WAT: Improve the Worst-Class Robustness in Adversarial Training

Boqi Li, Weiwei Liu[*]

School of Computer Science, Wuhan University, China
{lbq988, liuweiwei863}@gmail.com

Abstract

Deep Neural Networks (DNN) have been shown to be vulnerable to adversarial examples. Adversarial training (AT) is a popular and effective strategy to defend against adversarial attacks. Recent works have shown that a robust model well-trained by AT exhibits a remarkable robustness disparity among classes, and propose various methods to obtain consistent robust accuracy across classes. Unfortunately, these methods sacrifice a good deal of the average robust accuracy. Accordingly, this paper proposes a novel framework of worst-class adversarial training and leverages no-regret dynamics to solve this problem. Our goal is to obtain a classifier with great performance on worst-class and sacrifice just a little average robust accuracy at the same time. We then rigorously analyze the theoretical properties of our proposed algorithm, and the generalization error bound in terms of the worst-class robust risk. Furthermore, we propose a measurement to evaluate the proposed method in terms of both the average and worst-class accuracies. Experiments on various datasets and networks show that our proposed method outperforms the state-of-the-art approaches.

Introduction

Deep Neural Networks (DNNs) are known to be vulnerable to adversarial examples (Szegedy et al. 2014; Goodfellow, Shlens, and Szegedy 2015). An adversarial example in a small perturbation from test data can easily fool the DNN model, which remains a security issue and is unacceptable in some applications of DNN, such as road sign classification (Eykholt et al. 2018) , text classification (Ebrahimi et al. 2018), self-supervised learning (Wang and Liu 2022) and object detection (Xu et al. 2020).

Numerous works (Raghunathan, Steinhardt, and Liang 2018; Madry et al. 2018; Li, Zou, and Liu 2022) have attempted to improve the model robustness with various defenses. Adversarial Training (AT) (Goodfellow, Shlens, and Szegedy 2015; Madry et al. 2018) is one of the most widely used and effective methods of defense. AT generates adversarial examples from the training data in every mini-batch, then uses these examples to replace training data or adds them into the training data during the training phase.

Although AT obtains great average adversarial robustness over classes, Benz et al. (2020); Xu et al. (2021); Tian et al. (2021) find that a robust model well-trained by AT exhibits a large robustness disparity in different classes on various balanced datasets, like the left classifier in Figure 1. Thus, AT leaves some classes vulnerable and may not perform well on some specific classes in certain real-world secure systems. For example, in the autonomous driving context, a classifier that has been well trained by AT may perform well on traffic sign classification and achieve great adversarial robustness performance on average while still exhibiting vulnerabilities on specific signs, which represents a potential danger for users.

Recently, some works (Benz et al. 2020; Xu et al. 2021) have attempted to solve this problem. Benz et al. (2020) analyze this phenomenon and use cost-sensitive learning to make the performance consistent over classes. Xu et al. (2021) propose employing re-weight and re-margin strategies to solve this problem. Both of these methods obtain consistent robust accuracy over classes, but they sacrifice a good deal of the average robust accuracy, like middle classifier in Figure 1. To overcome the limitations of Benz et al. (2020); Xu et al. (2021), this paper proposes a novel min-max learning paradigm to optimize worst-class robust risk and leverages no-regret dynamics to solve the proposed min-max problem, our goal is to achieve a classifier with great performance on worst-class but sacrifice a little average robust accuracy like the right classifier in Figure 1. Moreover, we rigorously analyze the theoretical properties of our proposed algorithm, and the generalization error bound in terms of the worst-class robust risk. Empirically, we find that a trade-off exists between average and worst-class robust accuracies, and accordingly propose a measurement to evaluate the method in terms of both the average and worst-class accuracies.

The main contributions in this paper are as follows:

- We propose a novel framework of worst-class adversarial training that leverages no-regret dynamics to solve the problem.
- We analyze the theoretical properties of our proposed algorithm, and the generalization error bound in terms of the worst-class robust risk.
- A measurement is presented to evaluate the method in terms of both the average and worst-class accuracies.

[*] Corresponding author.

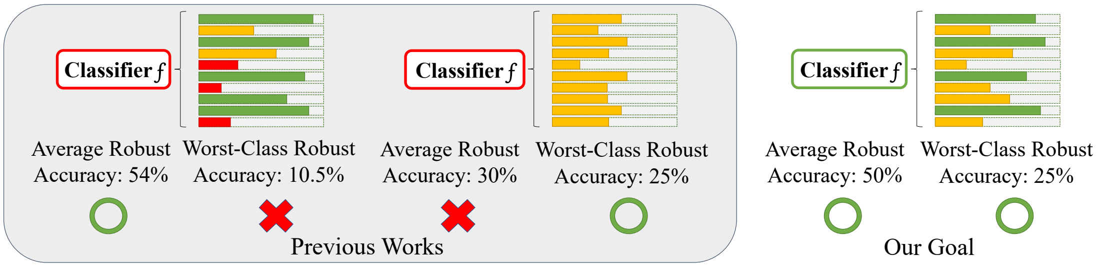

Classifier f
Average Robust Accuracy: 54%
Worst-Class Robust Accuracy: 10.5%
Classifier f
Average Robust Accuracy: 30%
Worst-Class Robust Accuracy: 25%
Previous Works
Classifier f
Average Robust Accuracy: 50%
Worst-Class Robust Accuracy: 25%
Our Goal

Figure 1: A brief introduction of our main idea. Previous works only care about average or worst-class robust accuracy, while our method considers both worst-class and average robust accuracy.

- Extensive experimental results on various datasets and networks verify that our proposed method outperforms state-of-the-art baselines.

Related Work

Adversarial Robustness. To improve adversarial robustness of DNN, adversarial training (Goodfellow, Shlens, and Szegedy 2015; Madry et al. 2018) is one of the most effective defenses. A large number of works (Zhang et al. 2019; Tsipras et al. 2019; Yang et al. 2020) have explored the trade-off between robustness and accuracy. Amongst them, TRADES (Zhang et al. 2019) is one of the most popular methods due to its promising experimental results. Besides, Ma, Wang, and Liu (2022) analyze the trade-off between robustness and fairness. Montasser, Hanneke, and Srebro (2019); Yin, Ramchandran, and Bartlett (2019); Xu and Liu (2022) theoretically analyze the adversarially robust generalization of a model while Simon-Gabriel et al. (2019) analyzes the first-order adversarial vulnerability of neural networks. Recently, a few works have been developed to further improve its performance, such as using unlabeled data (Carmon et al. 2019), feature alignments (Yan et al. 2021), wider networks (Wu et al. 2021) and a few tricks (Pang et al. 2021).

