Theoretically Principled Trade-off between Robustness and Accuracy

Hongyang Zhang [1 2] **Yaodong Yu** [3] **Jiantao Jiao** [4] **Eric P. Xing** [1 5] **Laurent El Ghaoui** [4] **Michael I. Jordan** [4]

Abstract

We identify a trade-off between robustness and accuracy that serves as a guiding principle in the design of defenses against adversarial examples. Although this problem has been widely studied empirically, much remains unknown concerning the theory underlying this trade-off. In this work, we decompose the prediction error for adversarial examples (robust error) as the sum of the natural (classification) error and boundary error, and provide a differentiable upper bound using the theory of classification-calibrated loss, which is shown to be the tightest possible upper bound uniform over all probability distributions and measurable predictors. Inspired by our theoretical analysis, we also design a new defense method, TRADES, to trade adversarial robustness off against accuracy. Our proposed algorithm performs well experimentally in real-world datasets. The methodology is the foundation of our entry to the NeurIPS 2018 Adversarial Vision Challenge in which we won the 1st place out of ~2,000 submissions, surpassing the runner-up approach by 11.41% in terms of mean ℓ_2 perturbation distance.

1. Introduction

In response to the vulnerability of deep neural networks to small perturbations around input data (Szegedy et al., 2013), adversarial defenses have been an imperative object of study in machine learning (Huang et al., 2017), computer vision (Song et al., 2018; Xie et al., 2017; Meng & Chen, 2017), natural language processing (Jia & Liang, 2017), and many other domains. In machine learning, study of adversarial defenses has led to significant advances in understanding and defending against adversarial threat (He et al., 2017). In computer vision and natural language processing, adversarial defenses serve as indispensable building blocks for a range of security-critical systems and applications, such as autonomous cars and speech recognition authorization. The problem of adversarial defenses can be stated as that of learning a classifier with high test accuracy on both natural and *adversarial examples*. The adversarial example for a given labeled data $(\boldsymbol{x}, y)$ is a data point $\boldsymbol{x}'$ that causes a classifier c to output a different label on $\boldsymbol{x}'$ than y, but is "imperceptibly similar" to $\boldsymbol{x}$. Given the difficulty of providing an operational definition of "imperceptible similarity," adversarial examples typically come in the form of *restricted attacks* such as ϵ-bounded perturbations (Szegedy et al., 2013), or *unrestricted attacks* such as adversarial rotations, translations, and deformations (Brown et al., 2018; Engstrom et al., 2017; Gilmer et al., 2018; Xiao et al., 2018; Alaifari et al., 2019; Zhang et al., 2019a). The focus of this work is the former setting, though our framework can be generalized to the latter.

Despite a large literature devoted to improving the robustness of deep-learning models, many fundamental questions remain unresolved. One of the most important questions is how to trade off adversarial robustness against natural accuracy. Statistically, robustness can be be at odds with accuracy when no assumptions are made on the data distribution (Tsipras et al., 2019). This has led to an empirical line of work on adversarial defense that incorporates various kinds of assumptions (Su et al., 2018; Kurakin et al., 2017). On the theoretical front, methods such as *relaxation based defenses* (Kolter & Wong, 2018; Raghunathan et al., 2018a) provide provable guarantees for adversarial robustness. They, however, ignore the performance of classifier on the non-adversarial examples, and thus leave open the theoretical treatment of the putative robustness/accuracy trade-off.

The problem of adversarial defense becomes more challenging when computational issues are considered. For example, the straightforward empirical risk minimization (ERM) formulation of robust classification involves minimizing the robust 0-1 loss $\max_{\boldsymbol{x}':\|\boldsymbol{x}'-\boldsymbol{x}\|\leq\epsilon} \mathbf{1}\{c(\boldsymbol{x}') \neq y\}$, a loss which is NP-hard to optimize even if $\epsilon = 0$ in general. Hence, it is natural to expect that some prior work on adversarial defense replaced the 0-1 loss $\mathbf{1}(\cdot)$ with a surrogate loss (Madry et al., 2018; Kurakin et al., 2017; Uesato et al., 2018). However, there is little theoretical guarantee on the tightness of this approximation.

[1]Carnegie Mellon University [2]Toyota Technological Institute at Chicago [3]University of Virginia [4]University of California, Berkeley [5]Petuum Inc.. Correspondence to: Hongyang Zhang <hongyanz@cs.cmu.edu>, Yaodong Yu <yy8ms@virginia.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

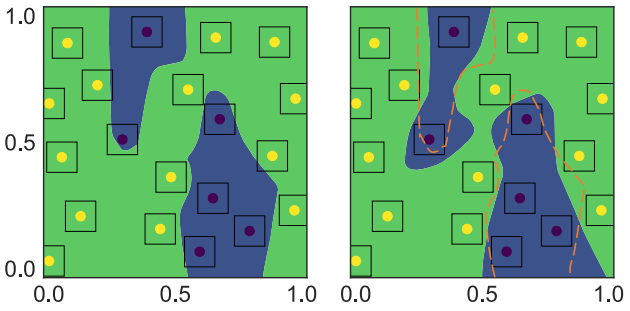


Figure 1. **Left figure:** decision boundary learned by natural training method. **Right figure:** decision boundary learned by our adversarial training method, where the orange dotted line represents the decision boundary in the left figure. It shows that both methods achieve zero natural training error, while our adversarial training method achieves better robust training error than the natural training method.

1.1. Our methodology and results

We begin with an example that illustrates the trade-off between accuracy and adversarial robustness in Section 2.4, a phenomenon which has been demonstrated by Tsipras et al. (2019), but without theoretical guarantees. We constructed a toy example where the Bayes optimal classifier achieves *natural error* 0% and *robust error* 100%, while the trivial all-one classifier achieves both *natural error* and *robust error* 50% (Table 1). Despite a large literature on the analysis of robust error in terms of generalization (Schmidt et al., 2018; Cullina et al., 2018; Yin et al., 2018) and computational complexity (Bubeck et al., 2018b;a), the trade-off between the natural error and the robust error has not been a focus of theoretical study.

We show that the *robust error* can in general be bounded tightly using two terms: one corresponds to the *natural error* measured by a *surrogate* loss function, and the other corresponds to how likely the input features are close to the ϵ-extension of the decision boundary, termed as the *boundary error*. We then minimize the differentiable upper bound. Our theoretical analysis naturally leads to a new formulation of adversarial defense which has several appealing properties; in particular, it inherits the benefits of scalability to large datasets exhibited by Tiny ImageNet, and the algorithm achieves state-of-the-art performance on a range of benchmarks while providing theoretical guarantees. For example, while the defenses overviewed in (Athalye et al., 2018) achieve robust accuracy no higher than ~47% under white-box attacks, our method achieves robust accuracy as high as ~57% in the same setting. The methodology is the foundation of our entry to the NeurIPS 2018 Adversarial Vision Challenge where we won first place out of ~2,000 submissions, surpassing the runner-up approach by 11.41% in terms of mean ℓ_2 perturbation distance.

1.2. Summary of contributions

Our work tackles the problem of trading accuracy off against robustness and advances the state-of-the-art in multiple ways.

- Theoretically, we characterize the trade-off between accuracy and robustness for classification problems via decomposing the robust error as the sum of the natural error and the boundary error. We provide differentiable upper bounds on both terms using the theory of classification-calibrated loss, which are shown to be the tightest upper bounds uniform over all probability distributions and measurable predictors.
- Algorithmically, inspired by our theoretical analysis, we propose a new formulation of adversarial defense, TRADES, as optimizing a regularized surrogate loss. The loss consists of two terms: the term of empirical risk minimization encourages the algorithm to maximize the natural accuracy, while the regularization term encourages the algorithm to push the decision boundary away from the data, so as to improve adversarial robustness (see Figure 1).
- Experimentally, we show that our proposed algorithm outperforms state-of-the-art methods under both black-box and white-box threat models. In particular, the methodology won the final round of the NeurIPS 2018 Adversarial Vision Challenge.

2. Preliminaries

We illustrate our methodology using the framework of binary classification, but it can be generalized to other settings as well.

2.1. Notation

We will use *bold capital* letters such as $\boldsymbol{X}$ and $\boldsymbol{Y}$ to represent random vector, *bold lower-case* letters such as $\boldsymbol{x}$ and $\boldsymbol{y}$ to represent realization of random vector, *capital* letters such as X and Y to represent random variable, and *lower-case* letters such as x and y to represent realization of random variable. Specifically, we denote by $\boldsymbol{x} \in \mathcal{X}$ the sample instance, and by $y \in \{-1, +1\}$ the label, where $\mathcal{X} \subseteq \mathbb{R}^d$ indicates the instance space. $\mathsf{sign}(x)$ represents the sign of scalar x with $\mathsf{sign}(0) = +1$. Denote by $f : \mathcal{X} \to \mathbb{R}$ the *score function* which maps an instance to a confidence value associated with being positive. It can be parametrized, e.g., by deep neural networks. The associated binary classifier is $\mathsf{sign}(f(\cdot))$. We will frequently use $\mathbf{1}\{\text{event}\}$, the 0-1 loss, to represent an indicator function that is 1 if an event happens and 0 otherwise. For norms, we denote by $\|\boldsymbol{x}\|$ a generic norm. Examples of norms include $\|\boldsymbol{x}\|_\infty$, the infinity norm of vector $\boldsymbol{x}$, and $\|\boldsymbol{x}\|_2$, the ℓ_2 norm of

vector $\boldsymbol{x}$. We use $\mathbb{B}(\boldsymbol{x}, \epsilon)$ to represent a neighborhood of $\boldsymbol{x}$: $\{\boldsymbol{x}' \in \mathcal{X} : \|\boldsymbol{x}' - \boldsymbol{x}\| \leq \epsilon\}$. For a given score function f, we denote by $\mathrm{DB}(f)$ the decision boundary of f; that is, the set $\{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) = 0\}$. The set $\mathbb{B}(\mathrm{DB}(f), \epsilon)$ denotes the neighborhood of the decision boundary of f: $\{\boldsymbol{x} \in \mathcal{X} : \exists \boldsymbol{x}' \in \mathbb{B}(\boldsymbol{x}, \epsilon) \text{ s.t. } f(\boldsymbol{x})f(\boldsymbol{x}') \leq 0\}$. For a given function $\psi(\boldsymbol{u})$, we denote by $\psi^*(\boldsymbol{v}) := \sup_{\boldsymbol{u}}\{\boldsymbol{u}^T\boldsymbol{v} - \psi(\boldsymbol{u})\}$ the conjugate function of ψ, by ψ^{**} the bi-conjugate, and by ψ^{-1} the inverse function. We will frequently use $\phi(\cdot)$ to indicate the surrogate of 0-1 loss.

2.2. Robust (classification) error

In the setting of adversarial learning, we are given a set of instances $\boldsymbol{x}_1, ..., \boldsymbol{x}_n \in \mathcal{X}$ and labels $y_1, ..., y_n \in \{-1, +1\}$. We assume that the data are sampled from an unknown distribution $(\boldsymbol{X}, Y) \sim \mathcal{D}$. To characterize the robustness of a score function $f : \mathcal{X} \to \mathbb{R}$, Schmidt et al. (2018); Cullina et al. (2018); Bubeck et al. (2018b) defined *robust (classification) error* under the threat model of bounded ϵ perturbation: $\mathcal{R}_{\mathrm{rob}}(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\mathbf{1}\{\exists \boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon) \text{ s.t. } f(\boldsymbol{X}')Y \leq 0\}$. This is in sharp contrast to the standard measure of classifier performance—the *natural (classification) error* $\mathcal{R}_{\mathrm{nat}}(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\mathbf{1}\{f(\boldsymbol{X})Y \leq 0\}$ We note that the two errors satisfy $\mathcal{R}_{\mathrm{rob}}(f) \geq \mathcal{R}_{\mathrm{nat}}(f)$ for all f; the robust error is equal to the natural error when $\epsilon = 0$.

2.3. Boundary error

We introduce the *boundary error* defined as $\mathcal{R}_{\mathrm{bdy}}(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\mathbf{1}\{\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f), \epsilon), f(\boldsymbol{X})Y > 0\}$. We have the following decomposition of $\mathcal{R}_{\mathrm{rob}}(f)$:

$$\mathcal{R}_{\mathrm{rob}}(f) = \mathcal{R}_{\mathrm{nat}}(f) + \mathcal{R}_{\mathrm{bdy}}(f). \tag{1}$$

2.4. Trade-off between natural and robust errors

Our study is motivated by the trade-off between natural and robust errors. Tsipras et al. (2019) showed that training robust models may lead to a reduction of standard accuracy. To illustrate the phenomenon, we provide a toy example.

Example. Consider the case $(X, Y) \sim \mathcal{D}$, where the marginal distribution over the instance space is a uniform distribution over $[0, 1]$, and for $k = 0, 1, ..., \lceil \frac{1}{2\epsilon} - 1 \rceil$,

$$\begin{aligned}\eta(x) &:= \Pr(Y = 1 | X = x) \\ &= \begin{cases} 0, & x \in [2k\epsilon, (2k+1)\epsilon), \\ 1, & x \in ((2k+1)\epsilon, (2k+2)\epsilon]. \end{cases}\end{aligned} \tag{2}$$

See Figure 2 for the visualization of $\eta(x)$. We consider two classifiers: a) the Bayes optimal classifier $\mathsf{sign}(2\eta(x) - 1)$; b) the all-one classifier which always outputs "positive." Table 1 displays the trade-off between natural and robust errors: the minimal natural error is achieved by the Bayes optimal classifier with large robust error, while the optimal robust error is achieved by the all-one classifier with large natural error.

