Are Adversarial Examples Created Equal? A Learnable Weighted Minimax Risk for Robustness under Non-uniform Attacks

Huimin Zeng[1*], **Chen Zhu**[2*], **Tom Goldstein**[2], **Furong Huang**[2]
[1] Technical University of Munich
[2] University of Maryland, College Park
huimin.zeng@tum.de, {chenzhu,tomg,furongh}@cs.umd.edu

Abstract

Adversarial Training is proved to be an efficient method to defend against adversarial examples, being one of the few defenses that withstand strong attacks. However, traditional defense mechanisms assume a uniform attack over the examples according to the underlying data distribution, which is apparently unrealistic as the attacker could choose to focus on more vulnerable examples. We present a weighted minimax risk optimization that defends against non-uniform attacks, achieving robustness against adversarial examples under perturbed test data distributions. Our modified risk considers importance weights of different adversarial examples and focuses adaptively on harder examples that are wrongly classified or at higher risk of being classified incorrectly. The designed risk allows the training process to learn a strong defense through optimizing the importance weights. The experiments show that our model significantly improves state-of-the-art adversarial accuracy under non-uniform attacks without a significant drop under uniform attacks.

Introduction

It is widely known that deep neural networks could be vulnerable to adversarially perturbed input examples (Szegedy et al. 2013; Huang et al. 2017). Having strong defenses against such attacks is of value, especially in high-stakes applications such as autonomous driving and financial credit/risk analysis. Adversarial defenses aim to learn a classifier that performs well on both the "clean" input examples (accuracy) and the adversarial examples (robustness) (Zhang et al. 2019b). Despite a large literature on studying adversarial defenses in machine learning, computer vision, natural language processing and more, one of the few defenses against adversarial attacks that withstands strong attacks is *adversarial training* (Carlini and Wagner 2017; Kannan, Kurakin, and Goodfellow 2018; Kurakin, Goodfellow, and Bengio 2016; Shaham, Yamada, and Negahban 2018). In adversarial training, adversarial examples generated via a chosen attack algorithm are included in the training on the fly. As is shown in many works (Carlini and Wagner 2017; Kannan, Kurakin, and Goodfellow 2018; Kurakin, Goodfellow, and Bengio 2016; Shafahi et al. 2019b; Shaham, Yamada, and Negahban 2018; Zhang et al. 2019a,b), adversarial training has demonstrated great success in the attack-defense game.

A major issue with adversarial training is that it seeks a model that is robust to adversarial perturbations on the training set. Adversarial training attempts to solve a robust optimization problem against a point-wise adversary that independently perturbs each example(Staib and Jegelka 2017). The traditional optimization objective is usually (unweighted) average of robust losses over all training data points; the robust loss for each training data point is evaluated on adversarial example that is independently generated for each training data point

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_n}\Big[\mathsf{robust\ loss}(f,\boldsymbol{x},y,\epsilon)\Big] \tag{1}$$

where $\mathcal{D}_n$ is the empirical distribution and the $\mathsf{robust\ loss}(f,\boldsymbol{x},y,\epsilon)$ could be any loss function that characterizes the risk of mis-classification of adversarial examples under the threat model of bounded ϵ perturbation on the input $(\boldsymbol{x},y)$ to f (For instance, the 0-1 robust loss is $\mathbf{1}\{\exists\|\boldsymbol{\delta}\|\leq\epsilon,\text{s.t. } f(\boldsymbol{x}+\boldsymbol{\delta})y\leq 0\}$).

This robust error in Equation (1) treats the adversarial examples generated around different training data points as equally important when optimizing the training objective. In other words, the training objective assumes that an attacker chooses to attack the input examples uniformly, regardless of how close these examples are to the decision boundary. As a result, the above robust error would fail to measure security against an attacker who focuses on the more vulnerable examples. As shown in Figure 1, the data points that are closer to decision boundary, are more vulnerable to attacks, since the attacker needs a relatively smaller perturbation to move them to wrong side of the decision boundary. Therefore, we aim to design robust neural networks against *non-uniform attacks*.

Our methodology Motivated by the idea that not all adversarial examples are equally important, we propose a novel weighted minimax risk for adversarial training that achieves both robustness against adversarial examples and accuracy for clean data examples. Our modified risk considers importance weights of different adversarial examples and adaptively focuses on vulnerable examples that are wrongly

*Equal Contributions

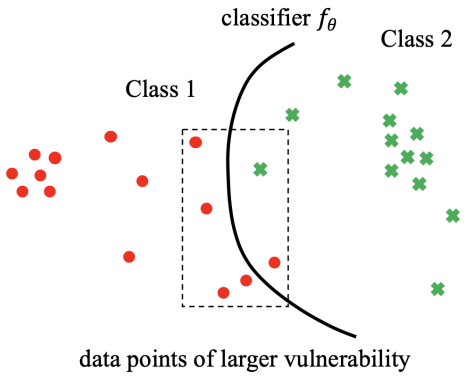

Figure 1: Vulnerability and Distance to Decision Boundary

classified or at high risk of being classified incorrectly. The designed weighted risk allows the training process to learn the distribution of the adversarial examples conditioned on a neural network model θ through optimization of the importance weights and learn to defend against strong non-uniform attacks.

Summary of contributions:

1. We introduce a novel distribution-aware training objective by integrating a re-weighting mechanism to the traditional minimax risk of adversarial training framework.
2. Based on the distribution-aware minimax risk, we are able to generate stronger adversarial examples, such that some state-of-the-art adversarial training algorithms (for instance, TRADES (Zhang et al. 2019b)) will perform poorly. On CIFAR10, the robust accuracy of the network (ResNet18 (He et al. 2016)) trained with standard adversarial training setting drops from 53.38% to 19.78% under our proposed attacks.
3. Thirdly, we propose a strong defense mechanism based on our re-weighting strategy, consistently increasing the robustness of models against strong non-uniform (distribution-aware) attacks. Our method improves the state-of-the-art robust accuracy from 19.78% to 23.62% on CIFAR10.
4. Besides, our defense mechanism matches the state-of-the-art under traditional evaluation metrics (uniform attacks). On CIFAR10, the network trained with our modified risk is able to achieve 54.10%, in comparison to the baseline of 53.38%.
5. Finally, we propose two new metrics to evaluate the robustness of the trained classifier under vulnerability- and distribution-aware attacks.

