta'^{(-1)}\|_0 = k. \tag{34}$$

Theorem 3. *Let's assume that $(x, y) \sim$ Bernoulli $(\theta^{(1)}, \theta^{(-1)}, t)$. Then, 1. the optimal Bayes' classifier $h^\star : \{-1, 1\}^d \to \{-1, 1\}$ on this distribution is of the form:*

$$h^\star(x) = \begin{cases} 1, & \|x - \theta^{(1)}\|_0 \leq \|x - \theta^{(-1)}\|_0 \\ -1, & \textit{otherwise}, \end{cases} \tag{35}$$

and 2. if k is even, the optimal classification error rate is:

$$\sum_{i=0}^{k/2} \binom{k}{i} (1/2 + t)^i (1/2 - t)^{k-i} - 1/2 \binom{k}{k/2} (1/2 + t)^{k/2} (1/2 - t)^{k/2} \tag{36}$$

Otherwise, if k is odd, the classification optimal error rate would be:

$$\sum_{i=0}^{\lfloor k/2 \rfloor} \binom{k}{i} (1/2+t)^{i}(1/2-t)^{k-i}, \tag{37}$$

where $k := \|\theta^{(1)} - \theta^{(-1)}\|_0$.

Proof. 1. We start with the equation of the Baye's optimal classifier as an optimal classifier on the given distribution:

$$h^\star(x) = \underset{y\in\{-1,1\}}{\arg\max}\ D(x,y) \tag{38}$$

$$= \underset{y\in\{-1,1\}}{\arg\max}\ 1/2 \times (1/2+t)^{k-\|x-\theta^{(y)}\|_0}(1/2-t)^{\|x-\theta^{(y)}\|_0}$$

$$= \underset{y\in\{-1,1\}}{\arg\max}\ (1/2+t)^{k-\|x-\theta^{(y)}\|_0}(1/2-t)^{\|x-\theta^{(y)}\|_0}$$

$$= \underset{y\in\{-1,1\}}{\arg\max}\ (1/2+t)^{k-\|x'-\theta'^{(y)}\|_0}(1/2-t)^{\|x'-\theta'^{(y)}\|_0}$$

That is:

$$h^\star(x) = \begin{cases} 1, & (1/2+t)^{k-\|x'-\theta'^{(1)}\|_0}(1/2-t)^{\|x'-\theta'^{(1)}\|_0} \geq \\ & \qquad (1/2+t)^{k-\|x'-\theta'^{(-1)}\|_0}(1/2-t)^{\|x'-\theta'^{(-1)}\|_0} \\ -1, & \text{otherwise} \end{cases} \tag{39}$$

By using the Eq. 34 in the above classification rule:

$$h^\star(x) = \begin{cases} 1, & (1/2+t)^{k-\|x'-\theta'^{(1)}\|_0}(1/2-t)^{\|x'-\theta'^{(1)}\|_0} \geq \\ & \qquad (1/2+t)^{\|x'-\theta'^{(1)}\|_0}(1/2-t)^{k-\|x'-\theta'^{(1)}\|_0} \\ -1, & \text{otherwise} \end{cases} \tag{40}$$

By rearranging this rule, we finally arrive at:

$$h^\star(x) = \begin{cases} 1, & \|x'-\theta'^{(1)}\|_0 \leq k/2 \\ -1, & \text{otherwise,} \end{cases} \tag{41}$$

which by applying the Eq. 34 again is the same as:

$$h^\star(x) = \begin{cases} 1, & \|x'-\theta'^{(1)}\|_0 \leq \|x'-\theta'^{(-1)}\|_0 \\ -1, & \text{otherwise} \end{cases} \tag{42}$$

And finally:

$$h^\star(x) = \begin{cases} 1, & \|x-\theta^{(1)}\|_0 \leq \|x-\theta^{(-1)}\|_0 \\ -1, & \text{otherwise} \end{cases} \tag{43}$$

2. The classification error rate for $h^\star$ can now be calculated as follows:

$$R(h^\star, D) = \mathbb{E}_{x,y\,\sim D}\left\{\mathbb{I}(h^\star(x) \neq y)\right\} \tag{44}$$

$$= \sum_{x,y} \mathbb{I}(h^\star(x) \neq y) \times D(x,y)$$

$$= \sum_{x:h^\star(x)=1} D(x,-1)\ +\ \sum_{x:h^\star(x)=-1} D(x,1)$$

By plugging in $h^\star$ and $D(x,y)$, we would get:

$$= 1/2 \sum_{x:\|x'-\theta'^{(1)}\|_0 \le k/2} (1/2+t)^{\|x'-\theta'^{(1)}\|_0}(1/2-t)^{k-\|x'-\theta'^{(1)}\|_0}$$

$$+ 1/2 \sum_{x:\|x'-\theta'^{(1)}\|_0 > k/2} (1/2+t)^{k-\|x'-\theta'^{(1)}\|_0}(1/2-t)^{\|x'-\theta'^{(1)}\|_0} \tag{45}$$

If k is even, this would simplify to:

$$1/2\sum_{i=0}^{k/2}\binom{k}{i}(1/2+t)^i(1/2-t)^{k-i} + \ 1/2\sum_{i=k/2+1}^{k}\binom{k}{i}(1/2+t)^{k-i}(1/2-t)^i \tag{46}$$

$$= \sum_{i=0}^{k/2}\binom{k}{i}(1/2+t)^i(1/2-t)^{k-i} - 1/2\binom{k}{k/2}(1/2+t)^{k/2}(1/2-t)^{k/2} \tag{47}$$

