To be Robust or to be Fair: Towards Fairness in Adversarial Training

Han Xu [*1] Xiaorui Liu [*1] Yaxin Li [1] Anil K. Jain [1] Jiliang Tang [1]

Abstract

Adversarial training algorithms have been proved to be reliable to improve machine learning models' robustness against adversarial examples. However, we find that adversarial training algorithms tend to introduce severe disparity of accuracy and robustness between different groups of data. For instance, a PGD adversarially trained ResNet18 model on CIFAR-10 has 93% clean accuracy and 67% PGD l_∞-8 robust accuracy on the class "automobile" but only 65% and 17% on the class "cat". This phenomenon happens in balanced datasets and does not exist in naturally trained models when only using clean samples. In this work, we empirically and theoretically show that this phenomenon can happen under general adversarial training algorithms which minimize DNN models' robust errors. Motivated by these findings, we propose a Fair-Robust-Learning (FRL) framework to mitigate this unfairness problem when doing adversarial defenses. Experimental results validate the effectiveness of FRL.

1. Introduction

The existence of adversarial examples (Goodfellow et al., 2014; Szegedy et al., 2013) causes great concerns when applying deep neural networks to safety-critical tasks, such as autonomous driving vehicles and face identification (Morgulis et al., 2019; Sharif et al., 2016). As countermeasures against adversarial examples, adversarial training algorithms aim to train a classifier that can classify the input samples correctly even when they are adversarially perturbed. Namely, they optimize the model to have the minimum adversarial risk of a sample that can be perturbed to be wrongly classified:

$$\min_f \mathbb{E}_x \left[\max_{||\delta|| \leq \epsilon} \mathcal{L}(f(x+\delta), y) \right].‘ \tag{1}$$

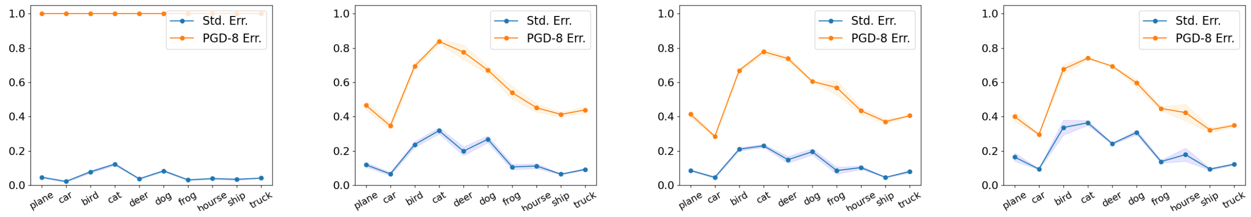
These adversarial training methods (Kurakin et al., 2016; Madry et al., 2017; Zhang et al., 2019b) have been shown to be one of the most effective and reliable approaches to improve the model robustness against adversarial attacks.

Although promising to improve the model's robustness, we reveal an intriguing property about adversarial training algorithms: they usually result in a large disparity of accuracy and robustness among different classes. As a preliminary study in Section 2, we apply natural training and PGD adversarial training (Madry et al., 2017) on the CIFAR10 dataset (Krizhevsky et al., 2009) using a PreAct-ResNet18 (He et al., 2016) architecture. For a naturally trained model, the model performance in each class is similar. However, in the adversarially trained model, there is a severe class-wise performance discrepancy (both accuracy and robustness). For example, the model has both low standard and robust errors on the samples from the class "car", but it has a much larger error rate on those "cat" images. Meanwhile, this fairness issue does not appear in natural models which are trained on clean data. Note that this phenomenon happens in a balanced CIFAR10 dataset and exists in other datasets, model structures and adversarial training algorithms. More details can be found in Section 2. If this phenomenon happens in real-world applications, it can raise huge concerns about safety. Imagine that a traffic sign recognizer has an overall high performance, but it is very inaccurate and vulnerable to perturbations only for some specific signs. The safety of this autonomous driving car is still not guaranteed. Meanwhile, this phenomenon can also lead to issues from the social ethics perspective. For example, a robustly trained face identification system might provide different levels of safety for the services provided to different ethnic communities. Thus, there is a pressing need to study this phenomenon.

In this work, we define this phenomenon as the "*robust fairness*" problem of adversarial training and we aim to understand and mitigate this fairness problem. We first explore the question: "*why does adversarial training have this accuracy/robustness disparity between classes while natural training does not present a similar issue?*" To answer this

[*] Equal contribution [1] Department of Computer Science and Engineering, Michigan State University, Michigan, U.S.. Correspondence to: Han Xu <xuhan1@msu.edu>, Xiaorui Liu <xiaorui@msu.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, PMLR 108, 2021. Copyright 2021 by the author(s).



(a) Natural Training (b) PGD Adversarial Training (c) TRADES $(1/\lambda = 1)$ (d) TRADES $(1/\lambda = 6)$

Figure 1. The class-wise standard / robust error of natural and adversarial training methods on CIFAR10, under PreAct ResNet18.

question, we first study on several cases on rather simple classification problems with mixture Gaussian distributions, where we design the data in different classes with different "difficulty levels" to be classified. By studying the class-wise performance of naturally and adversarially trained models, we deepen our understandings on this problem. Compared to natural training, adversarial training has a stronger tendency to favor the accuracy of the classes which are "easier" to be predicted. Meanwhile, adversarial training will sacrifice the prediction performance for the "difficult" classes. As a result, there is a much obvious class-wise performance discrepancy in adversarial training. Motivated by our empirical findings and theoretical understandings, we propose a dynamic debiasing framework, i.e., Fair Robust Learning (FRL), to mitigate the robust fairness issue in adversarial settings. Our main contributions are summarized as:

- We discover the robust fairness problem of adversarial training algorithms and empirically verify that this problem can be general;
- We build conceptual examples to understand the potential reasons that cause this fairness problem; and
- We propose a Fair Robust Learning (FRL) framework to mitigate the fairness issue in adversarial training.