Related Work

A number of defense mechanisms have been proposed to maintain accuracy for adversarial images. This includes detecting and rejecting adversarial examples (Ma et al. 2018; Meng and Chen 2017; Xu, Evans, and Qi 2017), along with other works such as label smoothing and logit squeezing (Mosbach et al. 2018; Shafahi et al. 2019a; Mosbach et al. 2018), gradient regularization (Elsayed et al. 2018; Finlay and Oberman 2019; Ross and Doshi-Velez 2018), local linearity regularization (Qin et al. 2019), and a Jacobian regularization (Jakubovitz and Giryes 2018). Adversarial training proposed by Madry et al. (2017) is among the few that are resistant to attacks by Athalye, Carlini, and Wagner (2018), which broke a suite of defenses. Adversarial training defends against test time adversarial examples by augmenting each minibatch of training data with adversarial examples during training.

Adversarial training is powerful in terms of defending against adversarial examples. We witnessed a surge of studies on designing loss functions for training robust classifiers. Many methods in the adversarial training literature treat all training examples equally without using sample-level information. Recently, however, Balaji, Goldstein, and Hoffman (2019) propose example-specific perturbation radius around every training example to combat the adversarial training's failure to generalize well to unperturbed test set. Moreover, Zhang et al. (2019b) provides a theoretical characterization of the trade-off between the natural accuracy and robust accuracy by investigating the Bayes decision boundary and introducing a new regularization based on the KL divergence of adversarial logit pairs, with which the trained model reaches state-of-the-art performance.

Distributionally robust optimization (DRO) is a tool that links generalization and robustness (Staib and Jegelka 2017; Ben-Tal et al. 2013; Blanchet et al. 2017; Delage and Ye 2010; Duchi, Glynn, and Namkoong 2016; Gao and Kleywegt 2016; Goh and Sim 2010). DRO seeks a model that performs well under adversarial joint perturbations of the entire training set. The adversary is not limited to moving points individually, but can move the entire distribution within an ϵ-ball of $\mathcal{D}_n$ for some notion of distance between distributions. The attacker has a specific attack budget to attack the distribution of the dataset; the perturbed distribution has to be ϵ-close to the uniform distribution. However in the non-uniform attack setting we consider, although the attacker might have constrained power to alter each image, their attack to the distribution might be unconstrained.

Weighted Minimax Risk Models

Rethinking Adversarial Training

Traditional training Traditional model training is the process of learning optimal model parameter $\boldsymbol{\theta}$ that characterizes a mapping from input space to output space $f_{\boldsymbol{\theta}} : \boldsymbol{\mathcal{X}} \rightarrow \mathcal{Y}$. The model is designed to minimize the expectation of the *natural loss function* $l(f_{\boldsymbol{\theta}}(\boldsymbol{x_i}), y_i)$ under the unknown underlying distribution of input examples $(\boldsymbol{x}_i, y_i) \sim \mathcal{D}$

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x}_i, y_i)\sim\mathcal{D}}\Big[l(f_{\boldsymbol{\theta}}(\boldsymbol{x_i}), y_i)\Big] \tag{2}$$

In practice, an assumption of input examples $(\boldsymbol{x}_i, y_i)_{i=1}^N$ being i.i.d. is often made, allowing unbiased empirical estimation of the expectation of the *natural* loss.

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N} l(f_{\boldsymbol{\theta}}(\boldsymbol{x_i}), y_i) \tag{3}$$

Performing full-batch gradient descent is too computationally expensive. Therefore, the models are usually trained by means of mini-batch gradient descent with batch size m. This is important, since this is statistically equivalent to full-batch gradient descent, but with larger variance, which is related to batch size.

$$\min_{\boldsymbol{\theta}} \frac{1}{m} \sum_{i=1}^{m} l(f_{\boldsymbol{\theta}}(\boldsymbol{x_i}), y_i) \tag{4}$$

Adversarial training Adversarial training (Madry et al. 2017) has been one of the most prevalent approaches to combat evasion attacks. Specifically, adversarial training solves a mini-max problem by alternating between a network parameter update and an update on input perturbations using projected stochastic gradient descent, seeking a convergence to an equilibrium. The optimizer, originally designed to minimize the natural loss on clean data examples, now takes additional adversarial examples generated during training into consideration. On each step, an inner loop generates the strongest perturbation $\boldsymbol{\delta}_i$ within the ϵ radius of each input example $\boldsymbol{x}_i$ (a specific norm bounded by ϵ (Szegedy et al. 2013)) using projected gradient descent (PGD), and then minimizes the *adversarial loss function* $l(f_{\boldsymbol{\theta}}(\boldsymbol{x_i} + \boldsymbol{\delta_i}), y_i)$ in expectation according to distribution $\mathcal{D}$

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x}_i, y_i) \sim \mathcal{D}} \left[\max_{\boldsymbol{\delta_i} : \|\boldsymbol{\delta_i}\| < \epsilon} l(f_{\boldsymbol{\theta}}(\boldsymbol{x_i} + \boldsymbol{\delta_i}), y_i). \right] \tag{5}$$

Therefore, during each update of the network parameters, adversarial examples (perturbations of the input examples) are generated through PGD search of a perturbation direction that maximizes the loss function, and are added to the input examples for next update of the network parameters. The idea behind adversarial training is that these adversarially generated perturbations, added to the training data, will force the model to proactively adjust the model parameters during training to combat potential adversarial perturbations at test time.

Corresponding to Equation (4), where the optimization objective is constructed over mini-batches, the assumption of adversarial examples being i.i.d. is still made for unbiased empirical estimation of the expectation of the *adversarial* loss

$$\min_{\boldsymbol{\theta}} \frac{1}{m} \sum_{i=1}^{m} \max_{\boldsymbol{\delta_i} : \|\boldsymbol{\delta_i}\| < \epsilon} l(f_{\boldsymbol{\theta}}(\boldsymbol{x_i} + \boldsymbol{\delta_i}), y_i) \tag{6}$$

Are adversarial examples created equal? In traditional training, it is reasonable to use the non-weighted sum of the loss evaluated at each data point as an unbiased estimation of the expectation of the natural loss. However in adversarial training, one often ignored issue is that the loss we optimize is no longer the natural loss on clean data. The goal of adversarial training is to combat adversarial examples at test time. Robustness is achieved by generating a strong (if not the strongest) adversarial perturbation $\boldsymbol{\delta}_i$ for each training data point $(\boldsymbol{x}_i, y_i)$. However it is unclear whether we should treat the generated adversarial examples $\{\boldsymbol{\delta}_i\}_{i=1}^N$ equally. In particular, the loss function in Equation (6) suffers from two problems.

problem (a): It puts equal weights on adversarial examples closer to the decision boundary and examples far away;

problem (b): It assumes that a white-box attacker will always perform a uniform attack on all data points, but, in practice, it might attack the distribution of the adversarial example as well.