Disparity of Class-Wise Robustness. In natural training, class-imbalance is a classical problem in long-tailed data. In such problem, major class has more data than minor class. Most of previous works to solve this problem can be concluded as resampling (Zhou and Liu 2006) and cost-sensitive learning (Zou et al. 2018). Recently, some works have opted to focus on the class-wise robustness disparity in the adversarial training. Benz et al. (2020) study this problem empirically, and find that AT obtains a larger robust disparity among classes than that of natural training even in balanced data (e.g., CIFAR-10). Tian et al. (2021) also find the similar experimental results on six different datasets. To solve this problem, Benz et al. (2020) use a cost-sensitive learning fashion which is widely used in natural learning with imbalanced datasets; Xu et al. (2021) propose a new method to reduce the class-wise variance of robust accuracy over classes. However their approaches both sacrifice a good deal of the average robust accuracy because they aim to make the performance consistent over classes. To address this issue, this paper aims to improve the worst-class adversarial robustness, while obtaining less average robust accuracy loss than previous works.

Preliminaries

This paper considers a K-class classification problem over input space $\mathcal{X}$ and output label space $\mathcal{Y} = \{1, 2, \cdots, K\}$. Assume $\mathcal{D}$ is a distribution over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. We denote the sample as $\mathcal{S} : \{\mathcal{X} \times \mathcal{Y}\}^n$. Let $\mathcal{F}$ be the hypothesis class, while $f(\mathbf{x}; \theta) : \mathcal{X} \to \mathcal{Y}$ is a classifier in $\mathcal{F}$, where $\mathbf{x}$ is the input variable and f is parametrized by θ. Let $\ell : \mathcal{F} \times \mathcal{Z} \to [0, B]$ be the loss function. Throughout this paper, we assume that ℓ is bounded. The expected natural risk $\mathcal{R}_{nat}(f)$ and expected robust risk $\mathcal{R}_{rob}(f)$ over distribution $\mathcal{D}$ and classifier $f(\mathbf{x}; \theta)$ can then be defined with respect to loss function ℓ as follows:

$$\mathcal{R}^{nat}(f) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D} \ell(f(\mathbf{x}; \theta), y) \tag{1}$$

$$\mathcal{R}^{rob}(f) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D} \max_{\mathbf{x}' \in \mathcal{B}(\mathbf{x},\epsilon)} \ell(f(\mathbf{x}'; \theta), y) \tag{2}$$

where $\mathcal{B}(\mathbf{x}, \epsilon) = \{\mathbf{x}' : ||\mathbf{x}' - \mathbf{x}||_p \leq \epsilon\}$ denotes the ℓ_p-norm ($p \geq 1$) ball centered at $\mathbf{x}$ with radius ϵ.

Worst-Class Adversarial Robustness

Typically, one aims to use ERM to obtain a good classifier from a hypothesis class with low empirical risk. However, a classifier with low empirical risk may not perform well on the worst class. To illustrate this phenomenon, we present the results of different AT variants on the CIFAR-10 in Figure 2. From results in Figure 2(b), we can see that TRADES (Zhang et al. 2019) obtains a worst-class robust accuracy of 25.6% under PGD-20 (Madry et al. 2018) attack, while the average robust accuracy of TRADES is 51.94%. A similar phenomenon occurs when different variants of AT are used on different datasets. This degree of robustness disparity among classes is unacceptable in certain real-world secure systems. To study this problem, we define class-wise risk and worst-class risk as follows. We use $\mathcal{D}_k$ to denote the distribution of sample belonging to class k class, and $\mathcal{S}_k$ to denote the sample drawn from $\mathcal{D}_k$.

$$\mathcal{R}_k^{nat}(f) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim \mathcal{D}_k} [\ell(f(\mathbf{x}; \theta), y)] \tag{3}$$

$$\mathcal{R}_k^{rob}(f) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim \mathcal{D}_k} [\max_{\mathbf{x}' \in \mathcal{B}(\mathbf{x},\epsilon)} \ell(f(\mathbf{x}'; \theta), y)] \tag{4}$$

Similarly, we define the worst-class natural risk as $\mathcal{R}_{wc}^{nat}(f) = \max_{k\in[K]} \mathcal{R}_k^{nat}(f)$ and worst-class robust risk as $\mathcal{R}_{wc}^{rob}(f) = \max_{k\in[K]} \mathcal{R}_k^{rob}(f)$, where $[K]$ denotes the set of all positive integers in $[1, K]$. It follows that we have $\mathcal{R}_{wc}^{rob}(f) \geq \mathcal{R}^{rob}(f) \geq \mathcal{R}^{nat}(f)$.

Disparity of Adversarial Robustness

Figures 2(a) and 2(b) show that a large gap exists between the worst-class robust accuracy and the average robust accuracy. Therefore, a classifier with low expected natural risk and expected robust risk may have high robust risk on some classes.

To solve this problem, recently, various strategies (Benz et al. 2020; Xu et al. 2021) aimed at making the robust performance of the model consistent over all classes have been proposed. For example, Xu et al. (2021) propose the re-weight and re-margin strategies on TRADES. Empirically, these works show that existing strategies typically sacrifice the average robust accuracy to improve worst-class robust accuracy. It is hard to choose proper weight for each class.