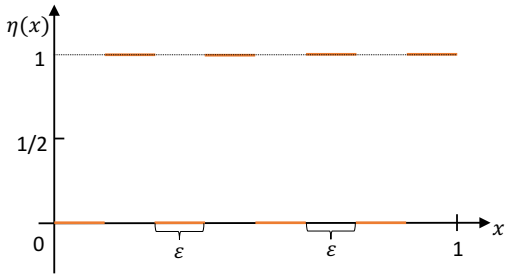


Figure 2. Counterexample given by Eqn. (2).

Table 1. Comparisons of natural and robust errors of Bayes optimal classifier and all-one classifier in example (2). The Bayes optimal classifier has the optimal natural error while the all-one classifier has the optimal robust error.

	Bayes Optimal Classifier	All-One Classifier
$\mathcal{R}_{\mathrm{nat}}$	0 (optimal)	1/2
$\mathcal{R}_{\mathrm{bdy}}$	1	0
$\mathcal{R}_{\mathrm{rob}}$	1	1/2 (optimal)

Our goal. In practice, one may prefer to trade-off between robustness and accuracy by introducing weights in (1) to bias more towards the natural error or the boundary error. Noting that both the natural error and the boundary error involve 0-1 loss functions, our goal is to devise *tight* differentiable upper bounds on both of these terms. Towards this goal, we utilize the theory of classification-calibrated loss.

2.5. Classification-calibrated surrogate loss

Definition. Minimization of the 0-1 loss in the natural and robust errors is computationally intractable and the demands of computational efficiency have led researchers to focus on minimization of a tractable *surrogate loss*, $\mathcal{R}_\phi(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\phi(f(\boldsymbol{X})Y)$. We then need to find quantitative relationships between the excess errors associated with ϕ and those associated with 0–1 loss. We make a weak assumption on ϕ: it is *classification-calibrated* (Bartlett et al., 2006). Formally, for $\eta \in [0, 1]$, define the *conditional ϕ-risk* by

$$H(\eta) := \inf_{\alpha\in\mathbb{R}} C_\eta(\alpha) := \inf_{\alpha\in\mathbb{R}} \left(\eta\phi(\alpha) + (1-\eta)\phi(-\alpha)\right),$$

and define $H^-(\eta) := \inf_{\alpha(2\eta-1)\leq 0} C_\eta(\alpha)$. The classification-calibrated condition requires that imposing the constraint that α has an inconsistent sign with the Bayes decision rule $\mathsf{sign}(2\eta - 1)$ leads to a strictly larger ϕ-risk:

Assumption 1 (Classification-Calibrated Loss)**.** *We assume that the surrogate loss ϕ is classification-calibrated, meaning that for any $\eta \neq 1/2$, $H^-(\eta) > H(\eta)$.*

We argue that Assumption 1 is indispensable for classification problems, since without it the Bayes optimal clas-

Table 2. Examples of classification-calibrated loss ϕ and associated ψ-transform. Here $\psi_{\log}(\theta) = \frac{1}{2}(1-\theta)\log_2(1-\theta) + \frac{1}{2}(1+\theta)\log_2(1+\theta)$.

Loss	$\phi(\alpha)$	$\psi(\theta)$
Hinge	$\max\{1-\alpha, 0\}$	θ
Sigmoid	$1-\tanh(\alpha)$	θ
Exponential	$\exp(-\alpha)$	$1-\sqrt{1-\theta^2}$
Logistic	$\log_2(1+\exp(-\alpha))$	$\psi_{\log}(\theta)$

sifier cannot be the minimizer of the ϕ-risk. Examples of classification-calibrated loss include hinge loss, sigmoid loss, exponential loss, logistic loss, and many others (see Table 2).

Properties. Classification-calibrated loss has many structural properties that one can exploit. We begin by introducing a functional transform of classification-calibrated loss ϕ which was proposed by Bartlett et al. (2006). Define the function $\psi : [0,1] \to [0,\infty)$ by $\psi = \widetilde{\psi}^{**}$, where $\widetilde{\psi}(\theta) := H^-\left(\frac{1+\theta}{2}\right) - H\left(\frac{1+\theta}{2}\right)$. Indeed, the function $\psi(\theta)$ is the largest convex lower bound on $H^-\left(\frac{1+\theta}{2}\right) - H\left(\frac{1+\theta}{2}\right)$. The value $H^-\left(\frac{1+\theta}{2}\right) - H\left(\frac{1+\theta}{2}\right)$ characterizes how close the surrogate loss ϕ is to the class of non-classification-calibrated losses.

Below we state useful properties of the ψ-transform. We will frequently use the function ψ to bound $\mathcal{R}_{\rm rob}(f) - \mathcal{R}^*_{\rm nat}$.

Lemma 2.1 (Bartlett et al. (2006))**.** *Under Assumption 1, the function ψ has the following properties: ψ is non-decreasing, continuous, convex on $[0,1]$ and $\psi(0) = 0$.*

3. Relating 0-1 loss to surrogate loss

In this section, we present our main theoretical contributions for binary classification and compare our results with prior literature. Binary classification problems have received significant attention in recent years as many competitions evaluate the performance of robust models on binary classification problems (Brown et al., 2018). We defer the discussion of multi-class problems to Section 4.

3.1. Upper bound

Our analysis leads to a guarantee on the performance of surrogate loss minimization. Intuitively, by Eqn. (1), $\mathcal{R}_{\rm rob}(f) - \mathcal{R}^*_{\rm nat} = \mathcal{R}_{\rm nat}(f) - \mathcal{R}^*_{\rm nat} + \mathcal{R}_{\rm bdy}(f) \le \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \mathcal{R}_{\rm bdy}(f)$, where the last inequality holds because we choose ϕ as a classification-calibrated loss (Bartlett et al., 2006). This leads to the following result.

Theorem 3.1. *Let $\mathcal{R}_\phi(f) := \mathbb{E}\phi(f(\boldsymbol{X})Y)$ and $\mathcal{R}^*_\phi := \min_f \mathcal{R}_\phi(f)$. Under Assumption 1, for any non-negative loss function ϕ such that $\phi(0) \ge 1$, any measurable $f : \mathcal{X} \to \mathbb{R}$, any probability distribution on $\mathcal{X} \times \{\pm 1\}$, and any $\lambda > 0$, we have*[1]

$$\begin{aligned}
&\mathcal{R}_{\rm rob}(f) - \mathcal{R}^*_{\rm nat} \\
&\le \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \Pr[\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f), \epsilon), f(\boldsymbol{X})Y > 0] \\
&\le \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \mathbb{E} \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} \phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda).
\end{aligned}$$

Quantity governing model robustness. Our result provides a formal justification for the existence of adversarial examples: learning models are vulnerable to small adversarial attacks because the probability that data lie around the decision boundary of the model, $\Pr[\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f), \epsilon), f(\boldsymbol{X})Y > 0]$, is large. As a result, small perturbations may move the data point to the wrong side of the decision boundary, leading to weak robustness of classification models.

3.2. Lower bound

We now establish a lower bound on $\mathcal{R}_{\rm rob}(f) - \mathcal{R}^*_{\rm nat}$. Our lower bound matches our analysis of the upper bound in Section 3.1 up to an arbitrarily small constant.

Theorem 3.2. *Suppose that $|\mathcal{X}| \ge 2$. Under Assumption 1, for any non-negative loss function ϕ such that $\phi(x) \to 0$ as $x \to +\infty$, any $\xi > 0$, and any $\theta \in [0,1]$, there exists a probability distribution on $\mathcal{X} \times \{\pm 1\}$, a function $f : \mathbb{R}^d \to \mathbb{R}$, and a regularization parameter $\lambda > 0$ such that $\mathcal{R}_{\rm rob}(f) - \mathcal{R}^*_{\rm nat} = \theta$ and*

$$\begin{aligned}
&\psi\Big(\theta - \mathbb{E} \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} \phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda)\Big) \le \mathcal{R}_\phi(f) - \mathcal{R}^*_\phi \\
&\qquad \le \psi\left(\theta - \mathbb{E} \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} \phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda)\right) + \xi.
\end{aligned}$$

Theorem 3.2 demonstrates that in the presence of extra conditions on the loss function, i.e., $\lim_{x \to +\infty} \phi(x) = 0$, the upper bound in Section 3.1 is tight. The condition holds for all the losses in Table 2.

4. Algorithmic Design for Defenses

Optimization. Theorems 3.1 and 3.2 shed light on algorithmic designs of adversarial defenses. In order to minimize $\mathcal{R}_{\rm rob}(f) - \mathcal{R}^*_{\rm nat}$, the theorems suggest minimizing[2]

$$\min_f \mathbb{E}\Big\{ \underbrace{\phi(f(\boldsymbol{X})Y)}_{\text{for accuracy}} + \underbrace{\max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} \phi(f(\boldsymbol{X})f(\boldsymbol{X}')/\lambda)}_{\text{regularization for robustness}} \Big\}. \tag{3}$$

[1] We study the population form of the risk functions, and mention that by incorporating the generalization theory for classification-calibrated losses (Bartlett et al., 2006) one can extend the analysis to finite samples. We leave this analysis for future research.

[2] For simplicity of implementation, we do not use the function ψ^{-1} and rely on λ to approximately reflect the effect of ψ^{-1}, the trade-off between the natural error and the boundary error, and the tight approximation of the boundary error using the corresponding surrogate loss function.

We name our method **TRADES** (TRadeoff-inspired Adversarial DEfense via Surrogate-loss minimization).

Intuition behind the optimization. Problem (3) captures the trade-off between the natural and robust errors: the first term in (3) encourages the natural error to be optimized by minimizing the "difference" between $f(\boldsymbol{X})$ and Y, while the second regularization term encourages the output to be smooth, that is, it pushes the decision boundary of classifier away from the sample instances via minimizing the "difference" between the prediction of natural example $f(\boldsymbol{X})$ and that of adversarial example $f(\boldsymbol{X}')$. This is conceptually consistent with the argument that smoothness is an indispensable property of robust models (Cisse et al., 2017). The tuning parameter λ plays a critical role on balancing the importance of natural and robust errors. To see how the λ affects the solution in the example of Section 2.4, problem (3) tends to the Bayes optimal classifier when $\lambda \to +\infty$, and tends to the all-one classifier when $\lambda \to 0$.

Comparisons with prior work. We compare our approach with several related lines of research in the prior literature. One of the best known algorithms for adversarial defense is based on *robust optimization* (Madry et al., 2018; Kolter & Wong, 2018; Wong et al., 2018; Raghunathan et al., 2018a;b). Most results in this direction involve algorithms that approximately minimize

$$\min_f \mathbb{E}\left\{\max_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)} \phi(f(\boldsymbol{X}')Y)\right\}, \tag{4}$$

where the objective function in problem (4) serves as an upper bound of the robust error $\mathcal{R}_{\rm rob}(f)$. In complex problem domains, however, this objective function might not be tight as an upper bound of the robust error, and may not capture the trade-off between natural and robust errors.

A related line of research is adversarial training by regularization (Kurakin et al., 2017; Ross & Doshi-Velez, 2017; Zheng et al., 2016). There are several key differences between the results in this paper and those of (Kurakin et al., 2017; Ross & Doshi-Velez, 2017; Zheng et al., 2016). Firstly, the optimization formulations are different. In the previous works, the regularization term either measures the "difference" between $f(\boldsymbol{X}')$ and Y (Kurakin et al., 2017), or its gradient (Ross & Doshi-Velez, 2017). In contrast, our regularization term measures the "difference" between $f(\boldsymbol{X})$ and $f(\boldsymbol{X}')$. While Zheng et al. (2016) generated the adversarial example $\boldsymbol{X}'$ by adding random Gaussian noise to $\boldsymbol{X}$, our method simulates the adversarial example by solving the inner maximization problem in Eqn. (3). Secondly, we note that the losses in (Kurakin et al., 2017; Ross & Doshi-Velez, 2017; Zheng et al., 2016) lack of theoretical guarantees. Our loss, with the presence of the second term in problem (3), makes our theoretical analysis significantly more subtle. Moreover, our algorithm takes the same computational resources as (Kurakin et al., 2017), which makes our method scalable to large-scale datasets. We defer the experimental comparisons of various regularization based methods to Table 5.