In the following section, we will introduce a modified adversarial loss, called *weighted minimax risk*, where the weights are learnable via a training process. We focus on ℓ_∞ norm bounded perturbations although the mechanism could be extended to other norms.

Re-weighting of Vulnerability and Robustness

In the previous section, **problem (a)** points to a potential problem with Equation (6) — all adversarial examples generated during adversarial training, despite their varying distances to the decision boundary and thus varying risk of being misclassified, are treated equally when empirically estimating the expectation of the adversarial loss.

Problem (b), on the other hand, reveals another unsatisfactory design of Equation (6) — due to the adversarial nature of evasion attacks, the test time adversarial examples $\boldsymbol{x}'_{\text{test}}$ do not necessarily have the same distribution as the training time adversarial examples $\boldsymbol{x}'_{\text{training}}$ generated in adversarial training. It is highly likely that the distribution of the adversarial risk is not equal to the independent identical distribution of clean data points.

In this subsection, we first define the "confidence margin" as a measurement of vulnerability of examples in the probability space. Positive margin indicates a correctly classified example and negative margin an incorrectly classified one.

Definition 1 (margin of a classifier f on example $(\boldsymbol{x}_i, y_i)$ (Zhang and Liang 2019)). *For a data point $(\boldsymbol{x}_i, y_i)$, the margin is the difference between the classifier's confidence in the correct label y_i and the maximal probability of an incorrect label t,* $\mathsf{margin}(f, \boldsymbol{x}_i, y_i) = p(f(\boldsymbol{x}_i) = y_i) - \max_{t \neq y_i} p(f(\boldsymbol{x}_i) = t)$.

Remark In the context of white-box attack, this margin is unfortunately accessible to the adversarial attackers. This is the key prerequisite for an adversarial attacker to perform non-uniform attack (more details in later sections).

Although it is impossible to know the distribution of test time adversarial examples, we could follow a principle to reduce the vulnerability of our model by focusing on vulnerable examples. In particular, we aim to design an importance weight c_i based on the margin of $\boldsymbol{x}'_{\text{training}}$. If the margin of the generated adversarial example during training $\boldsymbol{x}'_{\text{training}}$ is large, the adversarial example $\boldsymbol{x}'_{\text{training}}$ is a weak attack (a positive margin indicates the attack failed), and thus its importance weight c_i should be smaller. A more detailed description follows below.

1. if margin is **positive** and large (the adversarial $\boldsymbol{x}'_{\text{training}}$ is **correctly** classified and rather robust), the importance weight c_i should be **small**;
2. If margin is **positive** but **small** (the adversarial $\boldsymbol{x}'_{\text{training}}$ is **correctly** classified but **vulnerable**), the importance

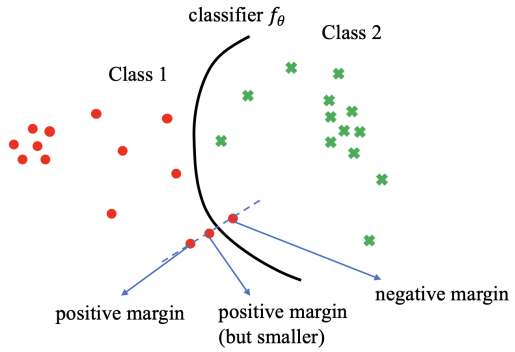

Figure 2: Margin and Vulnerability

weight c_i should be **moderate**;

3. if margin is **negative** (the adversarial $\boldsymbol{x}'_{\text{training}}$ is **incorrectly** classified), the importance weight c_i should be **large**.

Figure 2 shows the relation between margin and vulnerability of certain data points. It is straightforward to design a loss function, so that the focus of training is on the examples which are easier to be attacked (corresponding to small positive margin) or are already successfully attacked (corresponding to negative margin). Now, we formally propose Adaptive Margin-aware Risk.

Adaptive Margin-aware Risk *Adaptive margin-aware minimax risk* is a minimax optimization objective, using an exponential family parameterized by the margin of the adversarial examples in training.

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{m} \max_{\boldsymbol{\delta_i}:\|\boldsymbol{\delta_i}\|<\epsilon} e^{-\alpha \text{ margin}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i+\boldsymbol{\delta}_i),y_i)} l(f_{\boldsymbol{\theta}}(\boldsymbol{x_i}+\boldsymbol{\delta_i}),y_i) \quad (7)$$

where $\alpha > 0$ is a positive hyperparameter of this exponential weight kernel. With the intuition, we can see that there is a positive correlation between the exponential weight kernel and individual loss l. Larger individual loss will induce a larger weight, and vice versa.

Comparison with "natural and adversarial loss combined" Previous works (Goodfellow, Shlens, and Szegedy 2014; Kurakin, Goodfellow, and Bengio 2016) consider a loss that combines both the natural loss and adversarial loss with a hyperparameter λ, i.e.,

$$\min_{\boldsymbol{\theta}} \underbrace{\sum_{i=1}^{m} l(f_{\boldsymbol{\theta}}(\boldsymbol{x_i}),y_i)}_{\text{natural loss}} + \lambda \underbrace{\sum_{i=1}^{m} \max_{\boldsymbol{\delta_i}:\|\boldsymbol{\delta_i}\|<\epsilon} l(f_{\boldsymbol{\theta}}(\boldsymbol{x_i}+\boldsymbol{\delta_i}),y_i)}_{\text{adversarial loss}}. \quad (8)$$

This approach could be thought of as a limiting case of our proposed margin kernel with small α, and it doesn't account for weighting adversarial examples with varying amplitudes.