Otherwise, if k is odd, the error rate would be:

$$\sum_{i=0}^{\lfloor k/2 \rfloor}\binom{k}{i}(1/2+t)^i(1/2-t)^{k-i} \tag{48}$$

□

Remark 4. *Note that the final equations of the optimal classification error rate for this distribution can be simply rewritten by using the Hypergeometric functions.*

Theorem 4. *Let $(x,y) \sim$ Bernoulli$(\theta^{(1)}, \theta^{(-1)}, t)$. Let $B_0^\epsilon(x)$ be the perturbation set. Then, under the assumption 1, 1. the $(0,\epsilon)$-optimal robust classifier $h^\star : \{-1,1\}^d \to \{-1,1\}$ on this distribution is of the form:*

$$h^{\star(adv)}(x) = \begin{cases} 1, & \|x'-\theta'^{(1)}\|_0 \le s \\ -1, & \textit{otherwise}, \end{cases} \tag{49}$$

where $s \in \{0,1,...,k\}$ is a threshold, and x', and $\theta'^{(1)}$ are the restrictions of x, and $\theta^{(1)}$ to the effective dimensions, defined in theorem 3, respectively, and 2. the classification $(0,\epsilon)$-optimal error rate would be:

$$R_{0,\epsilon}^{(adv)}(h,D) = 1/2\sum_{i=0}^{s+\epsilon}\binom{k}{i}(1/2+t)^i(1/2-t)^{k-i} + 1/2\sum_{i=0}^{k-s+\epsilon-1}\binom{k}{i}(1/2+t)^i(1/2-t)^{k-i} \tag{50}$$

Proof. 1. Let k be an effective dimension according to Fact 1. Let h be a $(0,\epsilon)$-optimal robust classifier. Let $S_1, S_{-1} \subset \{-1,1\}^d$ denote all the points that are assigned to the classes 1 and -1 by h, and have ℓ_0 distance of no more than $\lceil \frac{k}{2} \rceil$ to $\theta'^{(1)}$ and $\theta'^{(-1)}$, respectively. Further, let $x_1^\star := \arg\max_{x \in S_1} \|x-\theta'^{(1)}\|_0$, and $x_{-1}^\star := \arg\max_{x \in S_{-1}} \|x-\theta'^{(-1)}\|_0$. And also, let $d_1 = \|x_1^\star - \theta'^{(1)}\|_0$ and, $d_{-1} = \|x_{-1}^\star - \theta'^{(-1)}\|_0$. We claim that for such an optimal robust classifier h, all points $x \in \{-1,1\}^d$, which $\|x-\theta'^{(1)}\|_0 \le d_1$ can be labelled as 1, and all points $x \in \{-1,1\}^d$, which $\|x-\theta'^{(-1)}\|_0 \le d_{-1}$ can be labelled as -1. To show this, consider a set of points $\bar{S_1} = \{x \in \{-1,1\}^d \mid h(x) = -1, \|x-\theta'^{(1)}\|_0 \le d_1\}$. Then, by changing their labels to $+1$: 1. We will not destroy the robustness of any points in S_1, because they are all labelled as $+1$ by the h. 2. We will not destroy the robustness of the other points in $\{-1,1\}^d - S_1$, because in fact $x_1^\star$ and all points $T_1 = \{x \in \{-1,1\}^d : \|x-\theta'^{(1)}\|_0 = d_1\}$ have the same label $+1$ using the assumption 1. Also, ℓ_0 distance between $\bar{S_1}$ and $\{-1,1\}^d - S_1$ are greater or equal than the ℓ_0 distance between T_1 and $\{-1,1\}^d - S_1$. 3. Because for $\bar{S_1}$ the probability of being $+1$ is greater than being -1, this would yield a classifier that does not have a worse adversarial error rate compared to h. So, overall we can change these points label and similarly the points in $\bar{S}_{-1}$ without loss of accuracy.

Therefore, the $(0, \epsilon)$-optimal robust classifier basically thresholds $\|x' - \theta'^{(1)}\|_0$. Notably, the thresholds can be either 0 or k, which means that all points are labeled to either $+1$ or -1. The minimum adversarial error for $p = 0$ and ϵ is given by:

$$\min_{0 \leq s \leq k} 1/2 \sum_{i=0}^{s+\epsilon} \binom{k}{i} (1/2+t)^i (1/2-t)^{k-i} + 1/2 \sum_{i=0}^{k-s+\epsilon-1} \binom{k}{i} (1/2+t)^i (1/2-t)^{k-i} \tag{51}$$

For specifying the exact threshold, one can evaluate the adversarial error rate using a different value of k (Supplementary Figure in Sec. G). □

Lemma 4. *Let $B_p^\epsilon(x)$ be a perturbation set, and $h : \mathbb{R}^d \to L$ be a classifier. Then, $R_{p,\epsilon}^{(adv)}(h, D) \geq R(h, D)$.*

Proof.

$$R_{p,\epsilon}^{(adv)}(h, D) = \mathbb{E}_{x,y \sim D} \left\{ \mathbb{I}(\exists \delta \in B_p^\epsilon : h(x+\delta) \neq y) \right\} \tag{52}$$