Algorithm 1 Adversarial training by TRADES

input Step sizes η_1 and η_2, batch size m, number of iterations K in inner optimization, network architecture parametrized by θ

output Robust network f_θ

1: Randomly initialize network f_θ, or initialize network with pre-trained configuration
2: **repeat**
3: Read mini-batch $B = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_m\}$ from training set
4: **for** $i = 1, ..., m$ (in parallel) **do**
5: $\boldsymbol{x}_i' \leftarrow \boldsymbol{x}_i + 0.001 \cdot \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\mathcal{N}(\mathbf{0}, \mathbf{I})$ is the Gaussian distribution with zero mean and identity variance
6: **for** $k = 1, ..., K$ **do**
7: $\boldsymbol{x}_i' \leftarrow \Pi_{\mathbb{B}(\boldsymbol{x}_i,\epsilon)}(\eta_1 \mathsf{sign}(\nabla_{\boldsymbol{x}_i'} \mathcal{L}(f_\theta(\boldsymbol{x}_i), f_\theta(\boldsymbol{x}_i'))) + \boldsymbol{x}_i')$, where Π is the projection operator
8: **end for**
9: **end for**
10: $\theta \leftarrow \theta - \eta_2 \sum_{i=1}^m \nabla_\theta [\mathcal{L}(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i) + \mathcal{L}(f_\theta(\boldsymbol{x}_i), f_\theta(\boldsymbol{x}_i'))/\lambda]/m$
11: **until** training converged

Heuristic algorithm. In response to the optimization formulation (3), we use two heuristics to achieve more general defenses: a) extending to multi-class problems by involving multi-class calibrated loss; b) approximately solving the minimax problem via alternating gradient descent. For multi-class problems, a surrogate loss is *calibrated* if minimizers of the surrogate risk are also minimizers of the 0-1 risk (Pires & Szepesvári, 2016). Examples of multi-class calibrated loss include cross-entropy loss. Algorithmically, we extend problem (3) to the case of multi-class classifications by replacing ϕ with a multi-class calibrated loss $\mathcal{L}(\cdot, \cdot)$:

$$\min_f \mathbb{E}\left\{\mathcal{L}(f(\boldsymbol{X}), \boldsymbol{Y}) + \max_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)} \mathcal{L}(f(\boldsymbol{X}), f(\boldsymbol{X}'))/\lambda\right\}, \tag{5}$$

where $f(\boldsymbol{X})$ is the output vector of learning model (with softmax operator in the top layer for the cross-entropy loss $\mathcal{L}(\cdot, \cdot)$), $\boldsymbol{Y}$ is the label-indicator vector, and $\lambda > 0$ is the regularization parameter. The pseudocode of adversarial training procedure, which aims at minimizing the empirical form of problem (5), is displayed in Algorithm 1.

The key ingredient of the algorithm is to approximately solve the linearization of inner maximization in problem (5) by the *projected gradient descent* (see Step 7). We note that $\boldsymbol{x}_i$ is a global minimizer with zero gradient to the objective function $g(\boldsymbol{x}') := \mathcal{L}(f(\boldsymbol{x}_i), f(\boldsymbol{x}'))$ in the inner problem. Therefore, we initialize $\boldsymbol{x}_i'$ by adding a small, random perturbation around $\boldsymbol{x}_i$ in Step 5 to start the inner optimizer.

Table 3. Theoretical verification on the optimality of Theorem 3.1.

λ	$\mathcal{A}_{\rm rob}(f)$ (%)	$\mathcal{R}_\phi(f)$	$\Delta = \Delta_{\rm RHS} - \Delta_{\rm LHS}$
2.0	99.43	0.0006728	0.006708
3.0	99.41	0.0004067	0.005914
4.0	99.37	0.0003746	0.006757
5.0	99.34	0.0003430	0.005860

More exhaustive approximations of the inner maximization problem in terms of either optimization formulations or solvers would lead to better defense performance.

5. Experimental Results

In this section, we verify the effectiveness of TRADES by numerical experiments. We denote by $\mathcal{A}_{\rm rob}(f) = 1 - \mathcal{R}_{\rm rob}(f)$ the robust accuracy, and by $\mathcal{A}_{\rm nat}(f) = 1 - \mathcal{R}_{\rm nat}(f)$ the natural accuracy on test dataset. We release our code and trained models at `https://github.com/yaodongyu/TRADES`.

5.1. Optimality of Theorem 3.1

We verify the tightness of the established upper bound in Theorem 3.1 for binary classification problem on MNIST dataset. The negative examples are '1' and the positive examples are '3'. Here we use a Convolutional Neural Network (CNN) with two convolutional layers, followed by two fully-connected layers. The output size of the last layer is 1. To learn the robust classifier, we minimize the regularized surrogate loss in Eqn. (3), and use the hinge loss in Table 2 as the surrogate loss ϕ, where the associated ψ-transform is $\psi(\theta) = \theta$.

To verify the tightness of our upper bound, we calculate the left hand side in Theorem 3.1, i.e.,

$$\Delta_{\rm LHS} = \mathcal{R}_{\rm rob}(f) - \mathcal{R}^*_{\rm nat},$$

and the right hand side, i.e.,

$$\Delta_{\rm RHS} = (\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \mathbb{E} \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} \phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda).$$

As we cannot have access to the unknown distribution $\mathcal{D}$, we approximate the above expectation terms by test dataset. We first use natural training method to train a classifier so as to approximately estimate $\mathcal{R}^*_{\rm nat}$ and $\mathcal{R}^*_\phi$, where we find that the naturally trained classifier can achieve natural error $\mathcal{R}^*_{\rm nat} = 0\%$, and loss value $\mathcal{R}^*_\phi = 0.0$ for the binary classification problem. Next, we optimize problem (3) to train a robust classifier f. We take perturbation $\epsilon = 0.1$, number of iterations $K = 20$ and run 30 epochs on the training dataset. Finally, to approximate the second term in $\Delta_{\rm RHS}$, we use FGSM^k (white-box) attack (a.k.a. PGD attack) (Kurakin et al., 2017) with 20 iterations to approximately calculate the worst-case perturbed data $\boldsymbol{X}'$.

The results in Table 3 show the tightness of our upper bound in Theorem 3.1. It shows that the differences between $\Delta_{\rm RHS}$ and $\Delta_{\rm LHS}$ under various λ's are very small.

5.2. Sensitivity of regularization hyperparameter λ

The regularization parameter λ is an important hyperparameter in our proposed method. We show how the regularization parameter affects the performance of our robust classifiers by numerical experiments on two datasets, MNIST and CIFAR10. For both datasets, we minimize the loss in Eqn. (5) to learn robust classifiers for multi-class problems, where we choose $\mathcal{L}$ as the cross-entropy loss.

MNIST setup. We use the CNN which has two convolutional layers, followed by two fully-connected layers. The output size of the last layer is 10. We set perturbation $\epsilon = 0.1$, perturbation step size $\eta_1 = 0.01$, number of iterations $K = 20$, learning rate $\eta_2 = 0.01$, batch size $m = 128$, and run 50 epochs on the training dataset. To evaluate the robust error, we apply FGSM^k (white-box) attack with 40 iterations and 0.005 step size. The results are in Table 4.

CIFAR10 setup. We apply ResNet-18 (He et al., 2016) for classification. The output size of the last layer is 10. We set perturbation $\epsilon = 0.031$, perturbation step size $\eta_1 = 0.007$, number of iterations $K = 10$, learning rate $\eta_2 = 0.1$, batch size $m = 128$, and run 100 epochs on the training dataset. To evaluate the robust error, we apply FGSM^k (white-box) attack with 20 iterations and the step size is 0.003. The results are in Table 4.

We observe that as the regularization parameter $1/\lambda$ increases, the natural accuracy $\mathcal{A}_{\rm nat}(f)$ decreases while the robust accuracy $\mathcal{A}_{\rm rob}(f)$ increases, which verifies our theory on the trade-off between robustness and accuracy. Note that for MNIST dataset, the natural accuracy does not decrease too much as the regularization term $1/\lambda$ increases, which is different from the results of CIFAR10. This is probably because the classification task for MNIST is easier. Meanwhile, our proposed method is not very sensitive to the choice of λ. Empirically, when we set the hyperparameter $1/\lambda$ in $[1, 10]$, our method is able to learn classifiers with both high robustness and high accuracy. We will set $1/\lambda$ as either 1 or 6 in the following experiments.

5.3. Adversarial defenses under various attacks

Previously, Athalye et al. (2018) showed that 7 defenses in ICLR 2018 which relied on obfuscated gradients may easily break down. In this section, we verify the effectiveness of our method with the same experimental setup under both white-box and black-box threat models.

MNIST setup. We use the CNN architecture in (Carlini & Wagner, 2017) with four convolutional layers, followed by three fully-connected layers. We set perturbation $\epsilon = 0.3$, perturbation step size $\eta_1 = 0.01$, number of iterations $K = 40$, learning rate $\eta_2 = 0.01$, batch size $m = 128$, and run 100 epochs on the training dataset.

CIFAR10 setup. We use the same neural network architecture as (Madry et al., 2018), i.e., the wide residual network

Table 4. Sensitivity of regularization hyperparameter λ on MNIST and CIFAR10 datasets.

$1/\lambda$	$\mathcal{A}_{\text{rob}}(f)$ (%) on MNIST	$\mathcal{A}_{\text{nat}}(f)$ (%) on MNIST	$\mathcal{A}_{\text{rob}}(f)$ (%) on CIFAR10	$\mathcal{A}_{\text{nat}}(f)$ (%) on CIFAR10
1.0	94.75 ± 0.0712	99.28 ± 0.0125	44.68 ± 0.3088	87.01 ± 0.2819
2.0	95.45 ± 0.0883	99.29 ± 0.0262	48.22 ± 0.0740	85.22 ± 0.0543
3.0	95.57 ± 0.0262	99.24 ± 0.0216	49.67 ± 0.3179	83.82 ± 0.4050
4.0	95.65 ± 0.0340	99.16 ± 0.0205	50.25 ± 0.1883	82.90 ± 0.2217
5.0	95.65 ± 0.1851	99.16 ± 0.0403	50.64 ± 0.3336	81.72 ± 0.0286

WRN-34-10 (Zagoruyko & Komodakis, 2016). We set perturbation $\epsilon = 0.031$, perturbation step size $\eta_1 = 0.007$, number of iterations $K = 10$, learning rate $\eta_2 = 0.1$, batch size $m = 128$, and run 100 epochs on the training dataset.

5.3.1. White-box attacks

We summarize our results in Table 5 together with the results from (Athalye et al., 2018). We also implement methods in (Zheng et al., 2016; Kurakin et al., 2017; Ross & Doshi-Velez, 2017) on the CIFAR10 dataset as they are also regularization based methods. For MNIST dataset, we apply FGSM^k (white-box) attack with 40 iterations and the step size is 0.01. For CIFAR10 dataset, we apply FGSM^k (white-box) attack with 20 iterations and the step size is 0.003, under which the defense model in (Madry et al., 2018) achieves 47.04% robust accuracy. Table 5 shows that our proposed defense method can significantly improve the robust accuracy of models, which is able to achieve robust accuracy as high as 56.61%. We also evaluate our robust model on MNIST dataset under the same threat model as in (Samangouei et al., 2018) (C&W white-box attack Carlini & Wagner (2017)), and the robust accuracy is 99.46%. See appendix for detailed information of models in Table 5.

5.3.2. Black-box attacks

We verify the robustness of our models under black-box attacks. We first train models without using adversarial training on the MNIST and CIFAR10 datasets. We use the same network architectures that are specified in the beginning of this section, i.e., the CNN architecture in (Carlini & Wagner, 2017) and the WRN-34-10 architecture in (Zagoruyko & Komodakis, 2016). We denote these models by naturally trained models (*Natural*). The accuracy of the naturally trained CNN model is 99.50% on the MNIST dataset. The accuracy of the naturally trained WRN-34-10 model is 95.29% on the CIFAR10 dataset. We also implement the method proposed in (Madry et al., 2018) on both datasets. We denote these models by Madry's models (*Madry*). The accuracy of Madry et al. (2018)'s CNN model is 99.36% on the MNIST dataset. The accuracy of Madry et al. (2018)'s WRN-34-10 model is 85.49% on the CIFAR10 dataset.

For both datasets, we use FGSM^k (black-box) method to attack various defense models. For MNIST dataset, we set perturbation $\epsilon = 0.3$ and apply FGSM^k (black-box) attack with 40 iterations and the step size is 0.01. For CIFAR10 dataset, we set $\epsilon = 0.031$ and apply FGSM^k (black-box) attack with 20 iterations and the step size is 0.003. Note that the setup is the same as the setup specified in Section 5.3.1. We summarize our results in Table 6 and Table 7. In both tables, we use two source models (noted in the parentheses) to generate adversarial perturbations: we compute the perturbation directions according to the gradients of the source models on the input images. It shows that our models are more robust against black-box attacks transfered from naturally trained models and Madry et al. (2018)'s models. Moreover, our models can generate stronger adversarial examples for black-box attacks compared with naturally trained models and Madry et al. (2018)'s models.

5.4. Case study: NeurIPS 2018 Adversarial Vision Challenge

Competition settings. In the adversarial competition, the adversarial attacks and defenses are under the black-box setting. The dataset in this competition is Tiny ImageNet, which consists of 550,000 data (with our data augmentation) and 200 classes. The robust models only return label predictions instead of explicit gradients and confidence scores. The task for robust models is to defend against adversarial examples that are generated by the top-5 submissions in the un-targeted attack track. The score for each defense model is evaluated by the smallest perturbation distance that makes the defense model fail to output correct labels.

Competition results. The methodology in this paper was applied to the competition, where our entry ranked the 1st place. We implemented our method to train ResNet models. We report the mean ℓ_2 perturbation distance of the top-6 entries in Figure 3. It shows that our method outperforms other approaches with a large margin. In particular, we surpass the runner-up submission by 11.41% in terms of mean ℓ_2 perturbation distance.

6. Conclusions

In this paper, we study the problem of adversarial defenses against structural perturbations around input data. We focus on the trade-off between robustness and accuracy, and show an upper bound on the gap between robust error and optimal natural error. Our result advances the state-of-the-art work and matches the lower bound in the worst-case scenario. The bounds motivate us to minimize a new form of regularized surrogate loss, TRADES, for adversarial training.

Table 5. Comparisons of TRADES with prior defense models under white-box attacks.