As we see, Equation (8) designs a defense mechanism that treats adversarial examples $\boldsymbol{x}'_{\text{training}}$ equally and would fail if the attacker at test time chooses to attack the more vulnerable examples (closer to the decision boundary). This is a key difference compared to natural training when unseen examples are assumed to be from the same distribution as the training examples.

Distributionally Robust Adversrial Training

Attack Distribution of Adversarial Examples

The distribution of examples that the adversary deploys to attack, i.e., the attack distribution of adversarial examples may deviate from the empirical distribution $\mathcal{D}_n$ represented by the training examples. In the context of adversarial training, the objective function we use to achieving robustness against an "attack distribution-aware" adversary should be

$$\mathcal{L}'(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x'},y)\sim\mathcal{D}'}\Big[l(f_{\boldsymbol{\theta}}(\boldsymbol{x'}),y)\Big], \quad (9)$$

where $\mathcal{D}'$ denotes the unknown underlying distribution of the adversarial examples.

In a standard adversarial training framework, as reviewed in the third section, the learner generates the perturbation $\boldsymbol{\delta}_i^*$ (using PGD) and thus an adversarial example $\boldsymbol{x}'_i = \boldsymbol{x}_i + \boldsymbol{\delta}_i^*$ for each input example $\boldsymbol{x}_i$ to minimize the *adversarial loss*. The training objective used in practice is

$$\widehat{\mathcal{L}}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} l(f_{\boldsymbol{\theta}}(\boldsymbol{x_i}+\boldsymbol{\delta}_i^*),y_i) = \frac{1}{m}\sum_{i=1}^{m} l(f_{\boldsymbol{\theta}}(\boldsymbol{x}'_i),y_i) \quad (10)$$

The **problem** is that the objective $\widehat{\mathcal{L}}(\boldsymbol{\theta})$ (Equation (10)) used in standard adversarial training is often **not** an unbiased estimator of the true objective function $\mathcal{L}'(\boldsymbol{\theta})$ (Equation (9)) required, since the generated adversarial examples during adversarial training $(\boldsymbol{x}_i{}', y_i)$ are not necessarily good representation of the underlying distribution of the adversarial examples. This is exactly the challenge of achieving robust models; the adversarial attacks are unpredictable.

The true objective $\mathcal{L}'(\boldsymbol{\theta})$ illustrated in Equation (9) is unfortunately often intractable, since the underlying distribution of the adversarial examples is unknown. The problem reduces to an unbiased estimation of the unknown distribution of the adversarial examples.

Comparison with distributionally robust optimization In distributionally robust optimization (DRO) literature as surveyed in the related work section, the methods developed often assume that the divergence between the empirical distribution and the attack distribution is bounded by a threshold $\text{divergence}(\mathcal{D}_n, \mathcal{D}') \leq \rho$. Thus, the DRO (Namkoong and Duchi 2016) objective is

$$\min_{\theta} \max_{\text{divergence}(\mathcal{D}_n,\mathcal{D}')\leq\rho} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}'}[l(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y)] \quad (11)$$

Apart from the complexity of solving the inner constrained maximization problem, DRO requires evaluating the loss for every example in the entire training set before every minimization step, which can be expensive for practical models and datasets.

As illustrated previously, we introduce a risk estimator for each data point individually, so that the objective function is able to express the distribution of the adversarial examples (allowing a non-uniform attack) and learn it via training. It only requires evaluating an importance weight at each sample in the minibatch, but is able to improve distributional robustness against adversarial examples.

Definition 2 (Importance Weights). *For training data points $(\boldsymbol{x}_i, y_i)_{i=1}^N$ and their corresponding adversarial perturbations $(\boldsymbol{x}_i', y_i)_{i=1}^N$, we define the importance weight $s(f_{\boldsymbol{\theta}}, \boldsymbol{x}_i', y_i)$ between $(\boldsymbol{x}_i', y_i)$ and $(\boldsymbol{x}_i, y_i)$, i.e., the ratio of the adversarial example distribution and the clean data distribution evaluated at training data point $(\boldsymbol{x}_i, y_i)$, as*

$$s(f_{\boldsymbol{\theta}}, \boldsymbol{x}_i', y_i) := \frac{\mathcal{D}'(\boldsymbol{x}_i', y_i)}{\mathcal{D}(\boldsymbol{x}_i, y_i)}. \tag{12}$$

Remark In our adaptive margin-aware risk, the importance weight is parameterized as the learnable scaling factor $s(f_{\boldsymbol{\theta}}, \boldsymbol{x}_i', y_i) = e^{-\alpha \text{ margin}(f_{\boldsymbol{\theta}}, \boldsymbol{x}_i + \boldsymbol{\delta}_i, y_i)}$ as shown in Equation (7).

Therefore, our re-weighting strategy – adaptive margin-aware risk– proposes to train the objective function as follows (if we consider full-batch gradient descent)

$$\widetilde{\mathcal{L}}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} s(f_{\boldsymbol{\theta}}, \boldsymbol{x}_i', y_i) l(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i'), y_i) \tag{13}$$

$$\approx \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\Big[s(f_{\boldsymbol{\theta}}, \boldsymbol{x}', y) l(f_{\boldsymbol{\theta}}(\boldsymbol{x}'), y)\Big] \tag{14}$$

$$\approx \mathbb{E}_{(\boldsymbol{x}',y)\sim\mathcal{D}'}\Big[l(f_{\boldsymbol{\theta}}(\boldsymbol{x}'), y)\Big]. \tag{15}$$

Since the importance weight scaling factors $s(f_{\boldsymbol{\theta}}, \boldsymbol{x}', y)$ is learnable, our objective can be thought of as "learning" the *adversarial example distribution conditioned on a neural network model* $\boldsymbol{\theta}$ via learning of the importance weight $s(f_{\boldsymbol{\theta}}, \boldsymbol{x}', y)$ using the objective in Equation (13).