It is obvious that $B_p^0(x) \subseteq B_p^\epsilon(x)$ then:

$$\mathbb{E}_{x,y \sim D} \left\{ \mathbb{I}(\exists \delta \in B_p^\epsilon : h(x+\delta) \neq y) \right\} \geq \mathbb{E}_{x,y \sim D} \left\{ \mathbb{I}(\exists \delta \in B_p^0 : h(x+\delta) \neq y) \right\} \tag{53}$$

$$= \mathbb{E}_{x,y \sim D} \left\{ \mathbb{I}(h(x) \neq y) \right\} = R(h, D) \tag{54}$$

□

Theorem 5. *Let Bernoulli $(\theta^{(1)}, \theta^{(-1)}, t)$ be a Bernoulli model. Let $(x, y) \sim$ Bernoulli$(\theta^{(1)}, \theta^{(-1)}, t)$. Let $B_\infty^\epsilon(x)$ be a perturbation set. Then, 1. the (∞, ϵ)-optimal robust classifier $h^{\star(adv)} : \{-1, 1\}^d \to \{-1, 1\}$ on this distribution for $\epsilon < 1$ is of the form:*

$$h^{\star(adv)}(x) = \begin{cases} 1, & \|T(x) - \theta^{(1)}\|_0 \leq \|T(x) - \theta^{(-1)}\|_0 \\ -1, & \textit{otherwise}, \end{cases} \tag{55}$$

where $T(x) = (sign(x_1), \ldots, sign(x_d))$ and $sign(r)$ is 1 if $r \geq 0$ and is -1 otherwise, and 2. the classification (∞, ϵ)-optimal error rate would be:

$$R_{\infty,\epsilon}^{(adv)}(h, D) = \begin{cases} R(h^\star, D), & \epsilon < 1 \\ 1/2, & \textit{otherwise}, \end{cases} \tag{56}$$

where $h^\star$ is the optimal Baye's classification rule.

Proof. Note that if $\epsilon < 1$, no ℓ_∞ attack could change the sign of any dimension of x. Therefore, applying the function T would recover the original clean input x. Therefore, the adversarial error rate of $h^\star \circ T$ would be the same as the error rate of $h^\star$. But adversarial error rate of any classifier is lower bounded by optimal Bayes error rate. Here, the function $h^\star \circ T$ achieves this lower bound. Hence, this classifier should be (∞, ϵ)-optimally robust. If $\epsilon \geq 1$, we can simply map all points of the distribution to $\overrightarrow{0}$. Therefore even the (∞, ϵ)-optimal robust classifier can not have better than random accuracy, which is $1/2$ here. □

B. Different Settings for the Adversarial Training

For the MNIST dataset, we tested a variety of settings to adversarially train a model on $\epsilon = 0.4$. However, it seems that training a robust network on 0.4 is not feasible when weights are randomly initialized. It is notable that this fact was also observed by (Chen et al., 2017).

1. We test (Glorot & Bengio, 2010) initialization and all other settings are the same as the main text.
2. We test different settings of optimizers, specifically, SGD with the learning rate from $\{0.1, 0.01, 0.001\}$ with and without momentum and Adam (Kingma & Ba, 2014) with the learning rate from $\{0.1, 0.01, 0.001\}$ and all other settings are the same as the main text.
3. Also, we try to change the batch size from $\{16, 32, 64, 128, 256, 512, 10^4\}$ and all other settings are the same as the main text.
4. We test standard ResNet-18 architecture (He et al., 2015b) with different settings of optimizers as mentioned below. We believe that this model can cover needed model capacity as discussed in (Madry et al., 2017).
5. We test different settings of PGD (Madry et al., 2017) by using different learning rates and step sizes and all other settings are the same as the main text. It is notable that by using FGSM, we can successfully train a model on $\epsilon = 0.4$. However, the model indeed suffers from the gradient obfuscation (Athalye et al., 2018).

C. Evaluation Details

When we use the training PGD as mentioned in the main text on the epsilon $= 0.4$ model, the attack leads to about 89% adversarial accuracy. Also, when we use the training PGD as mentioned in the main text on the epsilon = 0.5 model, the attack leads to about 80% adversarial accuracy!

1. FGSM (Goodfellow et al., 2014) is not a stronger attack than PGD (Madry et al., 2017). However, verifying that PGD is always stronger than FGSM on our models is essential. We verified that adversarial accuracy is always lower using training PGD on both models.
2. Unbounded attacks should always lead to zero accuracies. We verify that by using the training PGD but in the unbounded setting, we can make an attack, which leads to zero accuracies on both models.
3. PGD with random offset should not affect the model adversarial accuracy significantly. As the random offset is included in training PGD, attacking the model using random offset does not decrease the adversarial accuracy on models.
4. Using many random restarts should not effect model adversarial accuracy significantly. We used training PGD with 10 random restarts. However, we did not observe any significant changes in the adversarial accuracy of both models.
5. Using PGD with more steps should not affect adversarial accuracy significantly. We used the training PGD with 2000 steps as an attack but it did not affect models' adversarial accuracy significantly.
6. By increasing the magnitude of the attack, adversarial accuracy should monotonically decrease. We verify that by using the training PGD and increasing the allowed magnitude of perturbations, the adversarial accuracy of both models monotonically decreases.
7. The targeted PGD attack should not significantly change the adversarial accuracy. We verify that by using the PGD targeted attack with the same setting as in the training PGD, and considering the highest loss between 10 possible classes targets for each image (Gowal et al., 2019). Still, no significant changes are observed using this new attack on both models.
8. A brute force attack should be an effective attack on the model. We use a batch of size 128 and 10^8 samples of perturbation for each image. Specifically, we use Bernoulli distribution for sampling the perturbations. Each pixel of an image can change either $-\epsilon$ or $+\epsilon$. We verify that this method also is not effective in attacking both models.
9. We also verify that the minimum needed size of distortion for decreasing the adversarial accuracy of models to zero, is higher in both models in comparison to the base model, assuming the training PGD as an attack.