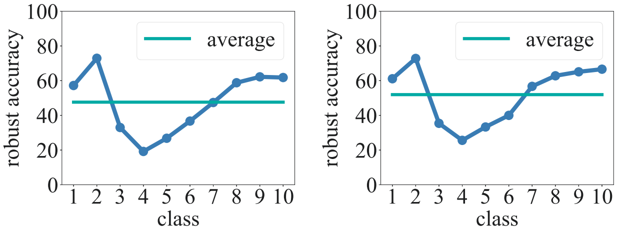
In Figure 3, we use TRADES to train a ResNet-18 (He et al. 2016) on CIFAR-10. We assign weight w_k for class-k and use a weighted loss $\sum_{k=1}^{K} w_k \ell_{trades}(\cdot,\cdot)$, where $\ell_{trades}(\cdot,\cdot)$ is the loss used in TRADES and is defined as $\ell_{trades} := \max_{\mathbf{x}' \in B(\mathbf{x},\epsilon)} CE(h_\theta(\mathbf{x}), y) + \beta KL(h_\theta(\mathbf{x}), h_\theta(\mathbf{x}'))$. We change the weight of class-4 from 0.05 to 0.25 and set the weights of the other classes to be $(1-w_4)/(K-1)$. In Figure 2(a), we find that the worst robust accuracy appears in class-4, so we choose to change the weights of class-4.

From the results in Figure 3, we can determine that when the weight of class-4 is increased from 0.05 to 0.15, the worst-class robust accuracy of TRADES grows by 23.1%, while the average robust accuracy of TRADES drops by 0.09%. Moreover, when the weight of class-4 is increased from 0.15 to 0.25, the worst-class and average robust accuracy drop at the same time. It is therefore demonstrably difficult to find the optimal weight for each class, and it is imperative to propose a measurement to simultaneously evaluate how much a given strategy would boost worst-class robust accuracy and decrease the average robust accuracy.

We use $\mathcal{A}$ to denote a vanilla adversarial training without any strategy, and $\mathcal{A}_\Delta$ to denote adversarial training with the strategy Δ. We run the algorithm $\mathcal{A}$ on hypothesis class $\mathcal{F}$ and sample S_{train}, and obtain the classifier $\hat{f} = \mathcal{A}(\mathcal{F}, \mathcal{S}_{train})$.

The average natural accuracy of a classifier f with respect to distribution $\mathcal{D}$ is defined as

$$Acc^{nat}(f,\mathcal{D}) = 1 - \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\{y \neq f(\theta,\mathbf{x})\} \tag{5}$$



(a) PGD on CIFAR-10

(b) TRADES on CIFAR-10

Figure 2: Class-wise robustness disparity of different AT using ResNet-18 on CIFAR-10. The robust accuracy (%) is evaluated under PGD-20 attack.

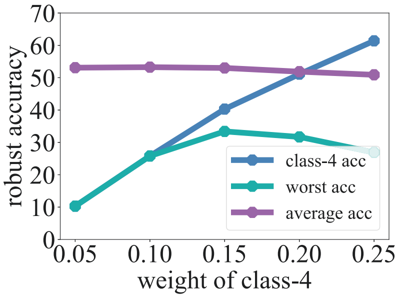


Figure 3: Trade-off between average and worst-class robust accuracy of ResNet-18 on CIFAR-10.

while average robust accuracy is defined as

$$Acc^{rob}(f,\mathcal{D}) = 1 - \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\{\exists \mathbf{x}' \in \mathcal{B}(\mathbf{x},\epsilon), \text{ s.t. } y \neq f(\theta,\mathbf{x}')\} \tag{6}$$

Similarly, we denote the k-th class natural accuracy as $Acc_k^{nat}(f,\mathcal{D})$, the worst-class natural accuracy as $Acc_{wc}^{nat}(f,\mathcal{D})$, the k-th class robust accuracy as $Acc_k^{rob}(f,\mathcal{D})$ and the worst-class robust accuracy as $Acc_{wc}^{rob}(f,\mathcal{D})$. Let the average robust accuracy, the accuracy of the k-th class and the worst-class accuracy of a classifier f on a test set $\mathcal{S}_{test}$ be $Acc^{rob}(f,\mathcal{S}_{test})$, $Acc_k(f,\mathcal{S}_{test})$ and $Acc_{wc}(f,\mathcal{S}_{test})$, respectively. For simplicity, we here use $\widehat{Acc}(f)$ to denote $Acc(f,\mathcal{S}_{test})$. This paper proposes a novel measurement to evaluate a method in terms of both the average and worst-class accuracy.

$$\begin{aligned}\hat{\rho}(\mathcal{F},\Delta,\mathcal{A},\mathcal{S}) = & \frac{\widehat{Acc}_{wc}(\mathcal{A}_\Delta(\mathcal{F})) - \widehat{Acc}_{wc}(\mathcal{A}(\mathcal{F}))}{\widehat{Acc}_{wc}(\mathcal{A}(\mathcal{F}))} \\ & - \frac{\widehat{Acc}(\mathcal{A}(\mathcal{F})) - \widehat{Acc}(\mathcal{A}_\Delta(\mathcal{F}))}{\widehat{Acc}(\mathcal{A}(\mathcal{F}))}\end{aligned} \tag{7}$$

Clearly, the larger the value of $\hat{\rho}$ is, the better a method performs.

Proposed Method

In this section, we formulate a novel min-max problem and then transform it into a two-player zero-sum game, and subsequently proposes a no-regret dynamics algorithm to solve the problem.

No-Regret Dynamics

Consider a two-player zero-sum game, in which a decision-maker repeatedly plays a game against an adversary. More specifically, the decision-maker plays before the adversary and does not know the action taken by the adversary in each round. No-regret dynamics is one of the most efficient methods of achieving an ϵ-coarse correlated equilibrium (Roughgarden and Iwama 2017).

Multiplicative Weight Updates Algorithm (Arora, Hazan, and Kale 2012) is one of the most widely used no-regret dynamic algorithms. Assume a game repeats for T rounds, while the decision-maker has a choice of n decisions. The decision-maker needs to repeatedly make a decision from the decision set and obtains an associated payoff from the adversary, while the best decision may not be known as a priori. Let $t = 1, 2, \cdots, T$ denote the current round. In each round t, the decision-maker produces a distribution $\mathbf{p^t}$ over the decision set and chooses an action from the set according to $\mathbf{p^t}$. At this time, the adversary chooses a cost vector $\mathbf{C^t}$. Let p_k^t be the k-th element of $\mathbf{p^t}$ while C_k^t denotes the k-th element of $\mathbf{C^t}$. Hedge Algorithm (Freund and Schapire 1997) is one of Multiplicative Weights Updates Algorithm that uses an exponential function to adjust the weight of every decision as follows.

$$p_k^t = \frac{\exp(\sum_{i=1}^{t-1} \eta C_k^i)}{\sum_{k=1}^{K} \exp(\sum_{i=1}^{t-1} \eta C_k^i)}. \tag{8}$$

Clearly, Hedge Algorithm produces the weights depending on past performance. Intuitively, this scheme works well because it tends to put heavy weights on high payoff decisions in the long run.