2. Preliminary Studies

In this section, we introduce our preliminary findings to show that adversarial training algorithms usually present the fairness issue, which is related to the strong disparity of standard accuracy and robustness among different classes. We first examine algorithms including PGD adversarial training (Madry et al., 2017) and TRADES (Zhang et al., 2019b) on the CIFAR10 dataset (Krizhevsky et al., 2009) under the l_∞-norm adversarial attack (under perturbation bound 8/255). In CIFAR10, we both naturally and adversarially train PreAct-ResNet18 (He & Garcia, 2009) models. The results are presented in Figure 1.

From Figure 1, we can observe that – for the naturally trained models, every class has similar standard error (around $7 \pm 5\%$) and 100% robust error under the 8/255 PGD attack. However, for adversarially trained models, the disparity phenomenon is severe. For example, a PGD-adversarially trained model has 32.8% standard error and 82.4% robust error for the samples in the class "cat", which are much larger than the model's average standard error 15.5% and average robust error 56.4%. While, the best class "car" only has 6.1% standard error and 34.3% robust error. These results suggest that adversarial training can cause strong disparities of standard / robustness performance between different classes, which are originally negligible in natural training.

Moreover, from Figure 1, we find that the reason of this fairness phenomenon might be due to the unequal influence of adversarial training on different classes. It tends to hurt the standard performance of classes which are intrinsically "harder" to be classified, but not effectively improve their robustness performance. In Table 1, we list the classes "dog" and "cat", which have the highest errors in natural training, as well as "car" and "ship", which have the lowest errors. We can observe that adversarial training increases the standard errors of "dog" and "cat" by a much larger margin than the classes "car" and "ship". Similarly, adversarial training gives poorer help to reduce the robust errors of "dog" and "cat". As a conclusion, we hypothesize adversarial training tends to make the hard classes even harder to be classified or robustly classified. In Section 3, we will theoretically confirm this hypothesis.

We further investigate other settings including model architecture WRN28, SVHN dataset, l_2-norm adversarial training and Randomized Smoothing (Cohen et al., 2019) algorithm. Results can be found at Appendix A.1 where we can make similar observations. These findings suggest that observations from Figure 1 and Table 1 are likely to be generalized into other adversarial training algorithms, model architectures, datasets and adversarial attacks.

Table 1. The Changes of Standard & Robust Error in Natural & Adversarial Training in CIFAR10.

Std. Error	**Cat**	**Dog**	**Car**	**Ship**
Nat. Train	11.3	10.0	1.8	3.5
PGD Adv. Train	34.8	26.9	6.1	6.4
Diff. (Adv. - Nat.)	23.5	16.9	4.3	2.9

Rob. Error	**Cat**	**Dog**	**Car**	**Ship**
Nat. Train	100	100	100	100
PGD Adv. Train	82.7	66.4	34.3	40.8
Diff. (Adv. - Nat.)	-17.3	-33.5	-65.7	-59.2

3. Theoretical Analysis

From our preliminary studies, we consistently observe that adversarially trained models have great performance disparity (both standard and robust errors) between different classes. Moreover, adversarial training tends to hurt the classes which are originally harder to be classified in natural training. What is the reason that leads to this phenomenon? Is this "unfairness" property an inherent property of adversarial training methods? These questions are not trivial to answer because it is closely related to another property of adversarial training: it usually degrades the model's average accuracy. Given that naturally trained models already present slight disparities between classes (see Figure 1 (a)), is the larger class-wise accuracy disparity in adversarial training only a natural outcome of the model's worse average accuracy?

To deepen our understandings on these questions, we theoretically study the effect of adversarial training on a binary classification task under a mixture Gaussian distribution. We design the two classes with different "difficulties" to be classified. In such case, adversarial training will not significantly degrade the average standard error, but its decision boundary, compared to naturally trained models, are more biased to favor the "easier" class and hurt the "harder" class. From this theoretical study, we aim to validate that adversarial training methods can have the intrinsic property to give unequal influence between different classes and consequently cause the fairness problem. In the following, we first introduce the necessary notations and definitions.

Notation. We use f to denote the classification model which is a mapping $f : \mathcal{X} \to \mathcal{Y}$ from input data space $\mathcal{X}$ and output labels $\mathcal{Y}$. Generally, for a classifier f, the overall *standard error* is defined as $\mathcal{R}_{\text{nat}}(f) = \text{Pr.}(f(x) \neq y)$; and its overall *robust error* is $\mathcal{R}_{\text{rob}}(f) = \text{Pr.}(\exists \delta, ||\delta|| \leq \epsilon, \text{s.t.} f(x+\delta) \neq y)$, which is the probability that there exists a perturbation to make the model give a wrong prediction.[1] We use $\mathcal{R}_{\text{nat}}(f;y)$ to denote the standard error conditional on a specific class $Y = y$.

3.1. A Binary Classification Task

We start by giving a rather simple example of a binary classification task with a Gaussian mixture data. Here, we aim to design the two classes with inherent different "difficulties" to be classified. Specifically, in the following definition, the data are from 2 classes $\mathcal{Y} = \{-1, +1\}$ and the data from each class follow a Gaussian distribution $\mathcal{D}$ which is centered on $-\theta$ and θ respectively. In our case, we specify that there is a K-factor difference between two classes' variance: $\sigma_{+1} : \sigma_{-1} = K : 1$ and $K > 1$.

$$y \overset{u.a.r}{\sim} \{-1, +1\}, \quad \theta = (\overbrace{\eta, ..., \eta}^{\text{dim = d}}),$$
$$x \sim \begin{cases} \mathcal{N}\left(\theta, \sigma_{+1}^2 I\right) & \text{if } y = +1 \\ \mathcal{N}\left(-\theta, \sigma_{-1}^2 I\right) & \text{if } y = -1 \end{cases} \tag{2}$$

Intuitively, the class "+1" is harder than class "-1" because it is less compacted in the data space. In Theorem 1, we formally show that the class "+1" can be indeed harder because an optimal linear classifier will give a larger error for the class "+1" than class "-1".