10. We also try to decrease our models' adversarial accuracy by using the black-box attack ZOO (Chen et al., 2017). Specifically, we used the default parameters in the art library (Nicolae et al., 2018). But this attack is not also effective on both models. Here, we note that indeed being sure about the effect of this attack needs more hyper parameters tuning.

11. Black box attacks should not outperform white-box attacks. We also verify our models' adversarial accuracy on black-box attacks is not lower than the reported adversarial accuracy. Specifically, for the black box attack, we used the training PGD on the model robust to the adversarial perturbations of size 0.3.

12. We also propose a new attack which we call "semi-black box". In this attack, we used the base model for training on perturbations of sizes 0.4 and 0.5, as a model for generation of the adversarial examples. We used the same setting as in the black-box attacks. However, again this new attack is also not effective.

D. Supplementary Figures for the Proposed Method

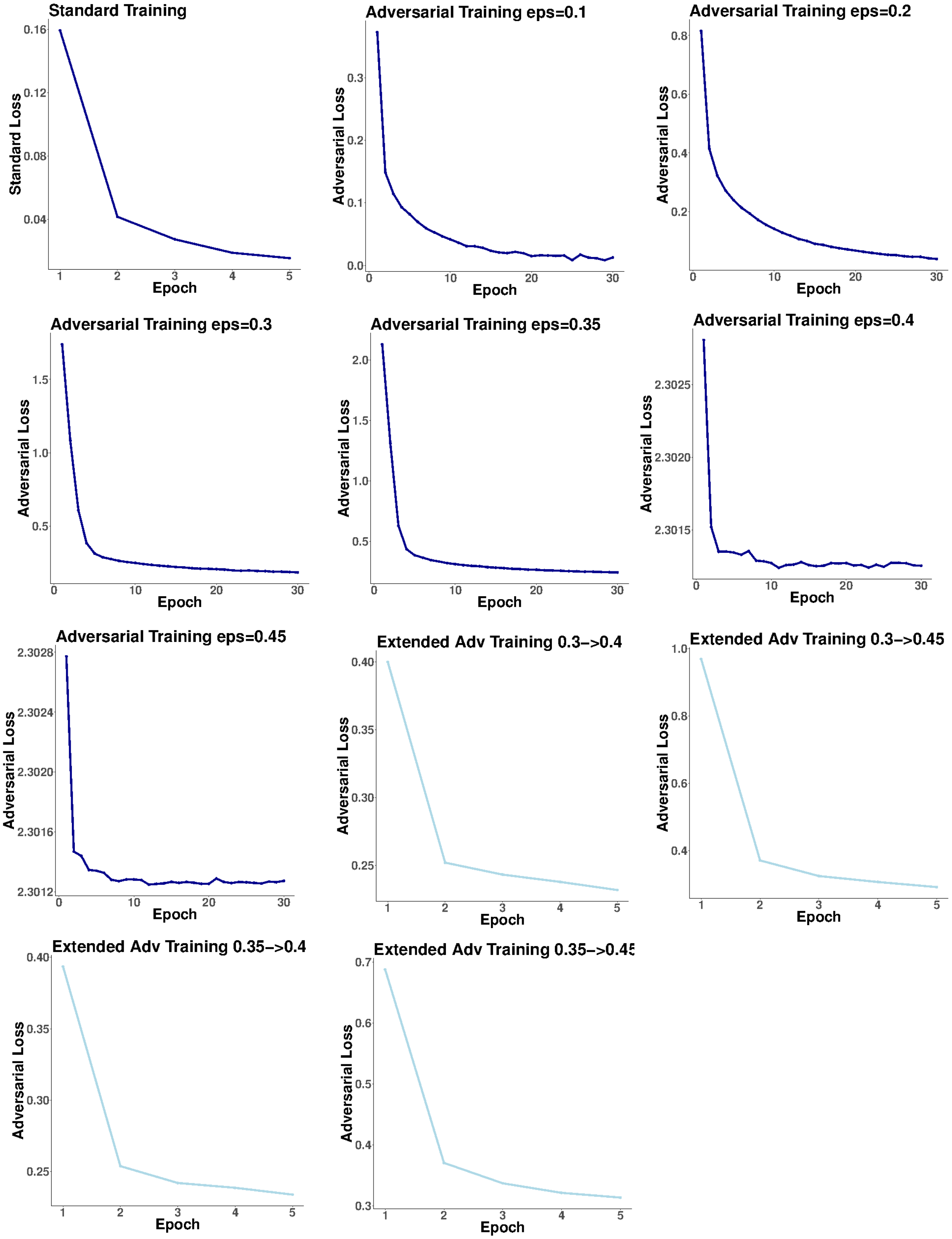


Figure 8. Training adversarial loss against the epoch number for various epsilons in the MNIST dataset. Adversarial training fails to decrease the training adversarial loss on epsilon greater than ≈ 0.35. However, training such models is feasible using extended adversarial training.