Worst-Class Adversarial Training

The loss of a classifier f on training set $\mathcal{S}_{tr}$ can be defined as

$$L_0^{tr}(f) = L^{tr}(f) = \frac{1}{|\mathcal{S}_{tr}|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S}_{tr}} \ell_{trades}(f(\mathbf{x}_i; \theta), y_i), \tag{9}$$

where $|\cdot|$ denotes the cardinality of a set. Let $L_k^{tr}(f)$ be the training loss on class k. Similarly, we use $L_0^{val}(f)$ and $L_k^{val}(f)$ to denote the loss of a classifier f on the validation set $\mathcal{S}_{val}$ and validation loss on class k, respectively. ℓ_{trades} is the loss used in TRADES.

We aim to minimize the following risk

$$\min_{f} \max_{k \in [0,K]} \mathcal{R}_k^{rob}(f), \tag{10}$$

where $\mathcal{R}_0^{rob}(f) = \mathcal{R}^{rob}(f)$. We then formulate (10) as a zero-sum game. In such a game, the learner has a decision set $\{\frac{\partial L_0^{tr}(f)}{\partial f}, \cdots, \frac{\partial L_K^{tr}(f)}{\partial f}\}$, $L_0^{tr}(f)$ is the excepted training loss and $L_k^{tr}(f)$ is the training loss of class-k for every $1 \leq k \leq K$. The best decision is not known as a priori.

Remark. *The reason that we add $\frac{\partial L_0^{tr}(f)}{\partial f}$ to decision set is the learner can directly choose $\frac{\partial L_0^{tr}(f)}{\partial f}$ as a decision in such a game.*

The weight of each decision is initialized as $1/(K + 1)$. In epoch t, we use the validation set to evaluate the classifier, and use validation loss to denote the cost. The learning rate is λ. In epoch t, the learner updates the model according to the following rule:

$$f^t = f^{t-1} - \lambda \sum_{k=0}^{K} w_k^t \frac{\partial L_k^{tr}(f^{t-1})}{\partial f}, \tag{11}$$

where

$$w_k^t = \frac{\exp(\sum_{i=1}^{t-1} \eta L_k^{val}(f^i))}{\sum_{k=0}^{K} \exp(\sum_{i=1}^{t-1} \eta L_k^{val}(f^i))}. \tag{12}$$

After the learner updates the model, it obtains a loss vector from the adversary. The algorithm is described in more detail in Algorithm 1. Algorithm 1 outputs $f^* = \arg\max_{k\in[K]} \min_{f\in\{f^1,\cdots,f^T\}} L_k^{val}(f)$. The following theorem provides the guarantee of the worst-class loss.

Algorithm 1: WAT: Worst-Class Adversarial Training

```
Input: training data $\mathcal{S}_{tr}$, validation data $\mathcal{S}_{val}$, learning rate $\lambda$, training epochs $T$, number of classes $K$ and hyper-parameter $\eta$.
Initialize $f^0, w_k^0 = \frac{1}{K+1}$ for every $k \in [K]$.
for $1 \leq t \leq T$ do
    use $\mathcal{S}_{tr}$to obtain $L_0^{tr}(f^{t-1}), \cdots, L_K^{tr}(f^{t-1})$.
    use $\mathcal{S}_{val}$ to obtain $L_0^{val}(f^{t-1}), \cdots, L_K^{val}(f^{t-1})$.
    $f^t = f^{t-1} - \lambda \sum_{k=0}^{K} w_k^t \frac{\partial L_k^{tr}(f^{t-1})}{\partial f}$
    for $0 \leq k \leq K$ do
        $w_k^{t+1} = \frac{\exp(\sum_{i=1}^{t} \eta L_k^{val}(f^i))}{\sum_{k=0}^{K} \exp(\sum_{i=1}^{t} \eta L_k^{val}(f^i))}$.
    end for
end for
Output: $f^* = \arg\max_{k\in[K]} \min_{f\in\{f^1,\cdots,f^T\}} L_k^{val}(f)$.
```

Theorem 1. *Assume the range of $L^{val}(f)$ is $[0, 1]$, and $1/T \sum_{t=1}^{T} L_k^{val}(f^t) \geq 1/(1-\eta) \min_t L_k^{val}(f^t)$ for every k and some $\eta \leq 1/2$. We then have*

$$\max_k \min_t L_k^{val}(f^t) \leq \frac{1}{T} \sum_{t=1}^{T} \sum_{k=0}^{K} w_k^t L_k^{val}(f^t) + \frac{\log(K+1)}{T\eta}. \tag{13}$$

Proof. The proof of Theorem 1 can be found in supplementary materials. □

Remark. *Theorem 1 shows that if we choose a proper η, after T rounds, the worst-class cost of the best classifier can be bounded by the average loss of previous rounds. Our bound also depends on η and T; a larger η and T will provide a tighter bound.*

Generalization Error Bound

This section provides the generalization error bound in terms of the worst-class robust risk. The empirical natural risk and robust risk are defined as $\hat{\mathcal{R}}^{nat}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(\mathbf{x}_i; \theta), y_i)$ and $\hat{\mathcal{R}}^{rob}(f) = \frac{1}{n} \sum_{i=1}^{n} \max_{\mathbf{x}' \in \mathcal{B}(\mathbf{x}, \epsilon)} \ell(f(\mathbf{x}'; \theta), y_i)$, respectively.