Defense	Defense type	Under which attack	Dataset	Distance	$\mathcal{A}_{\text{nat}}(f)$	$\mathcal{A}_{\text{rob}}(f)$
Buckman et al. (2018)	gradient mask	Athalye et al. (2018)	CIFAR10	0.031 (ℓ_∞)	-	0%
Ma et al. (2018)	gradient mask	Athalye et al. (2018)	CIFAR10	0.031 (ℓ_∞)	-	5%
Dhillon et al. (2018)	gradient mask	Athalye et al. (2018)	CIFAR10	0.031 (ℓ_∞)	-	0%
Song et al. (2018)	gradient mask	Athalye et al. (2018)	CIFAR10	0.031 (ℓ_∞)	-	9%
Na et al. (2017)	gradient mask	Athalye et al. (2018)	CIFAR10	0.015 (ℓ_∞)	-	15%
Wong et al. (2018)	robust opt.	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	27.07%	23.54%
Madry et al. (2018)	robust opt.	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	87.30%	**47.04%**
Zheng et al. (2016)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	94.64%	0.15%
Kurakin et al. (2017)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	85.25%	45.89%
Ross & Doshi-Velez (2017)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	95.34%	0%
TRADES ($1/\lambda = 1.0$)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	88.64%	49.14%
TRADES ($1/\lambda = 6.0$)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	84.92%	**56.61%**
TRADES ($1/\lambda = 1.0$)	regularization	DeepFool (ℓ_∞)	CIFAR10	0.031 (ℓ_∞)	88.64%	59.10%
TRADES ($1/\lambda = 6.0$)	regularization	DeepFool (ℓ_∞)	CIFAR10	0.031 (ℓ_∞)	84.92%	61.38%
TRADES ($1/\lambda = 1.0$)	regularization	LBFGSAttack	CIFAR10	0.031 (ℓ_∞)	88.64%	84.41%
TRADES ($1/\lambda = 6.0$)	regularization	LBFGSAttack	CIFAR10	0.031 (ℓ_∞)	84.92%	81.58%
TRADES ($1/\lambda = 1.0$)	regularization	MI-FGSM	CIFAR10	0.031 (ℓ_∞)	88.64%	51.26%
TRADES ($1/\lambda = 6.0$)	regularization	MI-FGSM	CIFAR10	0.031 (ℓ_∞)	84.92%	57.95%
TRADES ($1/\lambda = 1.0$)	regularization	C&W	CIFAR10	0.031 (ℓ_∞)	88.64%	84.03%
TRADES ($1/\lambda = 6.0$)	regularization	C&W	CIFAR10	0.031 (ℓ_∞)	84.92%	81.24%
Samangouei et al. (2018)	gradient mask	Athalye et al. (2018)	MNIST	0.005 (ℓ_2)	-	55%
Madry et al. (2018)	robust opt.	FGSM^{40} (PGD)	MNIST	0.3 (ℓ_∞)	99.36%	96.01%
TRADES ($1/\lambda = 6.0$)	regularization	FGSM^{40} (PGD)	MNIST	0.3 (ℓ_∞)	99.48%	96.07%
TRADES ($1/\lambda = 6.0$)	regularization	C&W	MNIST	0.005 (ℓ_2)	99.48%	99.46%

Table 6. Comparisons of TRADES with prior defenses under black-box FGSM^{40} attack on the MNIST dataset. The models inside parentheses are source models which provide gradients to adversarial attackers. We provide the average cross-entropy loss value $\mathcal{L}(f(\boldsymbol{X}), \boldsymbol{Y})$ of each defense model in the bracket. The defense model 'Madry' is the same model as in the antepenultimate line of Table 5. The defense model 'TRADES' is the same model as in the penultimate line of Table 5.

Defense Model	Robust Accuracy $\mathcal{A}_{\text{rob}}(f)$	
Madry	97.43% [0.0078484]	(Natural)
TRADES	**97.63%** [0.0075324]	(Natural)
Madry	97.38% [0.0084962]	(Ours)
TRADES	**97.66%** [0.0073532]	(Madry)

Table 7. Comparisons of TRADES with prior defenses under black-box FGSM^{20} attack on the CIFAR10 dataset. The models inside parentheses are source models which provide gradients to adversarial attackers. We provide the average cross-entropy loss value of each defense model in the bracket. The defense model 'Madry' is implemented based on (Madry et al., 2018), and the defense model 'TRADES' is the same model as in the 11th line of Table 5.

Defense Model	Robust Accuracy $\mathcal{A}_{\text{rob}}(f)$	
Madry	84.39% [0.0519784]	(Natural)
TRADES	**87.60%** [0.0380258]	(Natural)
Madry	66.00% [0.1252672]	(Ours)
TRADES	**70.14%** [0.0885364]	(Madry)

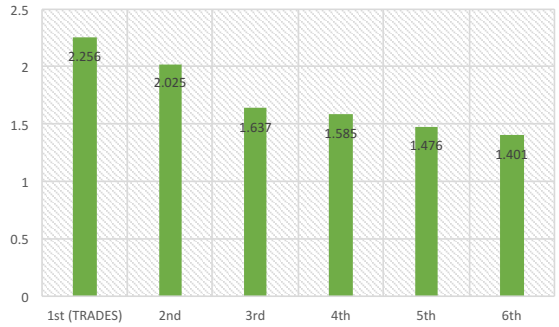


Figure 3. Top-6 results (out of ~2,000 submissions) in the NeurIPS 2018 Adversarial Vision Challenge. The vertical axis represents the mean ℓ_2 perturbation distance that makes robust models fail to output correct labels.

Experiments on real datasets and adversarial competition demonstrate the effectiveness of our proposed algorithms. It would be interesting to combine our methods with other related line of research on adversarial defenses, e.g., feature denoising technique (Xie et al., 2018) and network architecture design (Cisse et al., 2017), to achieve more robust learning systems.

Acknowledgements. We thank Maria-Florina Balcan and Avrim Blum for valuable discussions. Part of this work was done while H. Z. was visiting Simons Institute for the Theory of Computing, and Y. Y. was an intern at Petuum.

References

Alaifari, R., Alberti, G. S., and Gauksson, T. ADef: an iterative algorithm to construct adversarial deformations. In *International Conference on Learning Representations*, 2019.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, 2018.

Barthe, F. Extremal properties of central half-spaces for product measures. *Journal of Functional Analysis*, 182 (1):81–107, 2001.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018.

Brown, T. B., Carlini, N., Zhang, C., Olsson, C., Christiano, P., and Goodfellow, I. Unrestricted adversarial examples. *arXiv preprint arXiv:1809.08352*, 2018.

Bubeck, S., Lee, Y. T., Price, E., and Razenshteyn, I. Adversarial examples from cryptographic pseudo-random generators. *arXiv preprint arXiv:1811.06418*, 2018a.

Bubeck, S., Price, E., and Razenshteyn, I. Adversarial examples from computational constraints. *arXiv preprint arXiv:1805.10204*, 2018b.

Buckman, J., Roy, A., Raffel, C., and Goodfellow, I. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018.

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

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, 2017.

Cullina, D., Bhagoji, A. N., and Mittal, P. PAC-learning in the presence of adversaries. In *Advances in Neural Information Processing Systems*, pp. 228–239, 2018.

Dhillon, G. S., Azizzadenesheli, K., Lipton, Z. C., Bernstein, J., Kossaifi, J., Khanna, A., and Anandkumar, A. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., and Li, J. Boosting adversarial attacks with momentum. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9185–9193, 2018.

Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., and Madry, A. A rotation and a translation suffice: Fooling CNNs with simple transformations. *arXiv preprint arXiv:1712.02779*, 2017.

Engstrom, L., Ilyas, A., and Athalye, A. Evaluating and understanding the robustness of adversarial logit pairing. *arXiv preprint arXiv:1807.10272*, 2018.

Fawzi, A., Fawzi, H., and Fawzi, O. Adversarial vulnerability for any classifier. In *Advances in Neural Information Processing Systems*, pp. 1186–1195, 2018.

Gilmer, J., Adams, R. P., Goodfellow, I., Andersen, D., and Dahl, G. E. Motivating the rules of the game for adversarial example research. *arXiv preprint arXiv:1807.06732*, 2018.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, W., Wei, J., Chen, X., Carlini, N., and Song, D. Adversarial example defenses: Ensembles of weak defenses are not strong. *arXiv preprint arXiv:1706.04701*, 2017.

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

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

Jia, R. and Liang, P. Adversarial examples for evaluating reading comprehension systems. In *Empirical Methods in Natural Language Processing*, 2017.

Kannan, H., Kurakin, A., and Goodfellow, I. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

Kolter, J. Z. and Wong, E. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, 2018.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017.

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

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

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

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

Na, T., Ko, J. H., and Mukhopadhyay, S. Cascade adversarial machine learning regularized with a unified embedding. *arXiv preprint arXiv:1708.02582*, 2017.

Pires, B. Á. and Szepesvári, C. Multiclass classification calibration functions. *arXiv preprint arXiv:1609.06385*, 2016.

Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018a.

Raghunathan, A., Steinhardt, J., and Liang, P. S. Semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems*, pp. 10899–10909, 2018b.

Rauber, J., Brendel, W., and Bethge, M. Foolbox v0. 8.0: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*, 2017.

Ross, A. S. and Doshi-Velez, F. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. *arXiv preprint arXiv:1711.09404*, 2017.

Samangouei, P., Kabkab, M., and Chellappa, R. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Mądry, A. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems 31*, pp. 5019–5031, 2018.

Shaham, U., Yamada, Y., and Negahban, S. Understanding adversarial training: Increasing local stability of neural nets through robust optimization. *arXiv preprint arXiv:1511.05432*, 2015.

Sinha, A., Namkoong, H., and Duchi, J. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

Song, Y., Kim, T., Nowozin, S., Ermon, S., and Kushman, N. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018.

Su, D., Zhang, H., Chen, H., Yi, J., Chen, P.-Y., and Gao, Y. Is robustness the cost of accuracy? — a comprehensive study on the robustness of 18 deep image classification models. In *European Conference on Computer Vision*, 2018.

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

Tabacof, P. and Valle, E. Exploring the space of adversarial images. In *International Joint Conference on Neural Networks*, pp. 426–433, 2016.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

Uesato, J., O'Donoghue, B., Kohli, P., and van den Oord, A. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pp. 5025–5034, 2018.

Volpi, R., Namkoong, H., Sener, O., Duchi, J. C., Murino, V., and Savarese, S. Generalizing to unseen domains via adversarial data augmentation. In *Advances in Neural Information Processing Systems*, pp. 5339–5349, 2018.

Wong, E., Schmidt, F., Metzen, J., and Kolter, J. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems*, 2018.

Xiao, C., Zhu, J.-Y., Li, B., He, W., Liu, M., and Song, D. Spatially transformed adversarial examples. In *International Conference on Learning Representations*, 2018.

Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., and Yuille, A. Adversarial examples for semantic segmentation and object detection. In *International Conference on Computer Vision*, 2017.

Xie, C., Wu, Y., van der Maaten, L., Yuille, A., and He, K. Feature denoising for improving adversarial robustness. *arXiv preprint arXiv:1812.03411*, 2018.

Yin, D., Ramchandran, K., and Bartlett, P. Rademacher complexity for adversarially robust generalization. *arXiv preprint arXiv:1810.11914*, 2018.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *British Machine Vision Conference*, 2016.

Zhang, H., Xu, S., Jiao, J., Xie, P., Salakhutdinov, R., and Xing, E. P. Stackelberg GAN: Towards provable minimax equilibrium via multi-generator architectures. *arXiv preprint arXiv:1811.08010*, 2018.

Zhang, H., Chen, H., Song, Z., Boning, D., Dhillon, I. S., and Hsieh, C.-J. The limitations of adversarial training and the blind-spot attack. In *International Conference on Learning Representations*, 2019a.

Zhang, H., Shao, J., and Salakhutdinov, R. Deep neural networks with multi-branch architectures are intrinsically less non-convex. In *International Conference on Artificial Intelligence and Statistics*, pp. 1099–1109, 2019b.

Zhang, T. Covering number bounds of certain regularized linear function classes. *Journal of Machine Learning Research*, 2:527–550, 2002.

Zheng, S., Song, Y., Leung, T., and Goodfellow, I. Improving the robustness of deep neural networks via stability training. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4480–4488, 2016.

A. Other Related Works

Attack methods. Although deep neural networks have achieved great progress in various areas (Zhang et al., 2019b; 2018), they are brittle to adversarial attacks. Adversarial attacks have been extensively studied in the recent years. One of the baseline attacks to deep nerual networks is the *Fast Gradient Sign Method* (FGSM) (Goodfellow et al., 2015). FGSM computes an adversarial example as

$$\boldsymbol{x}' := \boldsymbol{x} + \epsilon \mathsf{sign}(\nabla_{\boldsymbol{x}} \phi(f(\boldsymbol{x})y)),$$

where $\boldsymbol{x}$ is the input instance, y is the label, $f : \mathcal{X} \to \mathbb{R}$ is the *score function* (parametrized by deep nerual network for example) which maps an instance to its confidence value of being positive, and $\phi(\cdot)$ is a surrogate of 0-1 loss. A more powerful yet natural extension of FGSM is the multi-step variant FGSM^k (also known as PGD attack) (Kurakin et al., 2017). FGSM^k applies *projected gradient descent* by k times:

$$\boldsymbol{x}'_{t+1} := \Pi_{\mathbb{B}(\boldsymbol{x},\epsilon)}(\boldsymbol{x}'_t + \epsilon \mathsf{sign}(\nabla_{\boldsymbol{x}} \phi(f(\boldsymbol{x}'_t)y))),$$

where $\boldsymbol{x}'_t$ is the t-th iteration of the algorithm with $\boldsymbol{x}'_0 := \boldsymbol{x}$ and $\Pi_{\mathbb{B}(\boldsymbol{x},\epsilon)}$ is the projection operator onto the ball $\mathbb{B}(\boldsymbol{x}, \epsilon)$. Both FGSM and FGSM^k are approximately solving (the linear approximation of) maximization problem:

$$\max_{\boldsymbol{x}' \in \mathbb{B}(\boldsymbol{x},\epsilon)} \phi(f(\boldsymbol{x}')y).$$

They can be adapted to the purpose of black-box attacks by running the algorithms on another similar network which is white-box to the algorithms (Tramèr et al., 2018). Though defenses that cause obfuscated gradients defeat iterative optimization based attacks, Athalye et al. (2018) showed that defenses relying on this effect can be circumvented. Other attack methods include MI-FGSM (Dong et al., 2018) and LBFGS attacks (Tabacof & Valle, 2016).