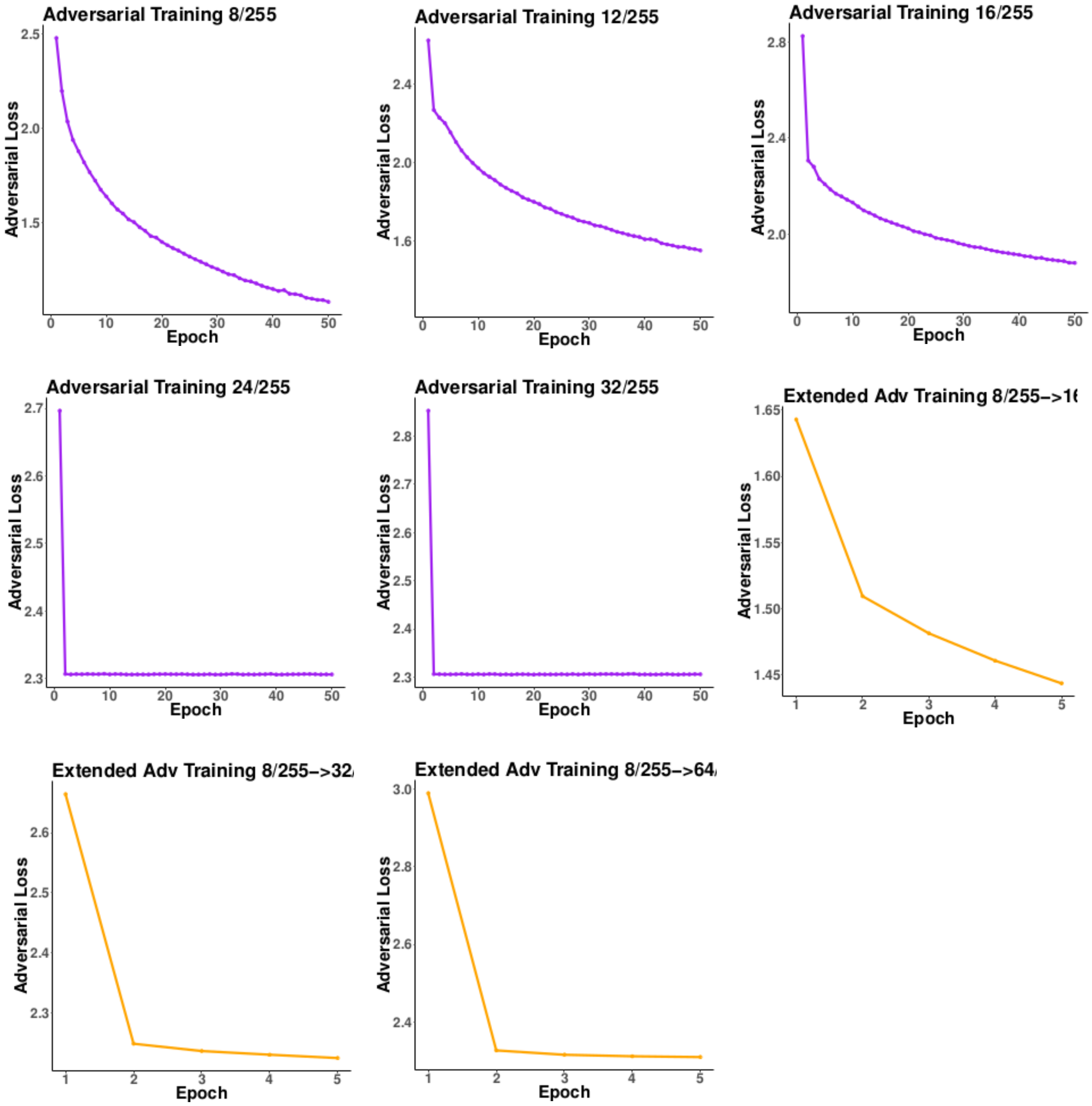


Figure 9. Training adversarial loss against the epoch number for various epsilons in the CIFAR10 dataset. Adversarial training fails to decrease the training adversarial loss on epsilon greater than $\approx \frac{16}{255}$. However, training such models is feasible using extended adversarial training.