Rademacher complexity (Bartlett and Mendelson 2002) is one of the classic measurements for generalization error. Let $\mathcal{S} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n\}$ be an independent and identically distributed (i.i.d.) sample with size n and σ_i be a random variable such that $\mathbb{P}[\sigma_i = 1] = \mathbb{P}[\sigma_i = -1] = 1/2$. The Rademacher complexity of function class $\mathcal{H}$ is defined as

$$\mathfrak{R}_{\mathcal{S}}(\mathcal{H}) := \frac{1}{n} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{h \in \mathcal{H}} \sum_{i=1}^{n} \sigma_i h(\mathbf{z}_i) \right]. \tag{14}$$

We next analyze the gap between the empirical risk and population risk of the worst class. Let the training set S_k

be drawn i.i.d. from the distribution $\mathcal{D}_k$. The empirical k-th class robust risk is defined as

$$\hat{\mathcal{R}}_k^{rob}(f) = \frac{1}{|\mathcal{S}_k|} \sum_{(\mathbf{x_i}, y_i) \in \mathcal{S}_k} \max_{\mathbf{x}' \in \mathcal{B}(\mathbf{x}, \epsilon)} \ell(f(\mathbf{x_i'}; \theta), y_i). \quad (15)$$

The empirical worst-class robust risk over $\mathcal{S} := \cup_{k \in [K]} \mathcal{S}_k$ is $\hat{\mathcal{R}}_{wc}^{rob}(f) = \max_k \hat{\mathcal{R}}_k^{rob}(f)$. and $\tilde{\ell}_{\mathcal{F}}$ is defined as $\ell_{\mathcal{F}} = \{(\mathbf{x}, y) \to \ell(f(\mathbf{x}), y) : f \in \mathcal{F}\}$. We assume $|\mathcal{S}_k| = |\mathcal{S}|/K$ holds for every k. We present the following Theorem.

Theorem 2. *Suppose that the range of $\ell(f(\mathbf{x}), y)$ is $[0, B]$. Let $\tilde{\ell}(f(\mathbf{x}), y) := \max_{\mathbf{x}' \in \mathcal{B}(\mathbf{x}, \epsilon)} \ell(f(\mathbf{x}'), y)$. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the following holds for all $f \in \mathcal{F}$,*

$$\mathcal{R}_{wc}^{rob}(f) \leq \hat{\mathcal{R}}_{wc}^{rob}(f) + 2B \max_k \mathfrak{R}_{\mathcal{S}_k}(\tilde{\ell}_{\mathcal{F}}) + 3B\sqrt{\frac{K \log \frac{2}{\delta}}{2|\mathcal{S}|}}.$$

Proof. The proof of Theorem 2 can be found in supplementary materials. □

Multi-Class Linear Classifiers

This section studies the generalization error of multi-class linear classifiers. We here consider a K-class classification problem. Let $\mathcal{F}_{\mathbf{W}}$ be a multi-class linear classifier hypothesis, and $f_{\mathbf{W}} : X \to \mathbb{R}^K$ in $\mathcal{F}_{\mathbf{W}}$ be parameterized by a matrix $\mathbf{W}$ with dimension $K \times d$. The k-th coordinate of $f_{\mathbf{W}}(\mathbf{x})$ is the score of the k-th class, and the prediction of $f_{\mathbf{W}}$ is the class with the highest score among the K classes. Let $\mathbf{w}_k \in \mathbb{R}^d$ be the k-th column of $\mathbf{W}^\top$ and be upper bounded by W under the ℓ_p norm ($p \geq 1$): $\mathcal{F}_{\mathbf{W}} = \{f_{\mathbf{W}}(\mathbf{x}) : ||\mathbf{W}^\top||_{p,\infty} \leq W\}$. For multi-class classification problems, we define the margin operator $\mathcal{M}(\boldsymbol{\xi}, y) : \mathbb{R}^K \times [K] \to \mathbb{R}$ as $\mathcal{M}(\boldsymbol{\xi}, y) = \xi_y - \max_{y' \neq y} \xi_{y'}$, and a classifier$f$ predicts correct if and only if $\mathcal{M}(\boldsymbol{\xi}, y) > 0$. The ramp loss is defined as follows:

$$\phi_\gamma(t) = \begin{cases} 1 & t \leq 0, \\ 1 - \frac{t}{\gamma} & 0 < t < \gamma, \\ 0 & t \geq \gamma. \end{cases} \quad (16)$$

Based on the margin operator and ramp loss, we have $\ell(f_{\mathbf{W}}(\mathbf{x}), y) = \phi_\gamma(\mathcal{M}(f_{\mathbf{W}}(\mathbf{x}), y))$ and $\tilde{\ell}(f_{\mathbf{W}}(\mathbf{x}), y) = \max\limits_{\mathbf{x}' \in \mathcal{B}(\mathbf{x}, \epsilon)} \phi_\gamma(\mathcal{M}(\mathbf{f_W}(\mathbf{x}), y))$. We use $\mathbb{1}(\cdot)$ to denote a {0,1}-valued indicator function, and present the following Theorem.

Theorem 3. *Consider the multi-class linear classifiers in the adversarial setting, and suppose that $\frac{1}{p} + \frac{1}{q} = 1$, $p, q \geq 1$. For any fixed $\gamma > 0$ and $W > 0$, we have with probability at least $1 - \delta$, for all $\mathbf{W}$ such that $\|\mathbf{W}^\top\|_{p,\infty} \leq W$,*

$$1 - Acc_{wc}^{rob}(f, \mathcal{D}) \leq \frac{K}{|\mathcal{S}|} \sum_{(x_i, y_i) \in \mathcal{S}} E_i + \frac{2WK^3}{\gamma|\mathcal{S}|} U + c,$$

where

$$c = \frac{2WK^2 \epsilon d^{\frac{1}{q}}}{\gamma\sqrt{|\mathcal{S}|}} + 3\sqrt{\frac{K \log \frac{2}{\delta}}{2|\mathcal{S}|}},$$

$$U = \max_{y,k} \mathbb{E}_{\boldsymbol{\sigma}} \left[\left\| \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S}_k} \sigma_i \mathbf{x}_i \mathbb{1}(y_i = y) \right\|_q \right],$$

$$E_i = \mathbb{1}\left(\langle \mathbf{w}_{y_i}, \mathbf{x}_i \rangle \leq \gamma + \max_{y' \neq y_i} (\langle \mathbf{w}_{y'}, \mathbf{x}_i \rangle + \epsilon \|\mathbf{w}_{y'} - \mathbf{w}_{y_i}\|_1) \right).$$

Proof. The proof of Theorem 3 can be found in supplementary materials. □

Remark. *Only if we optimize worst-class robust risk, as in our method, Theorem 2 and 3 hold. However, previous works do not optimize this risk and Theorem 2 and 3 are not applicable to them.*

Experiments

In this section, we conduct experiments on various datasets and models to evaluate the performance of our proposed method. Code is available at https://github.com/boqili/WAT.