Robust optimization based defenses. Compared with attack methods, adversarial defense methods are relatively fewer. Robust optimization based defenses are inspired by the above-mentioned attacks. Intuitively, the methods train a network by fitting its parameters to the adversarial examples:

$$\min_f \mathbb{E} \left\{ \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X},\epsilon)} \phi(f(\boldsymbol{X}')Y) \right\}. \tag{6}$$

Following this framework, Huang et al. (2015); Shaham et al. (2015) considered one-step adversaries, while Madry et al. (2018) worked with multi-step methods for the inner maximization problem. There are, however, two critical differences between the robust optimization based defenses and the present paper. Firstly, robust optimization based defenses lack of theoretical guarantees. Secondly, such methods do not consider the trade-off between accuracy and robustness.

Relaxation based defenses. We mention another related line of research in adversarial defenses—relaxation based defenses. Given that the inner maximization in problem (6) might be hard to solve due to the non-convexity nature of deep neural networks, Kolter & Wong (2018) and Raghunathan et al. (2018a) considered a convex outer approximation of the set of activations reachable through a norm-bounded perturbation for one-hidden-layer neural networks. Wong et al. (2018) later scaled the methods to larger models, and Raghunathan et al. (2018b) proposed a tighter convex approximation. Sinha et al. (2018); Volpi et al. (2018) considered a Lagrangian penalty formulation of perturbing the underlying data distribution in a Wasserstein ball. These approaches, however, do not apply when the activation function is ReLU.

Theoretical progress. Despite a large amount of empirical works on adversarial defenses, many fundamental questions remain open in theory. There are a few preliminary explorations in recent years. Fawzi et al. (2018) derived upper bounds on the robustness to perturbations of any classification function, under the assumption that the data is generated with a smooth generative model. From computational aspects, Bubeck et al. (2018b;a) showed that adversarial examples in machine learning are likely not due to information-theoretic limitations, but rather it could be due to computational hardness. From statistical aspects, Schmidt et al. (2018) showed that the sample complexity of robust training can be significantly larger than that of standard training. This gap holds irrespective of the training algorithm or the model family. Cullina et al. (2018) and Yin et al. (2018) studied the uniform convergence of robust error $\mathcal{R}_{\mathrm{rob}}(f)$ by extending the classic VC and Rademacher arguments to the case of adversarial learning, respectively. A recent work demonstrates the existence of trade-off between accuracy and robustness (Tsipras et al., 2019). However, the work did not provide any methodology about how to tackle the trade-off.

Differences with Adversarial Logit Pairing. We also compare TRADES with Adversarial Logit Pairing (ALP) (Kannan et al., 2018; Engstrom et al., 2018). The algorithm of ALP works as follows: given a fixed network f in each round, the

algorithm firstly generates an adversarial example $\boldsymbol{X}'$ by solving $\operatorname{argmax}_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)}\phi(f(\boldsymbol{X}')Y)$; ALP then updates the network parameter by solving a minimization problem

$$\min_f \mathbb{E}\left\{\alpha\phi(f(\boldsymbol{X}')Y)+(1-\alpha)\phi(f(\boldsymbol{X})Y)+\|f(\boldsymbol{X})-f(\boldsymbol{X}')\|_2/\lambda\right\},$$

where $0\le\alpha\le 1$ is a regularization parameter; the algorithm finally repeats the above-mentioned procedure until it converges. We note that there are fundamental differences between TRADES and ALP. While ALP simulates adversarial example $\boldsymbol{X}'$ by the FGSM^k attack, TRADES simulates $\boldsymbol{X}'$ by solving $\operatorname{argmax}_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)}\phi(f(\boldsymbol{X})f(\boldsymbol{X}')/\lambda)$. Moreover, while ALP uses the ℓ_2 loss between $f(\boldsymbol{X})$ and $f(\boldsymbol{X}')$ to regularize the training procedure without theoretical guarantees, TRADES uses the classification-calibrated loss according to Theorems 3.1 and 3.2.

B. Proofs of Main Results

In this section, we provide the proofs of our main results.

B.1. Proof of Theorem 3.1

Theorem 3.1 (restated). *Let $\mathcal{R}_\phi(f):=\mathbb{E}\phi(f(\boldsymbol{X})Y)$ and $\mathcal{R}_\phi^*:=\min_f\mathcal{R}_\phi(f)$. Under Assumption 1, for any non-negative loss function ϕ such that $\phi(0)\ge 1$, any measurable $f:\mathcal{X}\to\mathbb{R}$, any probability distribution on $\mathcal{X}\times\{\pm1\}$, and any $\lambda>0$, we have*

$$\begin{aligned}\mathcal{R}_{\mathrm{rob}}(f)-\mathcal{R}_{\mathrm{nat}}^*&\le\psi^{-1}(\mathcal{R}_\phi(f)-\mathcal{R}_\phi^*)+\Pr[\boldsymbol{X}\in\mathbb{B}(\mathrm{DB}(f),\epsilon),f(\boldsymbol{X})Y>0]\\&\le\psi^{-1}(\mathcal{R}_\phi(f)-\mathcal{R}_\phi^*)+\mathbb{E}\max_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)}\phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda).\end{aligned}$$

Proof. By Eqn. (1), $\mathcal{R}_{\mathrm{rob}}(f)-\mathcal{R}_{\mathrm{nat}}^*=\mathcal{R}_{\mathrm{nat}}(f)-\mathcal{R}_{\mathrm{nat}}^*+\mathcal{R}_{\mathrm{bdy}}(f)\le\psi^{-1}(\mathcal{R}_\phi(f)-\mathcal{R}_\phi^*)+\mathcal{R}_{\mathrm{bdy}}(f)$, where the last inequality holds because we choose ϕ as a classification-calibrated loss (Bartlett et al., 2006). This leads to the first inequality.

Also, notice that

$$\begin{aligned}\Pr[\boldsymbol{X}\in\mathbb{B}(\mathrm{DB}(f),\epsilon),f(\boldsymbol{X})Y>0]&\le\Pr[\boldsymbol{X}\in\mathbb{B}(\mathrm{DB}(f),\epsilon)]\\&=\mathbb{E}\max_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)}\mathbf{1}\{f(\boldsymbol{X}')\ne f(\boldsymbol{X})\}\\&=\mathbb{E}\max_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)}\mathbf{1}\{f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda<0\}\\&\le\mathbb{E}\max_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)}\phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda).\end{aligned}$$

This leads to the second inequality. □

B.2. Proof of Theorem 3.2

Theorem 3.2 (restated). *Suppose that $|\mathcal{X}|\ge 2$. Under Assumption 1, for any non-negative loss function ϕ such that $\phi(x)\to 0$ as $x\to+\infty$, any $\xi>0$, and any $\theta\in[0,1]$, there exists a probability distribution on $\mathcal{X}\times\{\pm1\}$, a function $f:\mathbb{R}^d\to\mathbb{R}$, and a regularization parameter $\lambda>0$ such that $\mathcal{R}_{\mathrm{rob}}(f)-\mathcal{R}_{\mathrm{nat}}^*=\theta$ and*

$$\psi\Big(\theta-\mathbb{E}\max_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)}\phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda)\Big)\le\mathcal{R}_\phi(f)-\mathcal{R}_\phi^*\le\psi\left(\theta-\mathbb{E}\max_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)}\phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda)\right)+\xi.$$

Proof. The first inequality follows from Theorem 3.1. Thus it suffices to prove the second inequality.

Fix $\epsilon>0$ and $\theta\in[0,1]$. By the definition of ψ and its continuity, we can choose $\gamma,\alpha_1,\alpha_2\in[0,1]$ such that $\theta=\gamma\alpha_1+(1-\gamma)\alpha_2$ and $\psi(\theta)\ge\gamma\tilde{\psi}(\alpha_1)+(1-\gamma)\tilde{\psi}(\alpha_2)-\epsilon/3$. For two distinct points $\boldsymbol{x}_1,\boldsymbol{x}_2\in\mathcal{X}$, we set $\mathcal{P}_{\mathcal{X}}$ such

that $\Pr[\boldsymbol{X}=\boldsymbol{x}_1]=\gamma$, $\Pr[\boldsymbol{X}=\boldsymbol{x}_2]=1-\gamma$, $\eta(\boldsymbol{x}_1)=(1+\alpha_1)/2$, and $\eta(\boldsymbol{x}_2)=(1+\alpha_2)/2$. By the definition of H^-, we choose function $f:\mathbb{R}^d\to\mathbb{R}$ such that $f(\boldsymbol{x})<0$ for all $\boldsymbol{x}\in\mathcal{X}$, $C_{\eta(\boldsymbol{x}_1)}(f(\boldsymbol{x}_1))\le H^-(\eta(\boldsymbol{x}_1))+\epsilon/3$, and $C_{\eta(\boldsymbol{x}_2)}(f(\boldsymbol{x}_2))\le H^-(\eta(\boldsymbol{x}_2))+\epsilon/3$. By the continuity of ψ, there is an $\epsilon'>0$ such that $\psi(\theta)\le\psi(\theta-\epsilon_0)+\epsilon/3$ for all $0\le\epsilon_0<\epsilon'$. We also note that there exists an $\lambda_0>0$ such that for any $0<\lambda<\lambda_0$, we have

$$0\le\mathbb{E}\max_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)}\phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda)<\epsilon'.$$

Thus, we have

$$\begin{aligned}
\mathcal{R}_\phi(f)-\mathcal{R}_\phi^*&=\mathbb{E}\phi(Yf(\boldsymbol{X}))-\inf_f\mathbb{E}\phi(Yf(\boldsymbol{X}))\\
&=\gamma[C_{\eta(\boldsymbol{x}_1)}(f(\boldsymbol{x}_1))-H(\eta(\boldsymbol{x}_1))]+(1-\gamma)[C_{\eta(\boldsymbol{x}_2)}(f(\boldsymbol{x}_2))-H(\eta(\boldsymbol{x}_2))]\\
&\le\gamma[H^-(\eta(\boldsymbol{x}_1))-H(\eta(\boldsymbol{x}_1))]+(1-\gamma)[H^-(\eta(\boldsymbol{x}_2))-H(\eta(\boldsymbol{x}_2))]+\epsilon/3\\
&=\gamma\tilde{\psi}(\alpha_1)+(1-\gamma)\tilde{\psi}(\alpha_2)+\epsilon/3\\
&\le\psi(\theta)+2\epsilon/3\\
&\le\psi\left(\theta-\mathbb{E}\max_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)}\phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda)\right)+\epsilon.
\end{aligned}$$

Furthermore, by Lemma C.1,

$$\begin{aligned}
\mathcal{R}_{\mathrm{rob}}(f)-\mathcal{R}_{\mathrm{nat}}^*&=\mathbb{E}[\mathbf{1}\{\mathsf{sign}(f(\boldsymbol{X}))\neq\mathsf{sign}(f^*(\boldsymbol{X})),\boldsymbol{X}\in\mathbb{B}(\mathrm{DB}(f),\epsilon)^\perp\}|2\eta(\boldsymbol{X})-1|]\\
&\quad+\Pr[\boldsymbol{X}\in\mathbb{B}(\mathrm{DB}(f),\epsilon),\mathsf{sign}(f^*(\boldsymbol{X}))=Y]\\
&=\mathbb{E}|2\eta(\boldsymbol{X})-1|\\
&=\gamma(2\eta(\boldsymbol{x}_1)-1)+(1-\gamma)(2\eta(\boldsymbol{x}_2)-1)\\
&=\theta,
\end{aligned}$$

where f^* is the Bayes optimal classifier which outputs "positive" for all data points. □

C. Extra Theoretical Results

In this section, we provide extra theoretical results for adversarial defenses.

C.1. A lemma

We denote by $f^*(\cdot):=2\eta(\cdot)-1$ the Bayes decision rule throughout the proofs.