E. Omitted Figures for the Convergence of Adversarial Training

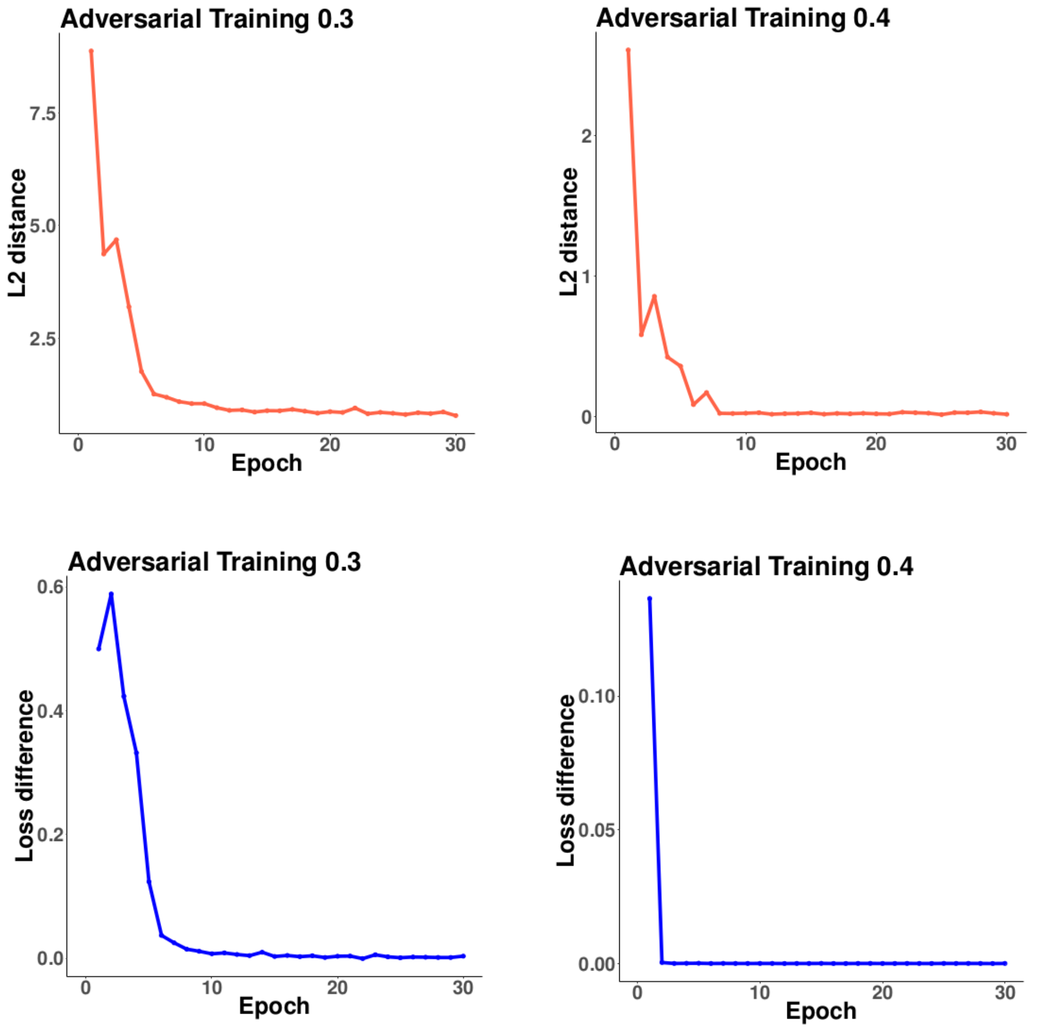


Figure 10. The difference of training adversarial loss and ℓ_2 distance of weights in two consecutive epochs. These figures empirically indicate the convergence of adversarial training.

F. Omitted Figure for the Mode Connectivity

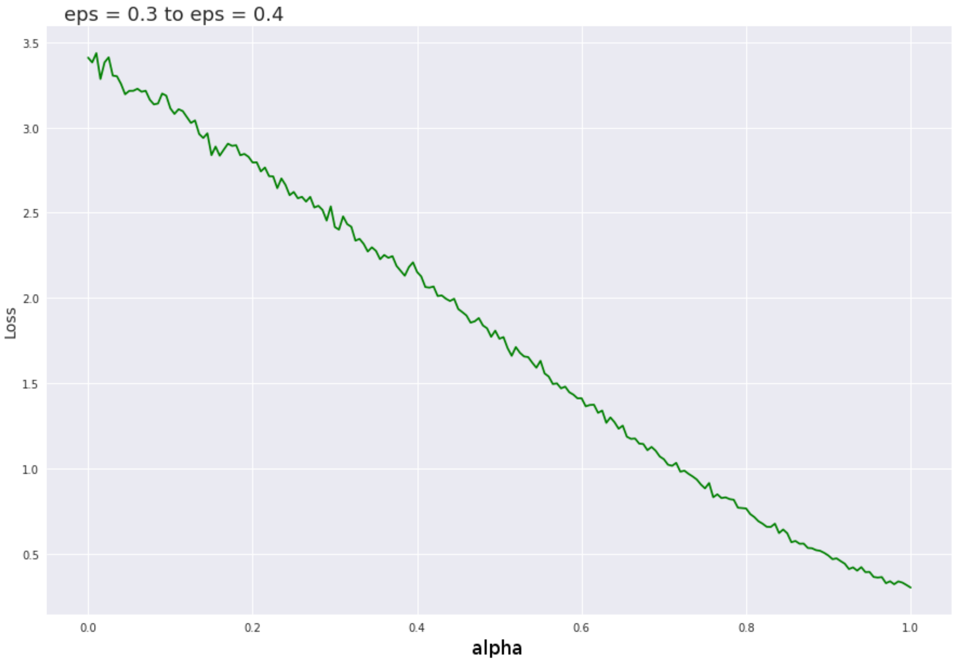


Figure 11. When initialized on weights of an already adversarially trained model with epsilon = 0.3, $\ell_\epsilon(\alpha)$ could be better approximated by a convex function. Here, for computing the adversarial loss we used the PGD with learning rate = 0.005 and 200 steps. This PGD is more accurate than the PGD used for producing a similar figure in the main text. As a result, the function could be better approximated by a convex function.