Theorem 1 *For a data distribution $\mathcal{D}$ in Eq. 2, the optimal linear classifier f_{nat} which minimizes the average standard classification error:*

$$f_{nat} = \arg\min_f Pr.(f(x) \neq y)$$

It has the intra-class standard error for the two classes:

$$\begin{aligned} \mathcal{R}_{nat}(f_{nat}, -1) = Pr.\{\mathcal{N}(0,1) \leq A - K \cdot \sqrt{A^2 + q(K)}\} \\ \mathcal{R}_{nat}(f_{nat}, +1) = Pr.\{\mathcal{N}(0,1) \leq -K \cdot A + \sqrt{A^2 + q(K)}\} \end{aligned} \tag{3}$$

where $A = \frac{2}{K^2-1}\frac{\sqrt{d}\eta}{\sigma}$ and $q(K) = \frac{2\log K}{K^2-1}$ which is a positive constant and only depends on K, As a result, the class "+1" has a larger standard error:

$$\mathcal{R}_{nat}(f_{nat}, -1) < \mathcal{R}_{nat}(f_{nat}, +1).$$

A detailed proof of Theorem 1 can be found in Appendix A.2. From Theorem 1, it demonstrates that class "+1" (with large variance) can be harder to be classified than class "-1", because an optimal natural classifier will present a larger standard error in class "+1". Note that the classwise difference is due to the positive term $q(K)$ in Eq. 16, which depends on the variance ratio K. If the two classes' variances are equal, i.e., $K = 1$, the standard errors for the two classes are the same. Next, we will show that, in the setting of Eq. 2, an optimal robust classifier (adversarial training) will give a model which further benefits the easier class "-1" and hurt the harder class "+1".

[1] In this section, we focus on l_∞-norm bounded perturbation.

3.2. Optimal Linear Model to Minimize Robust Error

In this subsection, we demonstrate that an adversararially trained model exacerbates the performance gap between these two classes, by giving a decision boundary which is closer to samples in the harder class "+1" and farther to the class "-1". Similar to Theorem 1, we calculate the classwise standard errors for robust classifiers.

Theorem 2 *For a data distribution $\mathcal{D}$ in Eq. 2, the optimal robust linear classifier f_{rob} which minimizes the average robust error:*

$$f_{rob} = \arg\min_{f} Pr.(\exists ||\delta|| \leq \epsilon \ \ s.t. \ \ f(x+\delta) \neq y)$$

It has the intra-class standard error for the two classes:

$$\begin{aligned} &\mathcal{R}_{nat}(f_{rob}, -1) \\ =&Pr.\{\mathcal{N}(0,1) \leq B - K \cdot \sqrt{B^2 + q(K)} - \frac{\sqrt{d}}{\sigma}\epsilon\} \\ &\mathcal{R}_{nat}(f_{rob}, +1) \\ =&Pr.\{\mathcal{N}(0,1) \leq -K \cdot B + \sqrt{B^2 + q(K)} - \frac{\sqrt{d}}{K\sigma}\epsilon\} \end{aligned} \tag{4}$$

where $B = \frac{2}{K^2-1}\frac{\sqrt{d}(\eta-\epsilon)}{\sigma}$ and $q(K) = \frac{2\log K}{K^2-1}$ is a positive constant and only depends on K,

Note that we limit the perturbation margin ϵ in the region $0 < \epsilon < \eta$ to guarantee that the robust optimization gives a reasonable classification in this setting in Eq. 2. The detailed proof is similar to Theorem 1 and is given in Appendix A.2. From the results in Theorems 1 and 2, an corollary can demonstrate that the robust classifier will further hurt the "harder" class's performance.

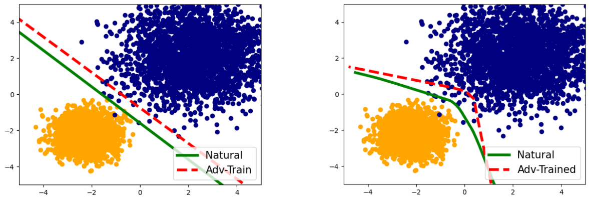
Corollary 1 *Adversarially Trained Models on $\mathcal{D}$ will increase the standard error for class "+1" and reduce the standard error for class "-1":*

$$\mathcal{R}_{nat}(f_{rob}, -1) < \mathcal{R}_{nat}(f_{nat}, -1).$$
$$\mathcal{R}_{nat}(f_{rob}, +1) > \mathcal{R}_{nat}(f_{nat}, +1).$$

Proof 1 *A simplified proof sketch for Corollary 1 can help shed light on the reason of this behavior for adversarial training. First, from the definition of the distribution $\mathcal{D}$, it is easy to have the intermediate result that natural / robust classifiers have the weight vectors $w_{nat} = w_{rob} = \mathbf{1}$. The only difference is their interception terms b_{nat} and b_{rob}. In Appendix A.2, we show that:*

$$b_{nat} = \frac{K^2+1}{K^2-1}d\eta - K\sqrt{\frac{4}{(K^2-1)^2} \cdot d^2\eta^2 + d\sigma^2 q(K)} := g(\eta). \tag{5}$$

In particular, for robust classifiers, if we look at the objective of robust classification:



(a) Logistic Regression (b) One Layer Perceptron

Figure 2. Logistic Regression and Multi-perceptron classifiers (natural and robust) on simulated binary data in Eq. 2.