Datasets and Baselines

The datasets used in the experiments are CIFAR-10 and CIFAR-100 (Krizhevsky, Hinton et al. 2009), which are described in more detail in supplementary materials.

TRADES (Zhang et al. 2019), **FRL** (Xu et al. 2021) and **Cost-sensitive Learning (CSL)** (Benz et al. 2020) are used as our baselines. **TRADES** is one of the most popular adversarial training methods. **FRL** has two variants: **FRL-RW** is based on the re-weight strategy, and **FRL-RWRM** is based on the re-weight and re-margin strategy. **CSL** is a classical approach to solving the class-imbalanced problem on imbalanced datasets (Ting 2000; Khan et al. 2018). To be fair, we use the same hyper-parameters and perform the model selection for each method.

Evaluations

We use the following measures to evaluate the performance of all methods.

Average and Worst-class accuracy. Following (Xu et al. 2021), we use average natural accuracy, average robust accuracy, worst-class natural accuracy and worst-class robust accuracy to evaluate the performance of all methods. We use three strong adversarial attacks PGD-100, CW(Carlini and Wagner 2017) attack and AutoAttack(Croce and Hein 2020) to evaluate robust accuracy. We set perturbation radius $\epsilon = 8/255$ for CIFAR-10 and CIFAR-100. Other details can be found in supplementary materials.

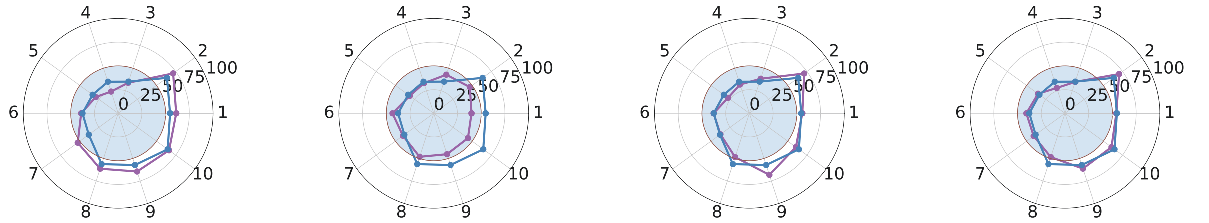
Class-wise Variance (CV). Class-wise variance is a common measure used in (Xu et al. 2021) and (Tian et al. 2021). The definition of CV given in (Tian et al. 2021) is presented below.

Definition 1. *(Tian et al. 2021) Given one dataset containing C classes, the accuracy of each class c is a_c, the average accuracy over all class is $\bar{a} = (\sum_{c=1}^{C} a_c)/C$, and the CV is defined as: $CV = (\sum_{c=1}^{C} (a_c - \bar{a})^2)/C$.*

CIFAR-10	Natural			PGD-100			CW			AutoAttack		
Method	Avg.	Wst.	ρ_{nat}	Avg.	Wst.	ρ_{pgd}	Avg.	Wst.	ρ_{cw}	Avg.	Wst.	ρ_{AA}
TRADES	**82.11**	64.6	0	**51.69**	25.2	0	50.38	24.1	0	**48.64**	21.7	0
FRL-RW	81.75	69.2	**0.067**	49.02	30.8	0.171	47.80	27.8	0.102	46.08	25.4	0.118
FRL-RWRM	80.69	**71.4**	0.088	49.16	32.0	0.221	47.45	28.1	0.108	45.94	26.1	0.147
CSL	76.29	67.1	-0.032	43.30	33.8	0.179	41.60	31.3	0.124	40.32	29.2	0.175
Ours	80.98	69.5	0.062	49.13	**36.6**	**0.403**	47.57	**33.3**	**0.326**	46.04	**30.1**	**0.334**

CIFAR-100	Natural			PGD-100			CW			AutoAttack		
Method	Avg.	Wst.	ρ_{nat}	Avg.	Wst.	ρ_{pgd}	Avg.	Wst.	ρ_{cw}	Avg.	Wst.	ρ_{AA}
TRADES	**54.57**	19.00	0	**27.39**	3.00	0	**24.87**	1.00	0	**23.57**	1.00	0
FRL-RW	53.08	**24.00**	**0.236**	25.76	3.00	-0.060	22.39	2.00	0.900	21.09	1.00	-0.105
FRL-RWRM	52.55	22.00	0.121	26.04	4.00	0.284	22.33	2.00	0.898	21.11	2.00	0.896
CSL	53.83	21.00	0.092	26.19	4.00	0.290	22.35	2.00	0.899	22.25	2.00	0.944
Ours	53.99	19.00	-0.020	26.91	**5.00**	**0.643**	24.26	**3.00**	**1.945**	22.89	**3.00**	**1.971**

Table 1: Comparison results of all methods using ResNet-18 on CIFAR-10 and CIFAR-100. We evaluate every method in terms of both accuracy (%) and ρ. We report the average natural accuracy, worst-class natural accuracy, average robust accuracy, worst-class robust accuracy, ρ_{nat}, ρ_{pgd}, ρ_{cw} and ρ_{AA} for every method. The best value in every metric is in bold font.



(a) Ours vs TRADES

(b) Ours vs CSL

(c) Ours vs FRL-RW

(d) Ours vs FRL-RWRM

Figure 4: Class-wise robust accuracy disparity of all methods using ResNet-18 on CIFAR-10. We compare our method and another method in terms of the class-wise robust accuracy evaluated under CW attack. We denote the results of our method with a blue line, while the results of the comparison methods are represented by a purple line.