G. Omitted Figure for the Theory Part

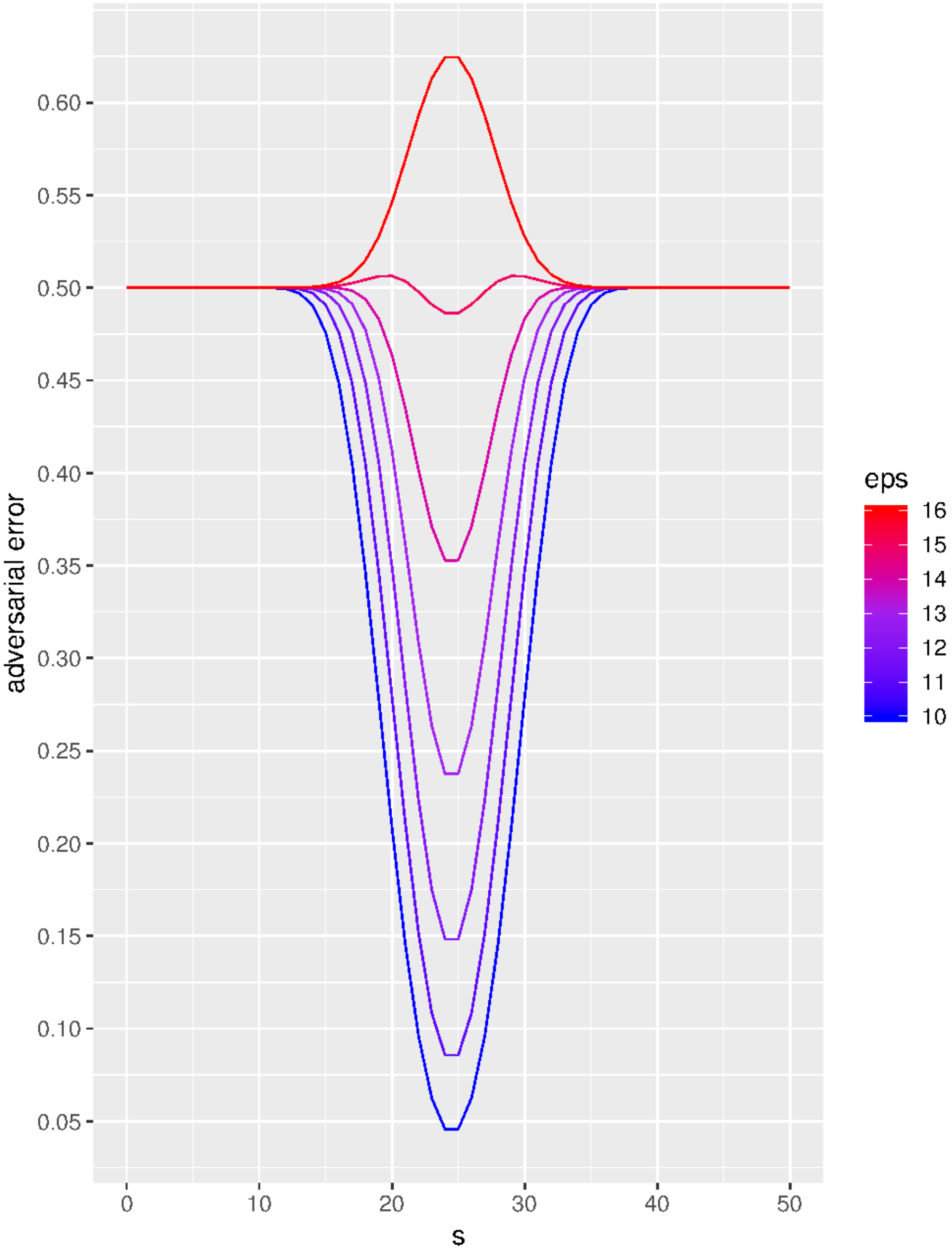
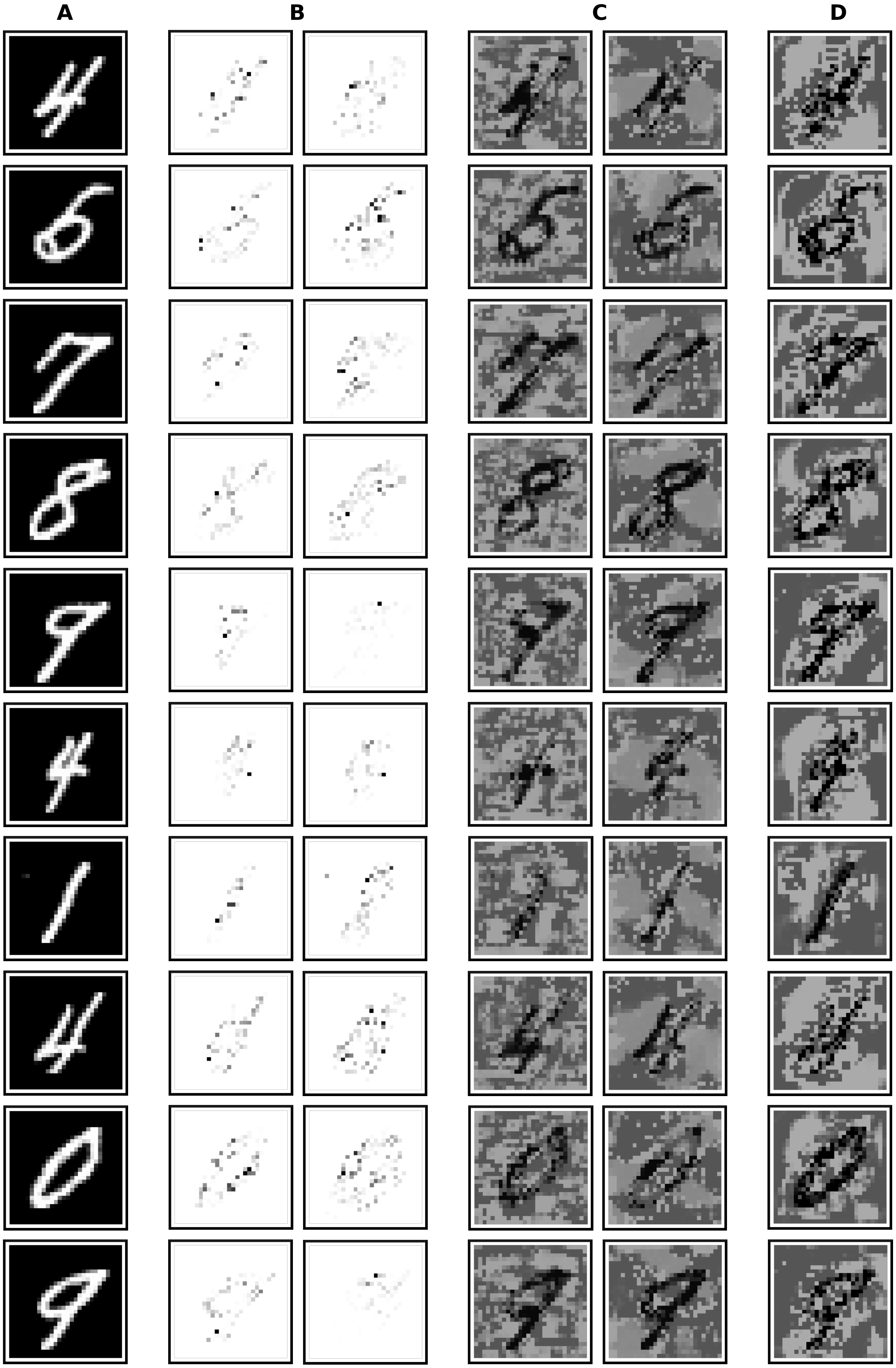