$$\begin{aligned} \mathcal{R}_{rob}(f) =&Pr.(\exists ||\delta|| \leq \epsilon \ \ s.t. \ \ f(x+\delta) \neq y) \\ =& \max_{||\delta|| \leq \epsilon} Pr.(f(x+\delta) \neq y) \\ =&\frac{1}{2}Pr.(f(x+\epsilon) \neq -1 | y = -1) \\ +&\frac{1}{2}Pr.(f(x-\epsilon) \neq +1 | y = +1) \end{aligned}$$

the robust classifier f_{rob} directly minimizes the standard error of samples $(x-\epsilon)$ for x in class "+1", and it minimizes the error of samples $(x+\epsilon)$ for x in class "-1". As a consequence, the robust model f_{rob} minimizes the errors of samples whose centers are $\pm\theta' = \pm(\eta-\epsilon, ..., \eta-\epsilon)$, which are both ϵ-distance closer to zero $\mathbf{0}$ compared to $\pm\theta$. Thus, we can get the interception term of the robust model, by only replacing η in b_{nat} by $(\eta - \epsilon)$:

$$b_{rob} = g(\eta - \epsilon).$$

In Appendix A.2, we show g is a monotone increasing function from 0 to η; thus we have the relation $0 < b_{rob} < b_{nat}$. This suggests that a robust classifier will predict more samples in $\mathcal{D}$ to be a negative "-1" class, and hence reduce the classification error for class "-1" but increase the error of class "+1".

In Figure 2,we give an illustration for the theories in a 2-dim space to show the effect of adversarial training. Particularly, we sampled Gaussian data from class "+1" (blue) which centered on $\theta = (2,2)$, and has variance $\sigma^2_{+1} = 2$, as well as class "-1" (yellow) which centered on $-\theta = (-2,-2)$, and has variance $\sigma^2_{-1} = 1$. We apply a logistic regression classifier for natural training (green line) and adversarial training (with perturbation bound $\epsilon = 0.5$, red line). From Figure 2, we get the consistent results with our theories: (1) there are more samples in the class "+1" than class "-1" which are wrongly predicted by a natural classifier; (2) adversarial training will further exacerbate the disparity by moving decision boundary closer to the harder class "+1". In Figure 2 (right), we also show the results for a non-linear MLP classifier, where we can make similar observations.

These theoretical understandings about adversarial training suggest that adversarial training will naturally bring unequal effect to different classes in the data distribution and consequently cause severe fairness issues.

4. Fair Robust Learning (FRL)

The observations from both preliminary experiments and theoretical understandings suggest that the fairness issues in adversarial training can be a general and serious concern. In this section, we first discuss fairness requirements that an optimal robust model should satisfy, and then introduce algorithms to achieve these objectives.

4.1. Objective of Robust Fairness

In this work, we desire an optimal robust model that can achieve the parity of both standard prediction accuracy and adversarial robustness among each class $y \in Y$:

- ***Equalized Accuracy***: one classifier f's standard error is statistically independent of the ground truth label y: Pr.$(f(x) \neq y | Y = y) \approx$ Pr.$(f(x) \neq y)$ for all $y \in Y$.
- ***Equalized Robustness***: one classifier f's robust error is statistically independent of the ground truth label y: Pr.$(\exists \, ||\delta|| \leq \epsilon, f(x+\delta) \neq y | Y = y) \approx$ Pr.$(\exists \, ||\delta|| \leq \epsilon, f(x+\delta) \neq y)$ for all $y \in Y$.

The notion of "Equalized Accuracy" is well studied in traditional fairness research (Buolamwini & Gebru, 2018; Zafar et al., 2017) which desires the model to provide equal prediction quality for different groups of people. The "Equalized Robustness" is our new desired "fairness" property for robustly trained models. For every class, the model should provide equal robustness and safety to resist adversarial attacks. Therefore, the robust model will have high overall safety but no obvious "weakness".

Faced with the fairness objectives mentioned above, in our work, we propose a Fair Robust Learning (FRL) strategy to train robust models that have equalized accuracy and robustness performance for each class. Formally, we aim to train a classifier f to have minimal overall robust error ($\mathcal{R}_{\text{rob}}(f)$), as well stressing f to satisfy a series of fairness constraints as:

$$\begin{aligned} &\underset{f}{\text{minimize}} \quad \mathcal{R}_{\text{rob}}(f) \\ &\text{s.t.} \quad \begin{cases} \mathcal{R}_{\text{nat}}(f,i) - \mathcal{R}_{\text{nat}}(f) \leq \tau_1 \\ \mathcal{R}_{\text{rob}}(f,i) - \mathcal{R}_{\text{rob}}(f) \leq \tau_2 \end{cases} \quad \text{for each } i \in Y \end{aligned} \tag{6}$$

where τ_1 and τ_2 are small and positive predefined paramters. The constraints in Eq. 6 restrict the model's error for each class $i \in Y$ (both standard error $\mathcal{R}_{\text{nat}}(f,i)$ and robust error $\mathcal{R}_{\text{rob}}(f,i)$) should not exceed the average level ($\mathcal{R}_{\text{nat}}(f)$ and $\mathcal{R}_{\text{rob}}(f)$) by a large margin. Thus, there will be no obvious worst group in the whole dataset. One thing to note is that in Eq 6, the robust error is always strongly related to the standard error (Zhang et al., 2019b; Tsipras et al., 2018) (see Eq. 7). Thus, during the debiasing process, we could have a twisted influence on the class i's standard and robust errors. For example, if we apply some importance weighting methods to upweight the cost of $\mathcal{R}_{\text{rob}}(f,i)$, we also implicitly upweight the cost for $\mathcal{R}_{\text{nat}}(f,i)$. Therefore, we propose to separate the robust error into the sum of *standard error* and *boundary error* inspired by (Zhang et al., 2019b) as:

$$\begin{aligned} &\mathcal{R}_{\text{rob}}(f,i) \\ =&\text{Pr.}\{\exists \delta, \text{ s.t. } f(x+\delta) \neq y | y = i\} \\ =&\text{Pr.}\{f(x) \neq y | y = i\} + \text{Pr.}\{\exists \delta, f(x+\delta) \cdot f(x) \leq 0 | y = i\} \\ =&\mathcal{R}_{\text{nat}}(f,i) + \mathcal{R}_{\text{bndy}}(f,i) \end{aligned} \tag{7}$$