Based on the previous analysis of computational feasibility in the third section, it is impractical to perform the full batch optimization regarding such problem. However, we verify that minimizing adaptive margin-aware risk in mini-batches is statistically equivalent to a full-batch version.

$$\begin{aligned}&\mathbb{E}_{(\boldsymbol{x}_i,y_i)\sim\mathcal{D}}[\widetilde{L}_m(\boldsymbol{\theta})]\\ &= \mathbb{E}_{(\boldsymbol{x}_i,y_i)\sim\mathcal{D}}[\frac{1}{m}\sum_{i=1}^{m} s(f_{\boldsymbol{\theta}}, \boldsymbol{x}_i', y_i) l(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i'), \boldsymbol{y}_i)] \\ &= \widetilde{L}(\boldsymbol{\theta})\end{aligned} \tag{16}$$

Defending against Vulnerability- and Distribution-aware Attacks

As we have argued before, a "smarter" white-box attacker could have access to the vulnerability of different adversarial examples, and therefore could focus on more vulnerable examples. More important, the attacker is able to **sample** the more vulnerable data points more frequently and craft adversarial perturbations to these sampled examples. In our work, the vulnerability is measured by the margin-aware weights. If the vulnerability of a data point is larger, then its margin-aware weight is larger, and it will be sampled by the attacker with higher probability.

To develop an efficient defense mechanism against non-uniform attacks, we augment the adversarial training framework using our proposed adaptive margin-aware risk, as shown in Algorithm 1.

Algorithm 1: Weighted adversarial training

Inputs network $f_{\boldsymbol{\theta}}$, training examples $\{\boldsymbol{x}_i\}_{i=1}^N$, number of steps for PGD K and step size of PGD η_1, learning rate η_2;
Output: robust network $f_{\boldsymbol{\theta}^*}$;
for *training iterations* **do**
  Read a mini-batch $B = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m\}$ from training set ;
  for *i=1,..,m* **do**
    Initialize $\boldsymbol{x}_i' = \boldsymbol{x}_i + 0.001\boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$;
    for *k = 1, 2, ..., K* **do**
      $\mathcal{L}(\boldsymbol{x}_i') = s(f_{\boldsymbol{\theta}^*}, \boldsymbol{x}_i', y_i) l(f_{\boldsymbol{\theta}^*}(\boldsymbol{x}_i'), y_i)$, where $s(f_{\boldsymbol{\theta}}, \boldsymbol{x}_i', y_i) = e^{-\alpha \text{ margin}(f_{\boldsymbol{\theta}}, \boldsymbol{x}_i + \boldsymbol{\delta}_i, y_i)}$;
      $\boldsymbol{x}_i' = \prod_{\mathbb{B}(\boldsymbol{x}_i, \epsilon)}(\boldsymbol{x}_i' + \eta_1 \text{sign}\nabla_{\boldsymbol{x}_i}\mathcal{L}(\boldsymbol{x}_i'))$, where $\prod$ is the projection operator;
    end
  end
  $\theta = \theta - \eta_2 \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{x}_i'))$
end

Evaluation During evaluation, for any test example $(\boldsymbol{x}_i, y_i)$, we define the normalized importance weights (normalized margin-aware weights) $\tilde{s}(f_{\boldsymbol{\theta}}, \boldsymbol{x}_i', y_i)$ as $\tilde{s}(f_{\boldsymbol{\theta}}, \boldsymbol{x}_i', y_i) := \frac{s(f_{\boldsymbol{\theta}}, \boldsymbol{x}_i', y_i)}{\sum_{i=1}^{N_{\text{test}}} s(f_{\boldsymbol{\theta}}, \boldsymbol{x}_i', y_i)}$. The normalized margin-aware weights could be interpreted as the the probability of attacking example $(\boldsymbol{x}_i, y_i)$. A uniform attack implies that the probability of attacking example $(\boldsymbol{x}_i, y_i)$ is $\frac{1}{N_{\text{test}}}$. For a non-uniform attack, the probability of attacking example $(\boldsymbol{x}_i, y_i)$ is $\tilde{s}(f_{\boldsymbol{\theta}}, \boldsymbol{x}_i', y_i)$. We argue that the traditional evaluation under uniform attack should be improved under the setting of non-uniform attack. More details are in the next section.

Experiments

Evaluation Metrics

Traditional evaluation metrics: Traditionally, we measure the performance of each method using natural accuracy on clean data, denoted as $\mathcal{A}_{nat}$. Robust accuracy $\mathcal{A}_{rob}$ is commonly used to evaluate the adversarial accuracy

$$\mathcal{A}_{rob} = \frac{1}{N_{\text{test}}}\sum_{i=1}^{N_{\text{test}}}[\mathbf{1}(f_{\boldsymbol{\theta}^*}(\boldsymbol{x}_i + \boldsymbol{\delta}_i^*), y_i)] \tag{17}$$

on the test examples uniformly. Note that $\mathcal{D}_n^{\text{test}}$ is the empirical distribution of the clean test examples and $\boldsymbol{\delta}^*$ is derived using the traditional unweighted loss function.

Our evaluation metric I: $\mathcal{A}_{sa}$ As we motivate in this paper, the evaluation of $\mathcal{A}_{rob}$ makes an unrealistic assumption that the adversary chooses to attack uniformly (although the perturbations at different examples $\boldsymbol{\delta}_i^*$ are different). Therefore $\mathcal{A}_{rob}$ is not necessarily the best way to evaluate the performance of the robustness under non-uniform attacks. We introduce an modified accuracy, namely $\mathcal{A}_{sa}$ that evaluate robustness under non-uniform attacks. To compute $\mathcal{A}_{sa}$,

the perturbations $\boldsymbol{\delta'}_i^*$ are crafted independently using the traditional unweighted loss. However, the attacker attacks the test examples non-uniformly, i.e., the adversarial examples are **sampled** according to a non-uniform distribution — the normalized importance weights:

$$\mathcal{A}_{sa} = \sum_{i=1}^{N_{\text{test}}} [\tilde{s}(f_{\boldsymbol{\theta}^*}, \boldsymbol{x}_i + \boldsymbol{\delta'}_i^*, y_i) \mathbf{1}(f_{\boldsymbol{\theta}^*}(\boldsymbol{x}_i + \boldsymbol{\delta'}_i^*), y_i)] \quad (18)$$

Our evaluation metric II: $\mathcal{A}_{tr}$Furthermore, we propose another evaluation metric, $\mathcal{A}_{tr}$. Here, the adversarial examples are not only crafted with importance-weighted loss, but also under the sophisticated selection (importance-based sampling):

$$\mathcal{A}_{tr} = \sum_{i=1}^{N_{\text{test}}} [\tilde{s}(f_{\boldsymbol{\theta}^*}, \boldsymbol{x}_i + \boldsymbol{\delta''}_i^*, y_i) \mathbf{1}(f_{\boldsymbol{\theta}^*}(\boldsymbol{x}_i + \boldsymbol{\delta''}_i^*), y_i)] \quad (19)$$

The perturbations $\boldsymbol{\delta''}^*$ are generated via the process in Algorithm 1. This $\mathcal{A}_{tr}$ reflects to what extent the attacker is able to transfer the margin-aware weights into the efficacy of the adversarial attacks, in terms of the generative process as well as sampling process.