Figure 12. Evaluating the final formula of ℓ_0 on the multi-dimensional Bernoulli distribution against different hyper-parameter values for $k = 50$ and $t = 0.30$.

H. Omitted Figures for Interpretability



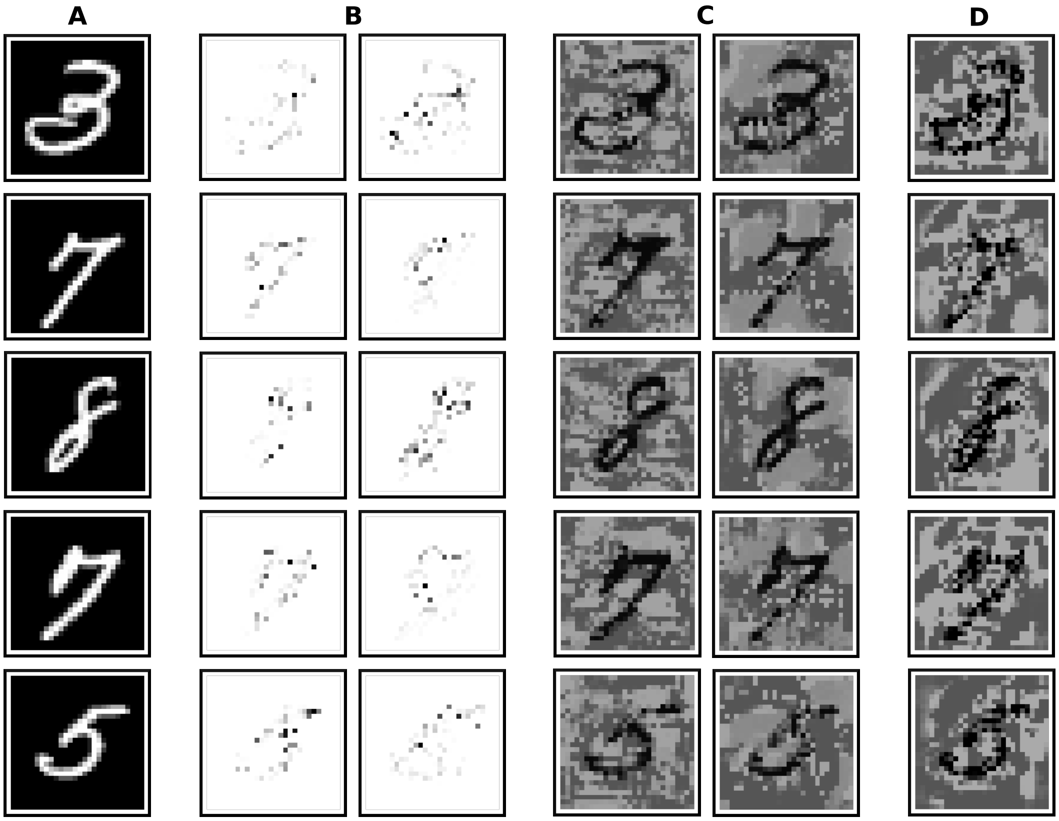


Figure 13. (A): Samples from MNIST. (B) The loss gradient (saliency map) with respect to the perturbed input (of epsilon = 0.3) that is obtained from the model trained on epsilon = 0.3 (Right) and the model that is trained on epsilon from 0.3 to 0.4 (Left). (C) The perturbation (of size epsilon = 0.45) for the corresponding models in (B). (D) Left: the original image. Right: the perturbations that are obtained from the model trained on epsilon from 0.3 to 0.5, based on the attack obtained from the actual gradient that is mentioned in Sec. 3.3. Overall, training on larger perturbations leads to improved interpretability of both the model and the attacks.