Lemma C.1. *For any classifier f, we have*

$$\begin{aligned}
\mathcal{R}_{\mathrm{rob}}(f)-\mathcal{R}_{\mathrm{nat}}^*=&\mathbb{E}[\mathbf{1}\{\mathsf{sign}(f(\boldsymbol{X}))\neq\mathsf{sign}(f^*(\boldsymbol{X})),\boldsymbol{X}\in\mathbb{B}(\mathrm{DB}(f),\epsilon)^\perp\}|2\eta(\boldsymbol{X})-1|]\\
&+\Pr[\boldsymbol{X}\in\mathbb{B}(\mathrm{DB}(f),\epsilon),\mathsf{sign}(f^*(\boldsymbol{X}))=Y].
\end{aligned}$$

Proof. For any classifier f, we have

$$\begin{aligned}
&\Pr(\exists\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)\text{ s.t. }\mathsf{sign}(f(\boldsymbol{X}'))\neq Y|\boldsymbol{X}=\boldsymbol{x})\\
=&\Pr(Y=1,\exists\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)\text{ s.t. }\mathsf{sign}(f(\boldsymbol{X}'))=-1|\boldsymbol{X}=\boldsymbol{x})\\
&+\Pr(Y=-1,\exists\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)\text{ s.t. }\mathsf{sign}(f(\boldsymbol{X}'))=1|\boldsymbol{X}=\boldsymbol{x})\\
=&\mathbb{E}[\mathbf{1}\{Y=1\}\mathbf{1}\{\exists\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)\text{ s.t. }\mathsf{sign}(f(\boldsymbol{X}'))=-1\}|\boldsymbol{X}=\boldsymbol{x}]\\
&+\mathbb{E}[\mathbf{1}\{Y=-1\}\mathbf{1}\{\exists\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)\text{ s.t. }\mathsf{sign}(f(\boldsymbol{X}'))=1\}|\boldsymbol{X}=\boldsymbol{x}]\\
=&\mathbf{1}\{\exists\boldsymbol{x}'\in\mathbb{B}(\boldsymbol{x},\epsilon)\text{ s.t. }\mathsf{sign}(f(\boldsymbol{x}'))=-1\}\mathbb{E}\mathbf{1}\{Y=1|\boldsymbol{X}=\boldsymbol{x}\}\\
&+\mathbf{1}\{\exists\boldsymbol{x}'\in\mathbb{B}(\boldsymbol{x},\epsilon)\text{ s.t. }\mathsf{sign}(f(\boldsymbol{x}'))=1\}\mathbb{E}\mathbf{1}\{Y=-1|\boldsymbol{X}=\boldsymbol{x}\}\\
=&\mathbf{1}\{\exists\boldsymbol{x}'\in\mathbb{B}(\boldsymbol{x},\epsilon)\text{ s.t. }\mathsf{sign}(f(\boldsymbol{x}'))=-1\}\eta(\boldsymbol{x})+\mathbf{1}\{\exists\boldsymbol{x}'\in\mathbb{B}(\boldsymbol{x},\epsilon)\text{ s.t. }\mathsf{sign}(f(\boldsymbol{x}'))=1\}(1-\eta(\boldsymbol{x}))\\
=&\begin{cases}1, & \boldsymbol{x}\in\mathbb{B}(\mathrm{DB}(f),\epsilon),\\ \mathbf{1}\{\mathsf{sign}(f(\boldsymbol{x}))=-1\}(2\eta(\boldsymbol{x})-1)+(1-\eta(\boldsymbol{x})), & \text{otherwise.}\end{cases}
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathcal{R}_{\rm rob}(f) &= \int_{\mathcal{X}} \Pr[\exists \boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X},\epsilon) \text{ s.t. } \mathsf{sign}(f(\boldsymbol{X}')) \neq Y | \boldsymbol{X}=\boldsymbol{x}] d\Pr_{\boldsymbol{X}}(\boldsymbol{x}) \\
&= \int_{\mathbb{B}(\mathrm{DB}(f),\epsilon)} \Pr[\exists \boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X},\epsilon) \text{ s.t. } \mathsf{sign}(f(\boldsymbol{X}')) \neq Y | \boldsymbol{X}=\boldsymbol{x}] d\Pr_{\boldsymbol{X}}(\boldsymbol{x}) \\
&\quad + \int_{\mathbb{B}(\mathrm{DB}(f),\epsilon)^{\perp}} \Pr[\exists \boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X},\epsilon) \text{ s.t. } \mathsf{sign}(f(\boldsymbol{X}')) \neq Y | \boldsymbol{X}=\boldsymbol{x}] d\Pr_{\boldsymbol{X}}(\boldsymbol{x}) \\
&= \Pr(\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon)) + \int_{\mathbb{B}(\mathrm{DB}(f),\epsilon)^{\perp}} [\mathbf{1}\{\mathsf{sign}(f(\boldsymbol{x}))=-1\}(2\eta(\boldsymbol{x})-1)+(1-\eta(\boldsymbol{x}))] d\Pr_{\boldsymbol{X}}(\boldsymbol{x}).
\end{aligned}$$

We have

$$\begin{aligned}
&\mathcal{R}_{\rm rob}(f) - \mathcal{R}_{\rm nat}(f^*) \\
&= \Pr(\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon)) + \int_{\mathbb{B}(\mathrm{DB}(f),\epsilon)^{\perp}} [\mathbf{1}\{\mathsf{sign}(f(\boldsymbol{x}))=-1\}(2\eta(\boldsymbol{x})-1)+(1-\eta(\boldsymbol{x}))] d\Pr_{\boldsymbol{X}}(\boldsymbol{x}) \\
&\quad - \int_{\mathbb{B}(\mathrm{DB}(f),\epsilon)^{\perp}} [\mathbf{1}\{\mathsf{sign}(f^*(\boldsymbol{x}))=-1\}(2\eta(\boldsymbol{x})-1)+(1-\eta(\boldsymbol{x}))] d\Pr_{\boldsymbol{X}}(\boldsymbol{x}) \\
&\quad - \int_{\mathbb{B}(\mathrm{DB}(f),\epsilon)} [\mathbf{1}\{\mathsf{sign}(f^*(\boldsymbol{x}))=-1\}(2\eta(\boldsymbol{x})-1)+(1-\eta(\boldsymbol{x}))] d\Pr_{\boldsymbol{X}}(\boldsymbol{x}) \\
&= \Pr(\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon)) - \int_{\mathbb{B}(\mathrm{DB}(f),\epsilon)} [\mathbf{1}\{\mathsf{sign}(f^*(\boldsymbol{x}))=-1\}(2\eta(\boldsymbol{x})-1)+(1-\eta(\boldsymbol{x}))] d\Pr_{\boldsymbol{X}}(\boldsymbol{x}) \\
&\quad + \mathbb{E}[\mathbf{1}\{\mathsf{sign}(f(\boldsymbol{X})) \neq \mathsf{sign}(\eta(\boldsymbol{X})-1/2), \boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon)^{\perp}\} |2\eta(\boldsymbol{X})-1|] \\
&= \Pr(\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon)) - \mathbb{E}[\mathbf{1}\{\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon)\} \min\{\eta(\boldsymbol{X}), 1-\eta(\boldsymbol{X})\}] \\
&\quad + \mathbb{E}[\mathbf{1}\{\mathsf{sign}(f(\boldsymbol{X})) \neq \mathsf{sign}(\eta(\boldsymbol{X})-1/2), \boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon)^{\perp}\} |2\eta(\boldsymbol{X})-1|] \\
&= \mathbb{E}[\mathbf{1}\{\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon)\} \max\{\eta(\boldsymbol{X}), 1-\eta(\boldsymbol{X})\}] \\
&\quad + \mathbb{E}[\mathbf{1}\{\mathsf{sign}(f(\boldsymbol{X})) \neq \mathsf{sign}(\eta(\boldsymbol{X})-1/2), \boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon)^{\perp}\} |2\eta(\boldsymbol{X})-1|] \\
&= \Pr[\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon), \mathsf{sign}(f^*(\boldsymbol{X})) = Y] \\
&\quad + \mathbb{E}[\mathbf{1}\{\mathsf{sign}(f(\boldsymbol{X})) \neq \mathsf{sign}(f^*(\boldsymbol{X})), \boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon)^{\perp}\} |2\eta(\boldsymbol{X})-1|].
\end{aligned}$$

□

C.2. Adversarial vulnerability under log-concave distributions

Theorem 3.1 states that for any classifier f, the value $\Pr[\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon)]$ characterizes the robustness of the classifier. In this section, we show that among all classifiers such that $\Pr[\mathsf{sign}(f(\boldsymbol{X}))=+1]=1/2$, linear classifier minimizes

$$\liminf_{\epsilon \to +0} \frac{\Pr[\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon)]}{\epsilon}, \tag{7}$$

provided that the marginal distribution over $\mathcal{X}$ is products of log-concave measures. A measure is *log-concave* if the logarithm of its density is a concave function. The class of log-concave measures contains many well-known (classes of) distributions as special cases, such as Gaussian and uniform measure over ball.

Our results are inspired by the isoperimetric inequality of log-concave distributions by the work of Barthe (2001). Intuitively, the isoperimetric problem consists in finding subsets of prescribed measure, such that its measure increases the less under enlargement. Our analysis leads to the following guarantee on the quantity (7).

Theorem C.2. *Let μ be an absolutely continuous log-concave probability measure on $\mathbb{R}$ with even density function and let $\mu^{\otimes d}$ be the products of μ with dimension d. Denote by $d\mu = e^{-M(x)}$, where $M : \mathbb{R} \to [0,\infty]$ is convex. Assume that $M(0)=0$. If $\sqrt{M(x)}$ is a convex function, then for every integer d and any classifier f with $\Pr[\mathsf{sign}(f(\boldsymbol{X}))=+1]=1/2$, we have*

$$\liminf_{\epsilon \to +0} \frac{\Pr_{\boldsymbol{X} \sim \mu^{\otimes d}}[\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon)]}{\epsilon} \geq c$$

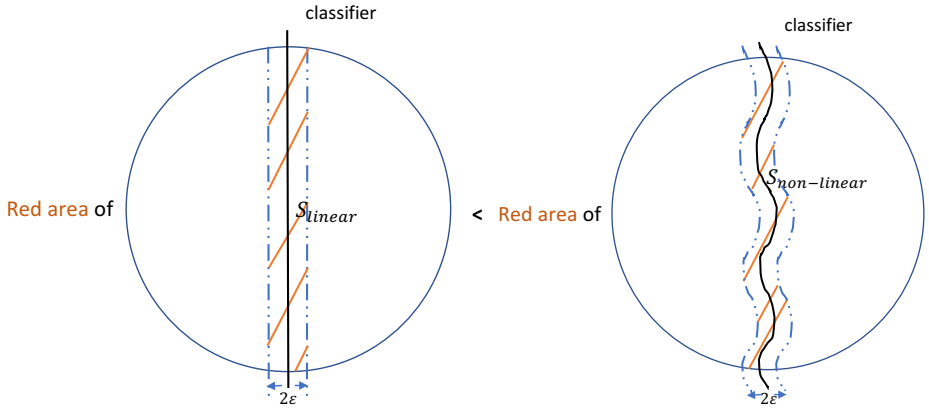


Figure 4. **Left figure:** boundary neighborhood of linear classifier. **Right figure:** boundary neighborhood of non-linear classifier. Theorem C.2 shows that the mass of S_{linear} is smaller than the mass of $S_{\text{non-linear}}$, provided that the underlying distribution over the instance space is the products of log-concave distribution on the real line.

for an absolute constant $c > 0$. Furthermore, among all such probability measures and classifiers, the linear classifier over products of Gaussian measure with mean 0 and variance $1/(2\pi)$ achieves the lower bound.

Theorem C.2 claims that under the products of log-concave distributions, the quantity $\Pr[\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f), \epsilon)]$ increases with rate at least $\Omega(\epsilon)$ for all classifier f, among which the linear classifier achieves the minimal value.

C.2.1. Proofs of Theorem C.2

For a Borel set $\mathcal{A}$ and for $\epsilon > 0$, denote by $\mathcal{A}_\epsilon = \{\boldsymbol{x} : d(\boldsymbol{x}, \mathcal{A}) \leq \epsilon\}$. The boundary measure of $\mathcal{A}$ is then defined as

$$\mu^+(\mathcal{A}) = \liminf_{\epsilon \to +0} \frac{\mu(\mathcal{A}_\epsilon) - \mu(\mathcal{A})}{\epsilon}.$$

The isoperimetric function is

$$I_\mu = \inf\{\mu^+(\mathcal{A}) : \mu(\mathcal{A}) = 1/2\}. \tag{8}$$

Before proceeding, we cite the following results from (Barthe, 2001).

Lemma C.3 (Theorem 9, (Barthe, 2001))**.** *Let μ be an absolutely continuous log-concave probability measure on $\mathbb{R}$ with even density function. Denote by $d\mu = e^{-M(x)}$, where $M : \mathbb{R} \to [0, \infty]$ is convex. Assume that $M(0) = 0$. If $\sqrt{M(x)}$ is a convex function, then for every integer d, we have $I_{\mu^{\otimes d}} \geq I_{\gamma^{\otimes d}}$, where γ is the Gaussian measure with mean 0 and variance $1/(2\pi)$. In particular, among sets of measure $1/2$ for $\mu^{\otimes d}$, the halfspace $[0, \infty) \times \mathbb{R}^{d-1}$ is solution to the isoperimetric problem (8).*

Now we are ready to prove Theorem C.2.