where $\mathcal{R}_{\text{bndy}}(f,i) = \text{Pr.}\{\exists \delta, f(x+\delta) \cdot f(x) \leq 0 | y = i\}$ represents the probability that a sample from class i lies close to the decision boundary and can be attacked. By separating the standard error and boundary error during adversarial training, we are able to independently solve the unfairness of both standard error and boundary error. Formally, we have the training objective as:

$$\begin{aligned} &\underset{f}{\text{minimize}} \quad \mathcal{R}_{\text{nat}}(f) + \mathcal{R}_{\text{bndy}}(f) \\ &\text{s.t.} \quad \begin{cases} \mathcal{R}_{\text{nat}}(f,i) - \mathcal{R}_{\text{nat}}(f) \leq \tau_1 \\ \mathcal{R}_{\text{bndy}}(f,i) - \mathcal{R}_{\text{bndy}}(f) \leq \tau_2 \text{ for each } i \in Y \end{cases} \end{aligned} \tag{8}$$

During training to optimize the boundary errors, we borrow the idea from (Zhang et al., 2019b), which minimizes the KL-divergence between the output logits of clean samples and their adversarial samples. In the following subsections, we explore effective methods to solve the problem in Eq. 8.

4.2. Reweight for Robust Fairness

In order to solve the fair robust training problem in Eq. 8, we first follow the main pipeline from traditional machine learning debiasing works such as (Agarwal et al., 2018; Zafar et al., 2017), which reduce the problem in Eq. 6 into a series of *Cost-sensitive* classification problems and continuously penalize the terms which violate the fairness constraints. We begin by introducing Lagrange multipliers $\phi = (\phi^i_{\text{nat}}, \phi^i_{\text{bndy}})$ (non-negative) for each constraint in Eq. 6 and form the Lagrangian:

$$\begin{aligned} L(f,\phi) =&\mathcal{R}_{\text{nat}}(f) + \mathcal{R}_{\text{bndy}}(f) \\ &+ \sum_{i=1}^{Y} \phi^i_{\text{nat}} (\mathcal{R}_{\text{nat}}(f,i) - \mathcal{R}_{\text{nat}}(f) - \tau_1)^+ \\ &+ \sum_{i=1}^{Y} \phi^i_{\text{bndy}} (\mathcal{R}_{\text{bndy}}(f,i) - \mathcal{R}_{\text{bndy}}(f) - \tau_2)^+ \end{aligned} \tag{9}$$

Thus, the problem in Eq. 8 equals to solving the max-min game between two rivals f and ϕ as:

$$\max_{\phi_{\text{nat}}, \phi_{\text{rob}} \geq 0} \min_{f} \ L(f, \phi). \tag{10}$$

We present the details for solving Eq. 10 in Algorithm 1. During the training process, we start from a pre-trained robust model and we test its class-wise standard / boundary errors on a separated validation set (step 5 and 6). We check whether the current model f violates some constraints in Eq. 8. For example, if there exists a class "i", whose standard error is higher than the average by a larger margin: $\mathcal{R}_{\text{nat}}(f, i) - \mathcal{R}_{\text{nat}}(f) - \tau_1 > 0$. We will adjust its multiplier term ϕ^i_{nat} according to the extent of violation (step 7). As a result, we increase the training weight for the standard error $\mathcal{R}_{\text{nat}}(f, i)$ for the class i. We also adjust the multiplier for boundary errors at the same time (step 8). Next, fixing the multipliers ϕ, the algorithm will solve the inner-minimization to optimize the model f. By repeating the steps 4-9, the model f and Lagrangian multiplier ϕ will be alternatively updated to achieve the equilibrium until we finally reach an optimal model that satisfies the fairness constraints. Note that we denote the framework with the proposed Reweight strategy as FRL (Reweight).

Algorithm 1 The Fair Robust Learning (FRL) Algorithm

```
1: Input: Fairness constraints specified by $\tau_1 > 0$ and $\tau_2 > 0$, test time attacking radius $\epsilon$ and hyper-param update rate $\alpha_1, \alpha_2$
2: Output: A fairly robust neural network $f$
3: Initialize network with a pre-trained robust model
   Set $\phi^i_{\text{nat}} = 0$, $\phi^i_{\text{bndy}} = 0$ and $\phi = (\phi_{\text{nat}}, \phi_{\text{bndy}})$,
4: repeat
5:     $\mathcal{R}_{\text{nat}}(f), \mathcal{R}_{\text{nat}}(f, i) = \text{EVAL}(f)$
6:     $\mathcal{R}_{\text{bndy}}(f), \mathcal{R}_{\text{bndy}}(f, i) = \text{EVAL}(f, \epsilon)$
7:     $\phi^i_{\text{nat}} = \phi^i_{\text{nat}} + \alpha_1 \cdot (\mathcal{R}_{\text{nat}}(f, i) - \mathcal{R}_{\text{nat}}(f) - \tau_1)$
8:     $\phi^i_{\text{bndy}} = \phi^i_{\text{bndy}} + \alpha_2 \cdot (\mathcal{R}_{\text{bndy}}(f, i) - \mathcal{R}_{\text{bndy}}(f) - \tau_2)$
9:     $f \leftarrow \text{TRAIN}(f, \phi, \epsilon)$
10: until Model $f$ satisfies all constraints
```

4.3. ReMargin for Robust Fairness

Though upweighting the cost for $\mathcal{R}_{\text{nat}}(f, i)$ has the potential to help penalize large $\mathcal{R}_{\text{nat}}(f, i)$ and improve the standard performance for worse groups. However, only upweighting one class's boundary error's cost could not succeed to fulfill the goal to help decrease its boundary error. In Section 5.3, we show this fact in PGD adversarial training on CIFAR10, where we find that even we give a large weight ratio for a class's boundary error, it is not sufficient to reduce the boundary error for this class. Thus, we cannot achieve to mitigate the boundary error disparity and robust error disparity.