Remark Empirically, these three metrics correspond to three different kind of adversarial attackers of different attacking power. $\mathcal{A}_{rob}$ is the traditional robust accuracy. Regarding this accuracy, the adversary is the weakest one in comparison to the others. This "naive" attacker attacks all samples uniformly and does not leverage the vulnerability of individual data points. $\mathcal{A}_{sa}$ is the accuracy evaluated on the adversarial examples, which are generated by the **unweighted** loss but sampled non-uniformly based on the normalized importance weights. When computing $\mathcal{A}_{sa}$, the network is dealing with a smarter attacker, since the adversary knows to attack vulnerable examples more frequently. Finally, $\mathcal{A}_{tr}$ measures the robustness of the trained model in the hardest case, where the adversarial examples are generated based on the **weighted** loss, but also are sampled based on the normalized importance weights. In this case, the attacker is the strongest one. It assigns larger energy to attack more vulnerable examples and samples such vulnerable adversarial examples more frequently. Therefore, when all hyperparameters (ϵ, α) are the same, we expect $\mathcal{A}_{tr} \leq \mathcal{A}_{sa} \leq \mathcal{A}_{rob}$ in most scenarios.

Hyperparameters of the margin-aware weights Recall that the importance weight $s(f_{\boldsymbol{\theta}}, \boldsymbol{x}'_i, y_i) = e^{-\alpha \ \mathsf{margin}(f_{\boldsymbol{\theta}}, \boldsymbol{x}_i + \boldsymbol{\delta}_i, y_i)}$. For a better understanding of the results, we clarify that the α used during training will be denoted as α_{train}. During test, the non-uniform attack model used to evaluate the robustness of a trained network uses the importance weight parameterized by α_{test}. Regardless training or testing, the value of α indicates the power of the adversarial attacker. If α_{train} is larger, then a stronger non-uniform attacker is included during training. Therefore, the resulted model should be able to withstand stronger non-uniform attacks. Similarly, if α_{test} is large, the attacker is able to exaggerate the re-weighting effect to a larger extent, corresponding to stronger attack power.

Defense	α_{train}	α_{test}	$\mathcal{A}_{\text{rob}}$ (%)	$\mathcal{A}_{\text{sa}}$ (%)	$\mathcal{A}_{\text{tr}}$(%)
PGD	-	1.0	93.95	74.85	74.65
PGD+ours	0.5	1.0	95.22	**80.54**	**80.53**
PGD	-	1.5	93.95	56.10	55.87
PGD+ours	0.5	1.5	95.22	**64.89**	**64.63**
PGD	-	2.0	93.95	35.32	35.04
PGD+ours	0.5	2.0	95.22	**44.96**	**44.70**
TRADES	-	1.0	95.59	83.18	83.07
TRADES+ours	2.0	1.0	95.20	**86.34**	**85.94**
TRADES	-	1.5	95.59	70.07	69.72
TRADES+ours	2.0	1.5	95.20	**78.10**	**77.22**
TRADES	-	2.0	95.59	52.15	51.52
TRADES+ours	2.0	2.0	95.20	**66.71**	**65.61**

Table 1: Robustness against non-uniform attacks on MNIST (Proposed Metrics). The adversarial examples are generated through 40-PGD with $\epsilon = 0.3$.

Defense	α_{train}	α_{test}	$\mathcal{A}_{\text{rob}}$ (%)	$\mathcal{A}_{\text{sa}}$ (%)	$\mathcal{A}_{\text{tr}}$(%)
PGD	-	1.0	49.29	25.09	22.91
PGD+ours	2.0	1.0	49.53	**26.49**	**23.94**
PGD	-	1.5	49.29	17.33	15.10
PGD+ours	2.0	1.5	49.53	**18.92**	**16.25**
PGD	-	2.0	49.29	11.66	9.72
PGD+ours	2.0	2.0	49.53	**13.19**	**10.81**
TRADES	-	1.0	53.38	33.36	31.10
TRADES+ours	2.0	1.0	54.10	**36.36**	**33.26**
TRADES	-	1.5	53.38	25.92	23.31
TRADES+ours	2.0	1.5	54.10	**29.52**	**25.84**
TRADES	-	2.0	53.38	19.78	17.14
TRADES+ours	2.0	2.0	54.10	**23.62**	**19.79**

Table 2: Robustness against non-uniform attacks on CIFAR10 (Proposed Metrics). The adversarial examples are generated through 20-PGD with $\epsilon = 0.031$.

Defense	α_{train}	α_{test}	$\mathcal{A}_{\text{rob}}$ (%)	$\mathcal{A}_{\text{sa}}$ (%)	$\mathcal{A}_{\text{tr}}$(%)
PGD	-	0.5	22.27	17.91	17.14
PGD+ours	0.3	0.5	22.75	**19.25**	**18.53**
PGD	-	1.0	22.27	14.33	12.98
PGD+ours	0.3	1.0	22.75	**16.34**	**14.99**
PGD	-	1.5	22.27	11.42	9.79
PGD+ours	0.3	1.5	22.75	**13.88**	**12.06**
PGD	-	2.0	22.27	9.04	7.32
PGD+ours	0.3	2.0	22.75	**11.80**	**9.66**
TRADES	-	0.5	27.90	23.95	22.94
TRADES+ours	5.0	0.5	28.16	**25.18**	**24.25**
TRADES	-	1.0	27.90	20.57	18.87
TRADES+ours	5.0	1.0	28.16	**22.40**	**20.74**
TRADES	-	1.5	27.90	17.74	15.33
TRADES+ours	5.0	1.5	28.16	**20.02**	**17.71**
TRADES	-	2.0	27.90	15.33	12.58
TRADES+ours	5.0	2.0	28.16	**17.79**	**15.01**

Table 3: Robustness against non-uniform attacks on Tiny ImageNet (Proposed Metrics). The adversarial examples are generated through 10-PGD with $\epsilon = 0.016$.