Proof. We note that

$$\begin{aligned}
&\Pr[\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f), \epsilon)] \\
&= \Pr[\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f), \epsilon), \mathsf{sign}(f(\boldsymbol{X})) = +1] + \Pr[\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f), \epsilon), \mathsf{sign}(f(\boldsymbol{X})) = -1].
\end{aligned}$$

To apply Lemma C.3, we set the $\mathcal{A}$ in Lemma C.3 as the event $\{\mathsf{sign}(f(\boldsymbol{X})) = +1\}$. Therefore, the set

$$\mathcal{A}_\epsilon = \{\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f), \epsilon), \mathsf{sign}(f(\boldsymbol{X})) = -1\}.$$

By Lemma C.3, we know that for linear classifier f_0 which represents the halfspace $[0, \infty) \times \mathbb{R}^{d-1}$, and any classifier f,

$$\begin{aligned}
&\liminf_{\epsilon \to +0} \frac{\Pr_{\boldsymbol{X} \sim \mu^{\otimes d}}[\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f), \epsilon), \mathsf{sign}(f(\boldsymbol{X})) = -1] - \Pr[\mathsf{sign}(f(\boldsymbol{X})) = +1]}{\epsilon} \\
&\geq \liminf_{\epsilon \to +0} \frac{\Pr_{\boldsymbol{X} \sim \gamma^{\otimes d}}[\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f_0), \epsilon), \mathsf{sign}(f_0(\boldsymbol{X})) = -1] - \Pr[\mathsf{sign}(f_0(\boldsymbol{X})) = +1]}{\epsilon}.
\end{aligned} \tag{9}$$

Similarly, we have

$$\begin{aligned}&\liminf_{\epsilon\to+0}\frac{\Pr_{\boldsymbol{X}\sim\mu^{\otimes d}}[\boldsymbol{X}\in\mathbb{B}(\mathrm{DB}(f),\epsilon),\mathsf{sign}(f(\boldsymbol{X}))=+1]-\Pr[\mathsf{sign}(f(\boldsymbol{X}))=-1]}{\epsilon}\\&\geq\liminf_{\epsilon\to+0}\frac{\Pr_{\boldsymbol{X}\sim\gamma^{\otimes d}}[\boldsymbol{X}\in\mathbb{B}(\mathrm{DB}(f_0),\epsilon),\mathsf{sign}(f_0(\boldsymbol{X}))=+1]-\Pr[\mathsf{sign}(f_0(\boldsymbol{X}))=-1]}{\epsilon}.\end{aligned}\tag{10}$$

Adding both sides of Eqns. (9) and (10), we have

$$\liminf_{\epsilon\to+0}\frac{\Pr_{\boldsymbol{X}\sim\mu^{\otimes d}}[\boldsymbol{X}\in\mathbb{B}(\mathrm{DB}(f),\epsilon)]}{\epsilon}\geq\liminf_{\epsilon\to+0}\frac{\Pr_{\boldsymbol{X}\sim\gamma^{\otimes d}}[\boldsymbol{X}\in\mathbb{B}(\mathrm{DB}(f_0),\epsilon)]}{\epsilon}\geq c.$$

□

C.3. Margin based generalization bounds

Before proceeding, we first cite a useful lemma. We say that function $f_1:\mathbb{R}\to\mathbb{R}$ and $f_2:\mathbb{R}\to\mathbb{R}$ have a γ separator if there exists a function $f_3:\mathbb{R}\to\mathbb{R}$ such that $|h_1-h_2|\leq\gamma$ implies $f_1(h_1)\leq f_3(h_2)\leq f_2(h_1)$. For any given function f_1 and $\gamma>0$, one can always construct f_2 and f_3 such that f_1 and f_2 have a γ-separator f_3 by setting $f_2(h)=\sup_{|h-h'|\leq2\gamma}f_1(h')$ and $f_3(h)=\sup_{|h-h'|\leq\gamma}f_1(h')$.

Lemma C.4 (Corollary 1, (Zhang, 2002))**.** *Let f_1 be a function $\mathbb{R}\to\mathbb{R}$. Consider a family of functions $f_2^\gamma:\mathbb{R}\to\mathbb{R}$, parametrized by γ, such that $0\leq f_1\leq f_2^\gamma\leq1$. Assume that for all γ, f_1 and f_2^γ has a γ separator. Assume also that $f_2^\gamma(z)\geq f_2^{\gamma'}(z)$ when $\gamma\geq\gamma'$. Let $\gamma_1>\gamma_2>...$ be a decreasing sequence of parameters, and p_i be a sequence of positive numbers such that $\sum_{i=1}^\infty p_i=1$, then for all $\eta>0$, with probability of at least $1-\delta$ over data:*

$$\mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}f_1(\mathcal{L}(\boldsymbol{w},\boldsymbol{X},Y))\leq\frac{1}{n}\sum_{i=1}^n f_2^\gamma(\mathcal{L}(\boldsymbol{w},\boldsymbol{x}_i,y_i))+\sqrt{\frac{32}{n}\left(\ln4\mathcal{N}_\infty(\mathcal{L},\gamma_i,\boldsymbol{x}_{1:n})+\ln\frac{1}{p_i\delta}\right)}$$

for all $\boldsymbol{w}$ and γ, where for each fixed γ, we use i to denote the smallest index such that $\gamma_i\leq\gamma$.

Lemma C.5 (Theorem 4, (Zhang, 2002))**.** *If $\|\boldsymbol{x}\|_p\leq b$ and $\|\boldsymbol{w}\|_q\leq a$, where $2\leq p<\infty$ and $1/p+1/q=1$, then $\forall\gamma>0$,*

$$\log_2\mathcal{N}_\infty(\mathcal{L},\gamma,n)\leq36(p-1)\frac{a^2b^2}{\gamma^2}\log_2[2\lceil4ab/\gamma+2\rceil+1].$$

Theorem C.6. *Suppose that the data is 2-norm bounded by $\|\boldsymbol{x}\|_2\leq b$. Consider the family Γ of linear classifier $\boldsymbol{w}$ with $\|\boldsymbol{w}\|_2=1$. Let $\mathcal{R}_{\mathrm{rob}}(\boldsymbol{w}):=\mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\mathbf{1}[\exists\boldsymbol{X}^{\mathrm{rob}}\in\mathbb{B}_2(\boldsymbol{X},\epsilon)$ such that $Y\boldsymbol{w}^T\boldsymbol{X}^{\mathrm{rob}}\leq0]$. Then with probability at least $1-\delta$ over n random samples $(\boldsymbol{x}_i,y_i)\sim\mathcal{D}$, for all margin width $\gamma>0$ and $\boldsymbol{w}\in\Gamma$, we have*

$$\mathcal{R}_{\mathrm{rob}}(\boldsymbol{w})\leq\frac{1}{n}\sum_{i=1}^n\mathbf{1}(\exists\boldsymbol{x}_i^{\mathrm{rob}}\in\mathbb{B}(\boldsymbol{x}_i,\epsilon)\ s.t.\ y_i\boldsymbol{w}^T\boldsymbol{x}_i^{\mathrm{rob}}\leq2\gamma)+\sqrt{\frac{C}{n}\left(\frac{b^2}{\gamma^2}\ln n+\ln\frac{1}{\delta}\right)}.$$

Proof. The theorem is a straightforward result of Lemmas C.4 and C.5 with

$$\mathcal{L}(\boldsymbol{w},\boldsymbol{x},y)=\min_{\boldsymbol{x}^{\mathrm{rob}}\in\mathbb{B}(\boldsymbol{x},\epsilon)}y\boldsymbol{w}^T\boldsymbol{x}^{\mathrm{rob}},$$

$$f_1(g)=\mathbf{1}(g\leq0)\quad\text{and}\quad f_2^\gamma(h)=\sup_{|g-h|<2\gamma}f_1(g)=f_1(g-2\gamma)=\mathbf{1}(g\leq2\gamma),$$

and $\gamma_i=b/2^i$ and $p_i=1/2^i$. □

We note that for the ℓ_2 ball $\mathbb{B}_2(\boldsymbol{x},\epsilon)=\{\boldsymbol{x}':\|\boldsymbol{x}-\boldsymbol{x}'\|_2\leq\epsilon\}$, we have

$$\mathbf{1}(\exists\boldsymbol{x}_i^{\mathrm{rob}}\in\mathbb{B}(\boldsymbol{x}_i,\epsilon)\text{ s.t. }y_i\boldsymbol{w}^T\boldsymbol{x}_i^{\mathrm{rob}}\leq2\gamma)=\max_{\boldsymbol{x}_i^{\mathrm{rob}}\in\mathbb{B}(\boldsymbol{x}_i,\epsilon)}\mathbf{1}(y_i\boldsymbol{w}^T\boldsymbol{x}_i^{\mathrm{rob}}\leq2\gamma)=\mathbf{1}(y_i\boldsymbol{w}^T\boldsymbol{x}_i\leq2\gamma+\epsilon).$$

Therefore, we can design the following algorithm—Algorithm 2.

Algorithm 2 Adversarial Training of Linear Separator via Structural Risk Minimization

Input: Samples $(\boldsymbol{x}_{1:n}, y_{1:n}) \sim \mathcal{D}$, a bunch of margin parameters $\gamma_1, ..., \gamma_T$.
1: For $k = 1, 2, ..., T$
2: Solve the minimax optimization problem:

$$\begin{aligned}\mathcal{L}_k(\boldsymbol{w}_k^*, \boldsymbol{x}_{1:n}, y_{1:n}) &= \min_{\boldsymbol{w}\in\mathbb{S}(\mathbf{0},1)} \frac{1}{n}\sum_{i=1}^{n} \max_{\boldsymbol{x}_i^{\text{rob}}\in\mathbb{B}(\boldsymbol{x}_i,\epsilon)} \mathbf{1}(y_i \boldsymbol{w}^T \boldsymbol{x}_i^{\text{rob}} \le 2\gamma_k) \\ &= \min_{\boldsymbol{w}\in\mathbb{S}(\mathbf{0},1)} \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(y_i \boldsymbol{w}^T \boldsymbol{x}_i \le 2\gamma_k + \epsilon).\end{aligned}$$

3: End For
4: $k^* = \text{argmin}_k \, \mathcal{L}_k(\boldsymbol{w}_k^*, \boldsymbol{x}_{1:n}, y_{1:n}) + \sqrt{\frac{C}{n}\left(\frac{b^2}{\gamma_k^2}\ln n + \ln\frac{1}{\delta}\right)}$.
Output: Hypothesis $\boldsymbol{w}_{k^*}$.

D. Extra Experimental Results

In this section, we provide extra experimental results to verify the effectiveness of our proposed method TRADES.

D.1. Experimental setup in Section 5.3.1

We use the same model, i.e., the WRN-34-10 architecture in Zagoruyko & Komodakis (2016), to implement the methods in Zheng et al. (2016), Kurakin et al. (2017) and Ross & Doshi-Velez (2017). The experimental setup is the same as TRADES, which is specified in the beginning of Section 5. For example, we use the same batch size and learning rate for all the methods. More specifically, we find that using one-step adversarial perturbation method like FGSM in the regularization term, defined in Kurakin et al. (2017), cannot defend against FGSM^k (white-box) attack. Therefore, we use FGSM^k with the cross-entropy loss to calculate the adversarial example $\boldsymbol{X}'$ in the regularization term, and the perturbation step size η_1 and number of iterations K are the same as in the beginning of Section 5.

As for defense models in Table 5, we implement the 'TRADES' models, the models trained by using other regularization losses in (Kurakin et al., 2017; Ross & Doshi-Velez, 2017; Zheng et al., 2016), and the defense model 'Madry' in the antepenultimate line of Table 5. We evaluate Wong et al. (2018)'s model based on the checkpoint provided by the authors. The rest of the models in Table 5 are reported in (Athalye et al., 2018).

D.2. Extra attack results in Section 5.3.1

Extra white-box attack results are provided in Table 8.

Table 8. Results of our method TRADES under different white-box attacks.

Defense	Under which attack	Dataset	Distance	$\mathcal{A}_{\text{nat}}(f)$	$\mathcal{A}_{\text{rob}}(f)$
TRADES ($1/\lambda = 1.0$)	FGSM	CIFAR10	0.031 (ℓ_∞)	88.64%	56.38%
TRADES ($1/\lambda = 1.0$)	FGSM^{1,000}	CIFAR10	0.031 (ℓ_∞)	88.64%	48.90%
TRADES ($1/\lambda = 1.0$)	DeepFool (ℓ_2)	CIFAR10	0.031 (ℓ_∞)	88.64%	84.49%
TRADES ($1/\lambda = 6.0$)	FGSM	CIFAR10	0.031 (ℓ_∞)	84.92%	61.06%
TRADES ($1/\lambda = 6.0$)	FGSM^{1,000}	CIFAR10	0.031 (ℓ_∞)	84.92%	56.43%
TRADES ($1/\lambda = 6.0$)	DeepFool (ℓ_2)	CIFAR10	0.031 (ℓ_∞)	84.92%	81.55%

The attacks in Table 5 and Table 8 include FGSM^k (Kurakin et al., 2017), DeepFool (ℓ_∞) (Moosavi-Dezfooli et al., 2016), LBFGSAttack (Tabacof & Valle, 2016), MI-FGSM (Dong et al., 2018), C&W (Carlini & Wagner, 2017), FGSM (Kurakin et al., 2017), and DeepFool (ℓ_2) (Moosavi-Dezfooli et al., 2016).

D.3. Extra attack results in Section 5.3.2

Extra black-box attack results are provided in Table 9 and Table 10. We apply black-box FGSM attack on the MNIST dataset and the CIFAR10 dataset.

Table 9. Comparisons of TRADES with prior defense models under black-box FGSM attack on the MNIST dataset. The models inside parentheses are source models which provide gradients to adversarial attackers.

Defense Model	Robust Accuracy $\mathcal{A}_{\text{rob}}(f)$	
Madry	97.68% (Natural)	98.11% (Ours)
TRADES	**97.75%** (Natural)	**98.44%** (Madry)

Table 10. Comparisons of TRADES with prior defense models under black-box FGSM attack on the CIFAR10 dataset. The models inside parentheses are source models which provide gradients to adversarial attackers.