To solve this problem, we propose an alternative strategy by enlarging the perturbation margin ϵ. It is evident from some existing works (Tramèr et al., 2020; Ding et al., 2018) that increasing the margin ϵ during adversarial training can effectively improve model's robustness against attacks under the current intensity ϵ. Therefore, enlarging the adversarial margin ϵ when generating adversarial examples during training specifically for the class i has the potential to improve this class's robustness and reduce the large boundary error $\mathcal{R}_{\text{bndy}}(f, i)$. In this work, we define this strategy as FRL (Remargin). The FRL (Remargin) resembles the procedure in Algorithm 1, except for the step 7, where we instead update the adversarial margin for the boundary errors. Specifically, we change the adversarial margin ϵ^i of the class "i" as follows:

$$\epsilon^i = \epsilon^i \cdot \exp\left(\alpha_2^* (\mathcal{R}_{\text{bndy}}(f, i) - \tau_2)\right)) \tag{11}$$

Besides FRL (Reweight) and FRL (Remargin), we can combine Reweight and Remargin, where we jointly update the weight of the boundary errors and change the margin.

5. Experiment

In this section, we present the experimental results to validate the effectiveness of the proposed framework (FRL) on building fairly robust DNN models. We implement and compare our proposed three strategies (i.e., Reweight, Remargin and Reweight+Remargin) on real-world data and discuss their effectiveness. The implementation of the proposed algorithms can be found via `https://drive.google.com/open?id=1IjE11VpV5CGXx633Nef3McgVOoRGMK9x`.

5.1. Experimental Settings

We conduct our experiments on benchamrk adversarial learning datasets, including CIFAR10 (Krizhevsky et al., 2009) and SVHN dataset (Netzer et al., 2011). For both datasets, we study the algorithms under the model architectures PreAct-ResNet18 and WRN28. We only present the results of PreAct-ResNet18 in this section and we leave the results of WRN28 in Appendix A.2.2. As baselines, we present the original performance of two popular adversarial training algorithms (Madry et al., 2017; Zhang et al., 2019b), and a debiasing method which is inherited from (Agarwal et al., 2018). It is a traditional debiasing technique and we directly apply it to upweight the cost of the class with the highest robust error in the training data. Other existing unfairness debiasing methods, such as (Zafar et al., 2017; Zhang et al., 2018) are not included in our experiment, because they are not proposed for deep learning models and have similar ideas with (Agarwal et al., 2018) to reweight the costs during training.

In our implementation for FRL methods, during the training process, we split the training sets to get validation sets with 300 samples in each class to help us adjust the hyperparameters. For each variant of our proposed FRL method, we pre-define the model to achieve fairness constraints to satisfy that both τ_1 and τ_2 are not larger than 5% or 7%. In the training, we start FRL from a pre-trained robust model (such as a PGD-adversarial training), and run FRL with model parameter learning rate 1e-3 and hyperparameter learning rate $\alpha_1 = \alpha_2 = 0.05$ in the first 40 epochs. Then we decay the model parameter learning rate and the hyperparameter learning rate by 0.1 every 40 epochs. During the evaluation phase, we report each trained model's average *standard error* rate, *boundary error* rate and *robust error* rate, as well as the worst intraclass error rate. Note that the boundary and robust errors are calculated by the PGD attack under l_∞-norm 8/255.

5.2. Fairness Performance

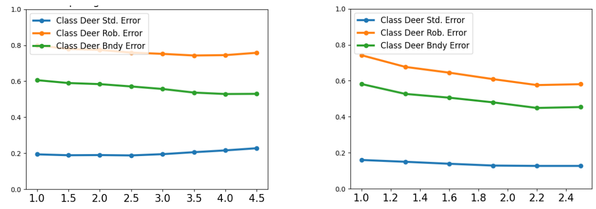
Table 2 shows each algorithm's performance on the CIFAR10 dataset, including each variant of FRL under the fairness constraints $\tau_1 = \tau_2 = 5\%$ and $\tau_1 = \tau_2 = 7\%$. From the experimental results, we can see that all FRL algorithms reduce the worst-class standard errors and robust errors under different degrees. FRL (Reweight) has the best performance to achieve the minimal "worst-class" standard error. Compared to vanilla methods, it has around 10% reduction to the worst class's standard error. However, it cannot equalize the boundary errors adequately. Alternatively, FRL (Remargin) is more effective than FRL (Reweight) to decrease the worst class boundary error. Furthermore, their combination FRL (Reweight + Remargin) is the most effective way to reduce the worst-class boundary and worst-class robust errors. It can accomplish to train a robust classifier with the worst-class robust error around 70%, which is $10 \sim 15\%$ lower than vanilla adversarial training methods. The baseline method (Baseline Reweight) (Agarwal et al., 2018) can only help decrease the worst-class standard error but cannot equalize boundary errors or robust errors. These results suggest that the FRL method can mitigate the fairness issues by improving the model's standard and robustness performance on the worst classes.

Notably, from the results in Table 2, we find that those methods, which use the strategy "Remargin", usually have a $1 \sim 2\%$ larger average standard error, compared to the PGD Adversarial Training or TRADES ($1/\lambda = 1$). This is because "Remargin" increases the perturbation margin ϵ for some classes during training. According to the existing works (Tramèr et al., 2020; Shafahi et al., 2019), using a large perturbation margin might degrade the model's standard performance generalization. Thus, in this work we limit every training sample's perturbation margin does not exceed $16/255$ to guarantee that the model gives an acceptable average standard performance. On the other hand, because the "Remargin" strategies increase the perturbation margin, their average boundary errors are smaller than the vanilla methods. As a result, the average robust errors are comparable with the vanilla methods or even have $2 \sim 3\%$ improvement.