We use CV_{nat} to denote the class-wise variance of natural accuracy and CV_{rob} to denote the class-wise variance of robustness accuracy, We also use ρ as defined in Eq.(7) to evaluate the method in terms of both the average and worst-class accuracies.

Results

In Table 1, we report the performance of every method using ResNet-18 on CIFAR-10 and CIFAR-100. We can clearly observe that our method successfully outperforms other methods on both CIFAR-10 and CIFAR-100. More specifically, under PGD-100 attack, our method improves the worst-class robust accuracy of all compared methods by at least 2.8% on the CIFAR-10 dataset and 1.0% on the CIFAR-100 dataset, while improving the worst-class robust accuracy of all compared methods for at least 2.0% on CIFAR-10 dataset and 1.0% on CIFAR-100 under CW attack. Under AutoAttack, our method improves the worst-class robust accuracy of all compared methods by at least 0.9% on the CIFAR-10 dataset and 1.0% on the CIFAR-100 dataset as well. Moreover, compared with TRADES, although all compared methods increase the robust accuracy, our method achieves the best ρ_{pgd}, ρ_{cw} and ρ_{AA} value; in short, we sacrifice the least average robust accuracy to obtain the highest worst-class robust accuracy.

Furthermore, to study the effectiveness of our method in more detail, we conduct a comparison of the class-wise robust accuracy evaluated under CW attack between our method and all compared methods in Figure 4. As shown in Figure 4(a), our method achieves higher robust accuracy of class-4 and class-5 than TRADES, thus, our method obtains a good performance on worst-class robust accuracy. In Figure 4(b), although CSL achieves a great performance on the worst class, it performs worse than our method on most other classes, which leads to a low average robust accuracy. From Figures 4(c) and 4(d), we can see that our method achieves higher robust accuracy on class-4 (the most vulnerable class) than the other two baselines. Moreover, our proposed method sig-

CIFAR-10	Natural			PGD-100			CW			AutoAttack		
Method	Avg.	Wst.	ρ_{nat}	Avg.	Wst.	ρ_{pgd}	Avg.	Wst.	ρ_{cw}	Avg.	Wst.	ρ_{AA}
TRADES	**84.51**	64.7	0	**53.68**	23.3	0	**53.18**	22.8	0	**51.22**	20.9	0
FRL-RW	83.93	74.5	**0.145**	50.59	30.0	0.230	50.58	29.1	0.227	48.36	27.1	0.241
FRL-RWRM	83.86	72.1	0.107	51.25	32.9	0.367	51.08	32.2	0.373	48.98	28.6	0.325
CSL	79.78	**75.1**	0.105	45.7	32.2	0.233	44.74	30.8	0.192	43.10	29.4	0.248
Ours	83.71	74.0	0.062	51.53	**34.9**	**0.458**	50.89	**33.4**	**0.422**	49.12	**30.7**	**0.428**

Table 2: Comparison results of all methods using WideResNet-34-10 on CIFAR-10.

CIFAR-10	Natural			PGD-100			CW			AutoAttack		
Method	Avg.	Wst.	ρ_{nat}	Avg.	Wst.	ρ_{pgd}	Avg.	Wst.	ρ_{cw}	Avg.	Wst.	ρ_{AA}
TRADES	**82.11**	64.6	0	**51.69**	25.2	0	**50.38**	24.1	0	**48.64**	21.7	0
Ours(η=0.01)	81.54	68.0	0.046	50.50	26.6	0.033	49.86	25.0	0.027	47.65	22.6	0.021
Ours(η=0.05)	81.76	69.3	**0.068**	50.06	34.2	0.326	49.53	31.7	0.298	47.05	28.1	0.262
Ours(η=0.1)	80.98	**69.5**	0.062	49.13	36.6	0.403	47.57	**33.3**	**0.326**	46.04	30.1	0.334
Ours(η=0.5)	79.30	67.3	0.008	48.09	**37.5**	**0.418**	45.42	32.5	0.250	43.98	**31.1**	**0.337**

Table 3: Results of our method with different η using ResNet-18 on CIFAR-10.

nificantly outperforms the other two baselines on class-5 and class-8, which contributes to the highest ρ_{cw} of our method. The results of class-wise robust accuracy disparity of all the methods evaluated under PGD-100 attack and AutoAttack on CIFAR-10 can be found in supplementary materials.

We go on to evaluate the performance of all the methods on WideResNet-34-10 (Zagoruyko and Komodakis 2016). The experimental results can be found in Table 2. From the results in Table 2, we can find that our method achieves the highest worst-class robust accuracy evaluated under all three attacks with at least 1.3% improvement. we also achieve the highest ρ_{pgd}, ρ_{cw} and ρ_{AA} while we have comparable result with compared methods in average robust accuracy evaluated under all three attacks on CIFAR-10.

Parameter Analysis on η

We study the impact of hyper-parameter η used in our method on average and worst-class robust accuracy. We vary the hyper-parameter η from $\{0.01, 0.05, 0.1, 0.5\}$, and show the results in Table 3. We find that a trade-off between the average robust accuracy and the worst-class robust accuracy exists, and if we improve the average robust accuracy, the worst-class robust accuracy decreases at the same time. However, a larger η does not lead to a larger ρ_{nat} and ρ_{cw} . In our experiments, we find $\eta = 0.1$ yields the best ρ_{nat} and ρ_{cw} while $\eta = 0.5$ yields the best ρ_{pgd} and ρ_{AA}.

Comparison between CV and ρ

From the results in Table 4, we can see that CSL obtains the lowest CV_{cw} value, while the average robust accuracy of CSL is the worst. Notably, CV_{cw} is not a good measurement because it does not consider the trade-off between average and worst-class robust accuracy. From the results in Table 4, we can also see that our method achieves the best ρ_{cw}, has the highest worst-class robust accuracy, and is comparable with FRL and CSL in average robust accuracy. Therefore, ρ_{cw} is a more reasonable measurement than CV_{cw} because it considers average robust accuracy and worst-class robust accuracy at the same time. The results evaluated under PGD-100 attack and AutoAttack are shown in supplementary materials.