Experimental Results and Analysis

In this section, we firstly show that while the robust network trained using unweighted adversarial training objective will fail in the presence of non-uniform attacks, the network trained by our defense mechanism is able to withstand the strong non-uniform attacks. Then, we verify that our proposed re-weighting approach, although designed for stronger non-uniform attacks, matches the state-of-the-art adversarial training based algorithms even in traditional uniform attack settings. Experiments are conducted on MNIST (LeCun 1998), CIFAR10 (Krizhevsky 2012) and Tiny ImageNet (Le and Yang 2015) datasets. Finally, we evaluate the trained models under different attack algorithms and the DRO setting (Staib and Jegelka 2017).

Baselines and experimental settings We use adversarial training (Madry et al. 2017) and TRADES (Zhang et al. 2019b) as baselines. In the context of TRADES, the robust regularization term is governed by a penalty strength λ. Moreover, regarding CIFAR10, we also include our reproduced results of IAAT(Balaji, Goldstein, and Hoffman 2019), YOPO(Zhang et al. 2019a) and AT4Free(Shafahi et al. 2019b). We then conduct ablation studies of the reweighting approaches on top of the loss function of the baselines.

Robustness under non-uniform attack As argued previously, the core of this work is that the minimax optimization objective for adversarial loss should take the distribution of adversarial examples into account and it should help the network defend against non-uniform attackers. Now, we show that the models trained with traditional adversarial training algorithms (PGD-based adversarial training and TRADES) will perform poorly in the presence of a non-uniform attacker whereas our method is able to better defend against such non-uniform attacker. The experimental results are demonstrated in Table 1, Table 2 and Table 3. For a given α_{attack}, we observe that $\mathcal{A}_{sa}$ and $\mathcal{A}_{tr}$ are smaller than $\mathcal{A}_{robust}$ in most cases on all datasets. For instance, on MNIST, if the attacker scales the margin error by 2, i.e. $\alpha_{attack} = 2.0$, and it generates and sample adversarial examples using the rescaled margin error, the accuracy on adversarial examples of baseline TRADES model will decrease dramatically, with $\mathcal{A}_{rob} = 95.59\%$ dropping to $\mathcal{A}_{sa} = 52.15\%$ and $\mathcal{A}_{tr} = 51.52\%$.

Moreover, for a trained model (trained with a specific α_{train}), if the α_{test} goes larger, indicating that the attacker is more powerful, $\mathcal{A}_{sa}$ and $\mathcal{A}_{tr}$ will drop even further. However, faced with the same non-uniform attacker, our model is able to achieve better robustness. For example, on MNIST, if trained with $\alpha_{train} = 2.0$, the modified TRADES model is able to achieve $\mathcal{A}_{sa} = 66.71\%$ and $\mathcal{A}_{tr} = 65.61\%$ when defending against $\alpha_{attack} = 2.0$, in comparison to $\mathcal{A}_{sa} = 52.15\%$ and $\mathcal{A}_{tr} = 51.52\%$ of the baseline method. Our model is able to consistently beat the baselines under varying α_{attack}'s for all tested datasets.

Robustness under uniform attack Comparing the results in Table 4 and Table 5, our modified defense mechanism,designed for non-uniform attacks, matches or slightly outperforms the state-of-the-art uniform attacks. On CIFAR10, the best robust accuracy of TRADES-trained model using our method is 54.10%, which is better than 53.38% of the baseline model. Actually, we are able to obtain similar observations from the results on MNIST and Tiny ImageNet on models trained using PGD and TRADES. To summarize, uniform attack results show that our modified training objective maintain the performance under uniform attacks and might even increase the performance of the trained models under traditional metrics.

Defense	$1/\lambda$	α_{train}	$\mathcal{A}_{\text{nat}}$ (%)	$\mathcal{A}_{\text{rob}}$(%)
AT4Free	-	-	81.80	39.00
YOPO-5-3	-	-	83.99	44.72
IAAT	-	-	88.60	48.27
PGD	-	-	82.00	49.29
PGD+ours	-	0.01	82.33	49.08
PGD+ours	-	0.05	81.75	49.25
PGD+ours	-	0.1	81.60	49.53
TRADES	5	-	82.93	53.38
TRADES+ours	5	0.1	82.98	54.10
TRADES+ours	5	1.0	83.17	54.05
TRADES+ours	5	1.5	82.83	53.91
TRADES+ours	5	2.0	83.41	54.10

Table 4: Natural error and robust error under uniform attacks on CIFAR10 (Traditional Metrics). The adversarial examples are generated through 20-PGD with $\epsilon = 0.031$.

Defense	$1/\lambda$	α_{train}	$\mathcal{A}_{\text{nat}}$ (%)	$\mathcal{A}_{\text{rob}}$(%)
PGD	-	-	35.02	22.27
PGD+ours	6	0.1	35.76	23.16
PGD+ours	6	0.15	34.60	22.17
PGD+ours	6	0.2	35.52	22.75
PGD+ours	6	0.25	33.10	21.36
PGD+ours	6	0.3	34.26	22.75
TRADES	6	-	45.44	27.90
TRADES+ours	6	0.5	44.58	28.28
TRADES+ours	6	1.5	45.64	28.24
TRADES+ours	6	2.0	45.72	28.74
TRADES+ours	6	2.5	45.34	28.44
TRADES+ours	6	3.5	45.35	28.26
TRADES+ours	6	5.0	45.15	28.16

Table 5: Natural error and robust error under uniform attacks on Tiny ImageNet (Traditional Metrics). The adversarial examples are generated via 10-PGD with $\epsilon = 0.016$.

Conclusion

This work studies the objective function for adversarial training. We argue that adversarial examples are not all created equal, and therefore the loss function should learn to weigh the individual examples during training. Our method improves the performance of both clean data natural accuracy and robust accuracy of the baseline under both uniform and non-uniform attack schemes. The learnable weighted minimax risk motivates us to analyze the adversarial risk from a different perspective. That is, we should introduce flexibility to the model and let it assign different penalties to the individual data points during adversarial training.