In Table 3, we present the experimental results on the SVHN dataset. We have similar observations as those on CIFAR10. FRL (Reweight) can achieve the minimal worst-class standard error and FRL (Reweight+Remargin) gives the minimum worst-class robust error. One intriguing fact is that FRL methods, including those which use "Remargin", do not cause an increase of the average standard error. Instead, each FRL method results in $1 \sim 2\%$ decrease of the average standard error. This might be due to the reason that these methods help improve the worst-class standard error by a large margin.

5.3. Ablation Study

From the results in Table 2, we find that FRL (Reweight) is not effective to improve the worst-class boundary error. As a result, it cannot sufficiently equalize the robustness performance between classes. In this subsection, we study the potential reasons that cause this fact. To have a closer look at this issue, we implement adversarial training on CIFAR10 dataset in two groups of experiments following the basic settings described in Section 5.1. For each group, we only upweight a chosen class's boundary error by different ratios (from 1.0 to 4.5 times of other classes) and keep other classes' weights fixed. Then, for each model, we present this model's standard / robustness performance for this class in Figure 3 (left). As a comparison, we also show the cases where we increase the perturbation margin (1.0 to 2.5 times of the original margin) (Figure 3 (right)). From Figure 3, we find that when we increase the weight only for the class "deer", it results in the boundary error for this class to decrease but also increasing its standard error. Thus,



(a) Upweight Boundary Error (b) Increase Perturbation Margin

Figure 3. The effect of upweighting on boundary error (left) and standard error (right) for the class "deer".

Table 2. Average & worst-class standard error, boundary error and robust error for various algorithms on CIFAR10.

	Avg. Std.	Worst Std.	Avg. Bndy.	Worst Bndy.	Avg. Rob.	Worst Rob.
PGD Adv. Training	15.5	33.8	40.9	55.9	56.4	82.7
TRADES$(1/\lambda = 1)$	**14.6**	31.2	43.1	64.6	57.7	84.7
TRADES$(1/\lambda = 6)$	19.6	39.1	**29.9**	49.5	**49.3**	77.6
Baseline Reweight	19.2	28.3	39.2	53.7	58.2	80.1
FRL(Reweight, 0.05)	16.0	**22.5**	41.6	54.2	57.6	73.3
FRL(Remargin, 0.05)	16.9	24.9	35.0	50.6	51.9	75.5
FRL(Reweight+Remargin, 0.05)	17.0	26.8	35.7	**44.5**	52.7	**69.5**
FRL(Reweight, 0.07)	16.1	23.8	38.4	55.2	54.0	75.2
FRL(Remargin, 0.07)	16.9	26.0	37.4	51.6	53.5	75.1
FRL(Reweight+Remargin, 0.07)	17.1	26.7	36.7	48.3	53.8	70.2

Table 3. Average & worst-class standard error, boundary error and robust error for various algorithms on SVHN.

	Avg. Std.	Worst Std.	Avg. Bndy.	Worst Bndy.	Avg. Rob.	Worst Rob.
PGD Adv. Training	9.4	19.8	37.0	53.9	46.4	73.7
TRADES$(1/\lambda = 1)$	9.9	18.6	39.1	60.6	48.0	78.3
TRADES$(1/\lambda = 6)$	10.5	23.4	**32.5**	52.5	**43.1**	76.6
Baseline Reweight	8.8	17.4	39.3	54.7	48.2	72.1
FRL(Reweight, 0.05)	7.9	**13.3**	38.2	56.4	46.1	69.7
FRL(Remargin, 0.05)	9.2	17.4	39.7	**49.6**	48.9	67.0
FRL(Reweight+Remargin, 0.05)	**7.7**	12.8	36.2	51.2	43.9	**64.0**
FRL(Reweight, 0.07)	8.0	13.6	37.2	54.2	45.0	67.8
FRL(Remargin, 0.07)	8.5	14.2	36.9	50.6	45.5	64.8
FRL(Reweight+Remargin, 0.07)	8.3	15.4	36.7	51.4	45.0	64.9

reweight only acts to leverage the inner-class boundary error and standard error. It cannot improve the robustness of this class over other classes to solve the fairness relationship between classes. As a contrast, increasing the margin will not increase the standard error while it effectively decreases this class's boundary error and robust error. Therefore, from the results in Table 2 and 3, we observe that "Remargin" based methods can successfully improve the worst-class robustness performance.

6. Related Work

Adversarial Attacks and Adversarial Training. The existence of adversarial attacks (Goodfellow et al., 2014; Szegedy et al., 2013; Carlini & Wagner, 2017) causes huge concerns when people adopt machine learning models in various application domains (Xu et al., 2019; Jin et al., 2020). As countermeasures against adversarial examples, adversarial training (robust optimization) algorithms (Goodfellow et al., 2014; Madry et al., 2017; Zhang et al., 2019b; Shafahi et al., 2019; Zhang et al., 2019a) are formulated as a min-max problem that directly minimize the model's risk on the adversarial samples. Another mainstream of defense methods are certified defense, which aims to provide provably robust DNNs under l_p norm bound (Wong & Kolter, 2018; Cohen et al., 2019) and guarantee the robustness.

Fairness in Machine Learning & Imbalanced Dataset. Fairness issues recently draw much attention from the community of machine learning. These issues can generally divided into two categorizations: (1) prediction outcome disparity (Zafar et al., 2017); and (2) prediction quality disparity (Buolamwini & Gebru, 2018). Unlike existing works, this work is the first study the unfairness issue in the adversarial setting. We also mention the imbalanced data learning problem (He & Garcia, 2009; Lin et al., 2017) as one related topic of our work. Since in our work, (i.e.. Figure 1), we show that the prediction performance differences are indeed existing between different classes. This phenomenon is also well studied in imbalanced data problems or long-tail distribution learning problems (Wang et al., 2017) where some classes have much fewer training samples than others. However, in our case, we show that this unfairness problem can generally happen in balanced datasets, so it desires new scopes and methods for further study.