Defense Model	Robust Accuracy $\mathcal{A}_{\text{rob}}(f)$	
Madry	84.02% (Natural)	67.66% (Ours)
TRADES	**86.84%** (Natural)	**71.52%** (Madry)

D.4. Experimental setup in Section 5.3.2

The robust accuracy of Madry et al. (2018)'s CNN model is 96.01% on the MNIST dataset. The robust accuracy of Madry et al. (2018)'s WRN-34-10 model is 47.66% on the CIFAR10 dataset. Note that we use the same white-box attack method introduced in the Section 5.3.1, i.e., FGSM^{20}, to evaluate the robust accuracies of Madry's models.

D.5. Interpretability of the robust models trained by TRADES

D.5.1. Adversarial examples on MNIST and CIFAR10 datasets

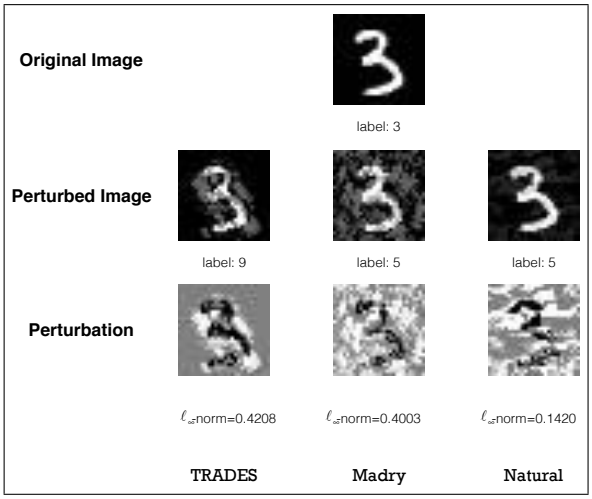
In this section, we provide adversarial examples on MNIST and CIFAR10. We apply **`foolbox`**[3] (Rauber et al., 2017) to generate adversarial examples, which is able to return the smallest adversarial perturbations under the ℓ_∞ norm distance. The adversarial examples are generated by using FGSM^k (white-box) attack on the models described in Section 5, including *Natural* models, *Madry*'s models and *TRADES* models. Note that the FGSM^k attack is `foolbox.attacks.LinfinityBasicIterativeAttack` in **`foolbox`**. See Figure 5 and Figure 6 for the adversarial examples of different models on MNIST and CIFAR10 datasets.

D.5.2. Adversarial examples on Bird-or-Bicycle dataset

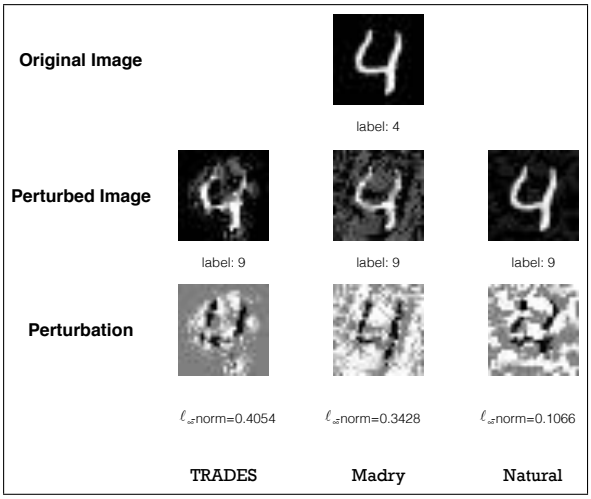
We find that the robust models trained by TRADES have strong interpretability. To see this, we apply a (spatial-tranformation-invariant) version of TRADES to train ResNet-50 models in response to the unrestricted adversarial examples of the bird-or-bicycle dataset (Brown et al., 2018). The dataset is *bird-or-bicycle*, which consists of 30,000 pixel-224×224 images with label either 'bird' or 'bicycle'. The unrestricted threat models include structural perturbations, rotations, translations, resizing, 17+ common corruptions, etc.

We show in Figures 7 and 8 the adversarial examples by the boundary attack with random spatial transformation on our robust model trained by the variant of TRADES. The boundary attack (Brendel et al., 2018) is a black-box attack method which searches for data points near the decision boundary and attack robust models by these data points. Therefore, the adversarial images obtained by boundary attack characterize the images around the decision boundary of robust models. We attack our model by boundary attack with random spatial transformations, a baseline in the competition. The classification accuracy on the adversarial test data is as high as 95% (at 80% coverage), even though the adversarial corruptions are perceptible to human. We observe that the robust model trained by TRADES has strong interpretability: in Figure 7 all of adversarial images have obvious feature of 'bird', while in Figure 8 all of adversarial images have obvious feature of 'bicycle'. This shows that images around the decision boundary of truly robust model have features of both classes.

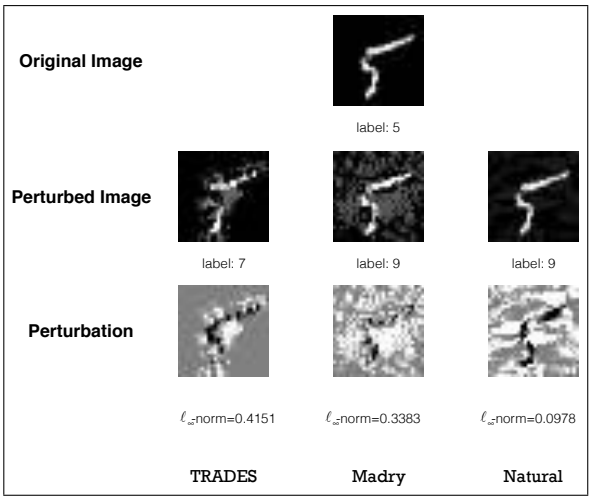
[3]Link: `https://foolbox.readthedocs.io/en/latest/index.html`



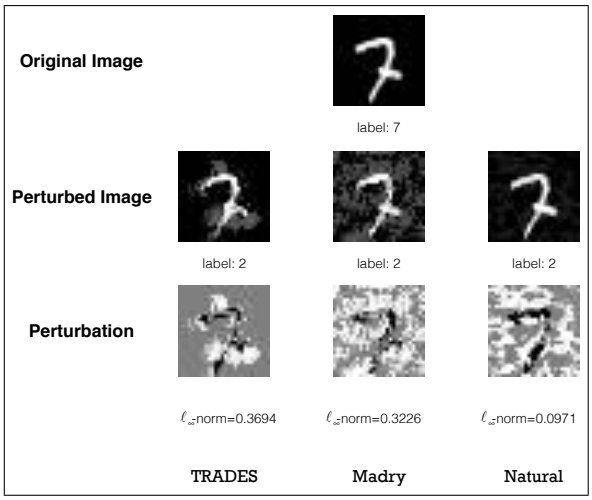
(a) adversarial examples of class '3'



(b) adversarial examples of class '4'

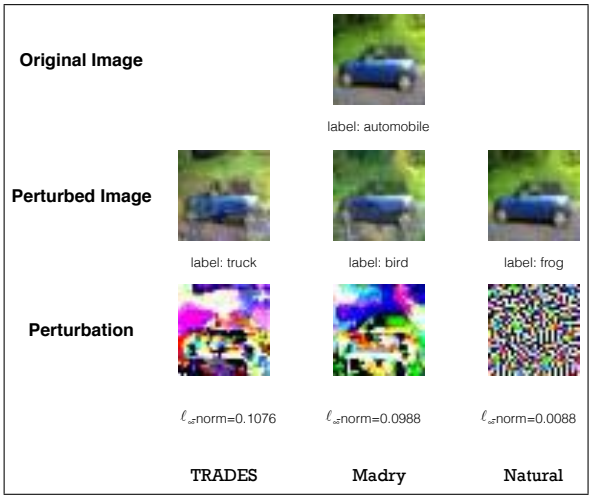


(c) adversarial examples of class '5'

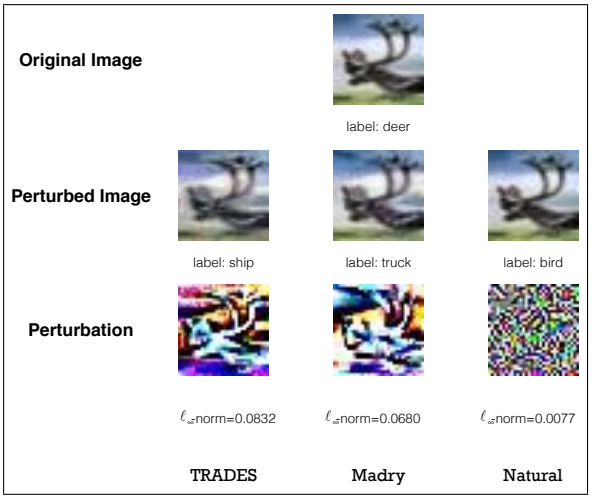


(d) adversarial examples of class '7'

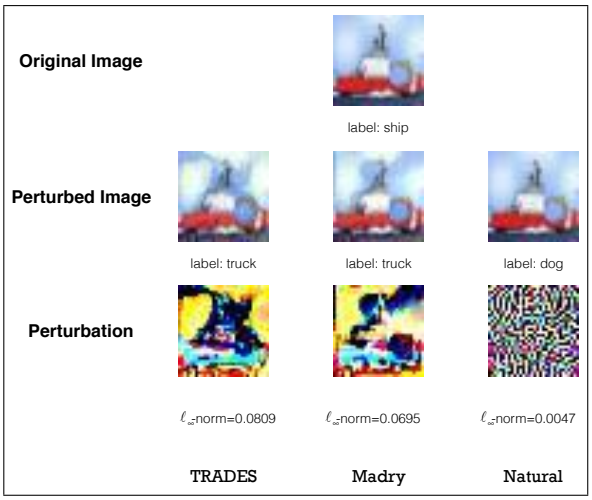
Figure 5. Adversarial examples on MNIST dataset. In each subfigure, the image in the first row is the original image and we list the corresponding correct label beneath the image. We show the perturbed images in the second row. The differences between the perturbed images and the original images, i.e., the perturbations, are shown in the third row. In each column, the perturbed image and the perturbation are generated by FGSM^k (white-box) attack on the model listed below. The labels beneath the perturbed images are the predictions of the corresponding models, which are different from the correct labels. We record the smallest perturbations in terms of ℓ_∞ norm that make the models predict a wrong label.



(a) adversarial examples of class 'automobile'



(b) adversarial examples of class 'deer'



(c) adversarial examples of class 'ship'

(d) adversarial examples of class 'truck'

Figure 6. Adversarial examples on CIFAR10 dataset. In each subfigure, the image in the first row is the original image and we list the corresponding correct label beneath the image. We show the perturbed images in the second row. The differences between the perturbed images and the original images, i.e., the perturbations, are shown in the third row. In each column, the perturbed image and the perturbation are generated by FGSM^k (white-box) attack on the model listed below. The labels beneath the perturbed images are the predictions of the corresponding models, which are different from the correct labels. We record the smallest perturbations in terms of ℓ_∞ norm that make the models predict a wrong label **(best viewed in color)**.

(a) clean example

(b) adversarial example by boundary attack with random spatial transformation

(c) clean example

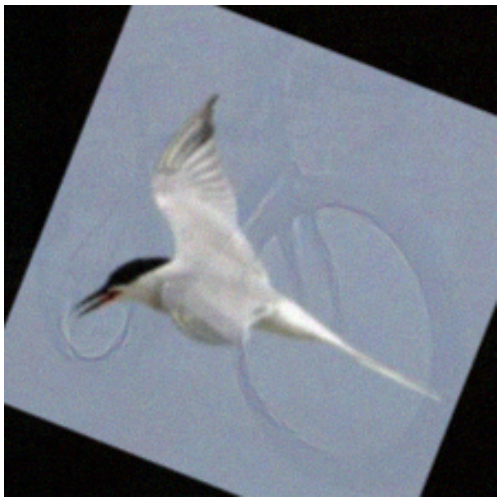
(d) adversarial example by boundary attack with random spatial transformation

(e) clean example

(f) adversarial example by boundary attack with random spatial transformation

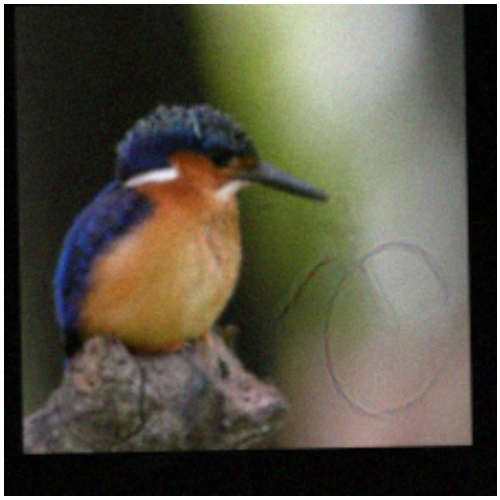
Figure 7. Adversarial examples by boundary attack with random spatial transformation on the ResNet-50 model trained by a variant of TRADES. The original label is 'bicycle', and our robust model recognizes the adversarial examples correctly as 'bicycle'. It shows in the second column that all of adversarial images have obvious feature of 'bird' **(best viewed in color)**.

(a) clean example



(b) adversarial example by boundary attack with random spatial transformation

(c) clean example



(d) adversarial example by boundary attack with random spatial transformation

(e) clean example

(f) adversarial example by boundary attack with random spatial transformation

Figure 8. Adversarial examples by boundary attack with random spatial transformation on the ResNet-50 model trained by a variant of TRADES. The original label is 'bird', and our robust model recognizes the adversarial examples correctly as 'bird'. It shows in the second column that all of adversarial images have obvious feature of 'bicycle' **(best viewed in color)**.