CIFAR-10	CW Attack			
Method	Avg.	Wst.	CV_{cw}	ρ_{cw}
TRADES	**50.38**	24.1	0.0269	0
FRL-RW	47.80	27.8	0.0215	0.102
FRL-RWRM	47.45	28.1	0.0172	0.108
CSL	41.60	31.3	**0.0027**	0.124
Ours	47.57	**33.3**	0.0147	**0.326**

Table 4: Comparison results between CV_{cw} and ρ_{cw} using ResNet-18 on CIFAR-10.

Conclusion

To improve the worst-class robustness in adversarial training, this paper proposes a novel framework of worst-class adversarial training and leverages no-regret dynamics to solve the problem. Theoretically, we provide the guarantee of the worst-class loss and analyze the generalization error bound in terms of the worst-class robust risk based on Rademacher complexity. Moreover, we propose a measurement to evaluate the method in terms of both the average and worst-class accuracies. Empirical results verify the superiority of our proposed approach.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant 61976161.

References

Arora, S.; Hazan, E.; and Kale, S. 2012. The Multiplicative Weights Update Method: a Meta-Algorithm and Applications. *Theory of Computing*, 8(1): 121–164.

Bartlett, P. L.; and Mendelson, S. 2002. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. *Journal of Machine Learning Research*, 3: 463–482.

Benz, P.; Zhang, C.; Karjauv, A.; and Kweon, I. S. 2020. Robustness May Be at Odds with Fairness: An Empirical Study on Class-wise Accuracy. *CoRR*, abs/2010.13365.

Carlini, N.; and Wagner, D. A. 2017. Towards Evaluating the Robustness of Neural Networks. In *S&P*.

Carmon, Y.; Raghunathan, A.; Schmidt, L.; Duchi, J. C.; and Liang, P. 2019. Unlabeled Data Improves Adversarial Robustness. In *NeurIPS*.

Croce, F.; and Hein, M. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, volume 119, 2206–2216.

Ebrahimi, J.; Rao, A.; Lowd, D.; and Dou, D. 2018. HotFlip: White-Box Adversarial Examples for Text Classification. In Gurevych, I.; and Miyao, Y., eds., *ACL*, 31–36.

Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; and Song, D. 2018. Robust Physical-World Attacks on Deep Learning Visual Classification. In *CVPR*, 1625–1634.

Freund, Y.; and Schapire, R. E. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1): 119–139.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *ICLR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

Khan, S. H.; Hayat, M.; Bennamoun, M.; Sohel, F. A.; and Togneri, R. 2018. Cost-Sensitive Learning of Deep Feature Representations From Imbalanced Data. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8): 3573–3587.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto.

Li, X.; Zou, X.; and Liu, W. 2022. Defending Against Adversarial Attacks via Neural Dynamic System. In *NeurIPS*.

Ma, X.; Wang, Z.; and Liu, W. 2022. On the Tradeoff Between Robustness and Fairness. In *NeurIPS*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*.

Montasser, O.; Hanneke, S.; and Srebro, N. 2019. VC Classes are Adversarially Robustly Learnable, but Only Improperly. In *COLT*, volume 99, 2512–2530.

Pang, T.; Yang, X.; Dong, Y.; Su, H.; and Zhu, J. 2021. Bag of Tricks for Adversarial Training. In *ICLR*.

Raghunathan, A.; Steinhardt, J.; and Liang, P. 2018. Certified Defenses against Adversarial Examples. In *ICLR*.

Roughgarden, T.; and Iwama, K. 2017. Twenty Lectures on Algorithmic Game Theory. *Bulletin of the EATCS*, 122.

Simon-Gabriel, C.; Ollivier, Y.; Bottou, L.; Schölkopf, B.; and Lopez-Paz, D. 2019. First-Order Adversarial Vulnerability of Neural Networks and Input Dimension. In *ICML*, volume 97, 5809–5817.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In *ICLR*.

Tian, Q.; Kuang, K.; Jiang, K.; Wu, F.; and Wang, Y. 2021. Analysis and Applications of Class-wise Robustness in Adversarial Training. In *KDD*, 1561–1570.

Ting, K. M. 2000. A Comparative Study of Cost-Sensitive Boosting Algorithms. In *ICML*, 983–990.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2019. Robustness May Be at Odds with Accuracy. In *ICLR*.

Wang, Z.; and Liu, W. 2022. Robustness Verification for Contrastive Learning. In *ICML*, volume 162, 22865–22883.

Wu, B.; Chen, J.; Cai, D.; He, X.; and Gu, Q. 2021. Do Wider Neural Networks Really Help Adversarial Robustness? In *NeurIPS*.

Xu, H.; Liu, X.; Li, Y.; Jain, A. K.; and Tang, J. 2021. To be Robust or to be Fair: Towards Fairness in Adversarial Training. In *ICML*, volume 139, 11492–11501.

Xu, J.; and Liu, W. 2022. On Robust Multiclass Learnability. In *NeurIPS*.

Xu, K.; Zhang, G.; Liu, S.; Fan, Q.; Sun, M.; Chen, H.; Chen, P.; Wang, Y.; and Lin, X. 2020. Adversarial T-Shirt! Evading Person Detectors in a Physical World. In *ECCV*, volume 12350, 665–681.

Yan, H.; Zhang, J.; Niu, G.; Feng, J.; Tan, V. Y. F.; and Sugiyama, M. 2021. CIFS: Improving Adversarial Robustness of CNNs via Channel-wise Importance-based Feature Selection. In *ICML*.

Yang, Y.; Rashtchian, C.; Zhang, H.; Salakhutdinov, R. R.; and Chaudhuri, K. 2020. A Closer Look at Accuracy vs. Robustness. In *NeurIPS*.

Yin, D.; Ramchandran, K.; and Bartlett, P. L. 2019. Rademacher Complexity for Adversarially Robust Generalization. In *ICML*, volume 97, 7085–7094.

Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. In *BMVC*.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E. P.; Ghaoui, L. E.; and Jordan, M. I. 2019. Theoretically Principled Trade-off between Robustness and Accuracy. In *ICML*, volume 97, 7472–7482.

Zhou, Z.; and Liu, X. 2006. Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1): 63–77.

Zou, Y.; Yu, Z.; Kumar, B. V. K. V.; and Wang, J. 2018. Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-training. In *ECCV*, volume 11207, 297–313.