Acknowledgements

Huang is supported by startup fund from Department of Computer Science of University of Maryland, National Science Foundation IIS-1850220 CRII Award 030742-00001, DOD-DARPA-Defense Advanced Research Projects Agency Guaranteeing AI Robustness against Deception (GARD), Laboratory for Physical Sciences at University of Maryland, and Adobe, Capital One and JP Morgan faculty fellowships.

Ethics Statement

Adversarial examples could cause extremely high threat to our society, adversarial defenses are therefore crucial. Despite the impressive accuracy of machine learning on diverse tasks such as object recognition, speech recognition, and playing Go, classifiers still fail catastrophically in the presence of small imperceptible but adversarial perturbations. The existence of such "adversarial examples" exposes a serious vulnerability in current ML systems such as autonomous driving systems, network systems and security monitoring systems. This vulnerability exposes our lives and national security at risk.

Adversarial examples could cause extremely high threat to our society, adversarial defenses are therefore crucial. Despite the impressive accuracy of machine learning on diverse tasks such as object recognition, speech recognition, and playing Go, classifiers still fail catastrophically in the presence of small imperceptible but adversarial perturbations. The existence of such "adversarial examples" exposes a serious vulnerability in current ML systems such as autonomous driving systems, network systems and security monitoring systems. This vulnerability exposes our lives and national security at risk.

Our work has the potential to improve almost all existing adversarial defense mechanisms using a robust error objective function. Our work provides a new methodology of designing new objective functions in adversarial defenses, a new perspective that is complementary to almost all previous works on adversarial defenses. If plugged into other methods, our method has the potential to build neural networks with stronger robustness without much hampering of the accuracy.

References

Athalye, A.; Carlini, N.; and Wagner, D. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples.

Balaji, Y.; Goldstein, T.; and Hoffman, J. 2019. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *arXiv preprint arXiv:1910.08051* .

Ben-Tal, A.; Den Hertog, D.; De Waegenaere, A.; Melenberg, B.; and Rennen, G. 2013. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science* 59(2): 341–357.

Blanchet, J.; Kang, Y.; Zhang, F.; He, F.; and Hu, Z. 2017. Doubly robust data-driven distributionally robust optimization. *arXiv preprint arXiv:1705.07168* .

Carlini, N.; and Wagner, D. 2017. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 3–14.

Delage, E.; and Ye, Y. 2010. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research* 58(3): 595–612.

Duchi, J.; Glynn, P.; and Namkoong, H. 2016. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv preprint arXiv:1610.03425* .

Elsayed, G.; Krishnan, D.; Mobahi, H.; Regan, K.; and Bengio, S. 2018. Large margin deep networks for classification. In *Advances in neural information processing systems*, 842–852.

Finlay, C.; and Oberman, A. M. 2019. Scaleable input gradient regularization for adversarial robustness. *arXiv preprint arXiv:1905.11468* .

Gao, R.; and Kleywegt, A. J. 2016. Distributionally robust stochastic optimization with Wasserstein distance. *arXiv preprint arXiv:1604.02199* .

Goh, J.; and Sim, M. 2010. Distributionally robust optimization and its tractable approximations. *Operations research* 58(4-part-1): 902–917.

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, S.; Papernot, N.; Goodfellow, I.; Duan, Y.; and Abbeel, P. 2017. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284* .

Jakubovitz, D.; and Giryes, R. 2018. Improving dnn robustness to adversarial attacks using jacobian regularization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 514–529.

Kannan, H.; Kurakin, A.; and Goodfellow, I. J. 2018. Adversarial Logit Pairing. *ArXiv* abs/1803.06373.

Krizhevsky, A. 2012. Learning Multiple Layers of Features from Tiny Images. *University of Toronto* .

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

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N* 7.

LeCun, Y. 1998. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/* .

Ma, X.; Li, B.; Wang, Y.; Erfani, S. M.; Wijewickrema, S.; Schoenebeck, G.; Song, D.; Houle, M. E.; and Bailey, J. 2018. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613* .

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

Meng, D.; and Chen, H. 2017. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 135–147.

Mosbach, M.; Andriushchenko, M.; Trost, T.; Hein, M.; and Klakow, D. 2018. Logit pairing methods can fool gradient-based attacks. *arXiv preprint arXiv:1810.12042* .

Namkoong, H.; and Duchi, J. C. 2016. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Advances in neural information processing systems*, 2208–2216.

Qin, C.; Martens, J.; Gowal, S.; Krishnan, D.; Dvijotham, K.; Fawzi, A.; De, S.; Stanforth, R.; and Kohli, P. 2019. Adversarial robustness through local linearization. In *Advances in Neural Information Processing Systems*, 13824–13833.

Ross, A. S.; and Doshi-Velez, F. 2018. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Thirty-second AAAI conference on artificial intelligence*.

Shafahi, A.; Ghiasi, A.; Huang, F.; and Goldstein, T. 2019a. Label Smoothing and Logit Squeezing: A Replacement for Adversarial Training? *arXiv preprint arXiv:1910.11585* .

Shafahi, A.; Najibi, M.; Ghiasi, M. A.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L. S.; Taylor, G.; and Goldstein, T. 2019b. Adversarial training for free! In *Advances in Neural Information Processing Systems*, 3353–3364.

Shaham, U.; Yamada, Y.; and Negahban, S. 2018. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. *Neurocomputing* 307: 195–204.

Staib, M.; and Jegelka, S. 2017. Distributionally robust deep learning as a generalization of adversarial training. In *NIPS workshop on Machine Learning and Computer Security*.

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

Xu, W.; Evans, D.; and Qi, Y. 2017. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155* .

Zhang, D.; Zhang, T.; Lu, Y.; Zhu, Z.; and Dong, B. 2019a. You Only Propagate Once: Painless Adversarial Training Using Maximal Principle. *arXiv preprint arXiv:1905.00877* .

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E. P.; Ghaoui, L. E.; and Jordan, M. I. 2019b. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573* .

Zhang, Y.; and Liang, P. 2019. Defending against white-box adversarial attacks via randomized discretization. *arXiv preprint arXiv:1903.10586* .