Fairness in Robust Learning A parallel and independent work (Nanda et al., 2020) also figures out that the phenomenon of class-wise unequal robustness can happen for many deep learning tasks in the wild. While our work is more focused on adversarial training algorithms and we argue that adversarial training methods can have the property to cause these fairness phenomena. In addition, our work also discusses various potential mitigation methods to achieve more balanced robustness for adversarial training methods.

7. Conclusion

In this work we first empirically and theoretically uncover one property of adversarial training algorithms: it can cause serious disparity for both standard accuracy and adversarial robustness between different classes of the data. As the first attempt to mitigate the fairness issues from adversarial training, we propose the Fair Robust Learning (FRL) framework. We validate the effectiveness of FRL on benchmark datasets. In the future, we want to examine if the fairness issue can be observed in other types of defense methods.

References

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., and Wallach, H. A reductions approach to fair classification. *arXiv preprint arXiv:1803.02453*, 2018.

Buolamwini, J. and Gebru, T. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pp. 77–91, 2018.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019.

Ding, G. W., Sharma, Y., Lui, K. Y. C., and Huang, R. Max-margin adversarial (mma) training: Direct input space margin maximization through adversarial training. *arXiv preprint arXiv:1812.02637*, 2018.

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

He, H. and Garcia, E. A. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

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

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pp. 125–136, 2019.

Jin, W., Li, Y., Xu, H., Wang, Y., and Tang, J. Adversarial attacks and defenses on graphs: A review and empirical study. *arXiv preprint arXiv:2003.00653*, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

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

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

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

Morgulis, N., Kreines, A., Mendelowitz, S., and Weisglass, Y. Fooling a real car with adversarial traffic signs. *arXiv preprint arXiv:1907.00374*, 2019.

Nanda, V., Dooley, S., Singla, S., Feizi, S., and Dickerson, J. P. Fairness through robustness: Investigating robustness disparity in deep learning. *arXiv preprint arXiv:2006.12621*, 2020.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pp. 3358–3369, 2019.

Sharif, M., Bhagavatula, S., Bauer, L., and Reiter, M. K. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pp. 1528–1540, 2016.

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.

Tramèr, F., Behrmann, J., Carlini, N., Papernot, N., and Jacobsen, J.-H. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. In *International Conference on Machine Learning*, pp. 9561–9571. PMLR, 2020.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Wang, Y.-X., Ramanan, D., and Hebert, M. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pp. 7029–7039, 2017.

Wong, E. and Kolter, Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pp. 5286–5295. PMLR, 2018.

Xu, H., Ma, Y., Liu, H., Deb, D., Liu, H., Tang, J., and Jain, A. Adversarial attacks and defenses in images, graphs and text: A review. *arXiv preprint arXiv:1909.08072*, 2019.

Zafar, M. B., Valera, I., Gomez Rodriguez, M., and Gummadi, K. P. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pp. 1171–1180, 2017.

Zhang, B. H., Lemoine, B., and Mitchell, M. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 335–340, 2018.

Zhang, D., Zhang, T., Lu, Y., Zhu, Z., and Dong, B. You only propagate once: Painless adversarial training using maximal principle. *arXiv preprint arXiv:1905.00877*, 2 (3), 2019a.

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

Zhang, Z., Song, Y., and Qi, H. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5810–5818, 2017.

A. Appendix

A.1. More Results of Preliminary Studies

In this section, we show more preliminary results on the robust fairness phenomenon of adversarial training in various settings. In addition to the results shown in Section 2, we present the results in the settings with one more architecture (WRN28), one more type of adversarial attack (l_2-norm attack), one more defense method (Randomized Smoothing) and one more dataset (SVHN). From all these settings, we observe the similar phenomenon as in Section 2, which show that the fairness phenomenon can be generally happening in adversarial training under different scenarios and can be a common concern during its application. Furthermore, we also present the detailed results as in Table 1, to show the fact that adversarial training usually gives an unequal influence on different classes, which can be a reason that causes this fairness phenomenon.

In detail, for each dataset under PreAct ResNet18 architecture, for each adversarial training algorithm (including PGD-adversarial training (Madry et al., 2017) and TRADES (Zhang et al., 2017)), we train the models following that as suggested by the original papers. We train the models for 200 epochs with learning rate 0.1 and decay the learning rate at the epoch 100 and 150 by factor 0.1. During the evaluation phase, we report the trained model's classwise *standard error* rate and *robust error* rate. In general settings without explicit mention, we study the models' robustness against l_∞-norm adversarial attack under $8/255$, where we implement PGD attack algorithm for 20 steps for robustness evaluation.

A.1.1. Robust Fairness in WRN28 in CIFAR10

Figure 5 presents robust fairness issues in CIFAR10 dataset under WRN28 models. Note that in Section 2 we also presented the corresponding results under PreAct ResNet18 models in Figure 1. We can observe the similar phenomenon about the robust fairness issues under both models. Moreover, as clear evidence of the unequal effect of adversarial training among different classes, in Table 4 and Table 5, we compare the classwise standard error and robust error between natural training and PGD adversarial training. From the experimental results, we get the conclusion that adversarial training usually increases a larger error rate for the classes, such as "dog" and "cat", which originally have larger errors in natural training. Similarly, adversarial training will also give less help to reduce the robust errors for these classes.

A.1.2. Robust Fairness in l_2-norm Adversarial Training

Figure 6 presents the robust fairness issues of adversarial training methods which target on l_2-norm attacks in CIFAR10 dataset. We further confirm the existence of robust unfairness in adversarial training methods. In Figure 6, we present the classwise standard errors and robust errors, which target on l_2-norm 0.5 adversarial attack. During the robustness evaluation, we implement PGD attack algorithm with step size 0.1 for 20 steps.

A.1.3. Robust Fairness for Certified Defenses

Certified defenses are another main type of effective defense strategies. Even though certified defenses do not train in the same way as traditional adversarial training methods, which train the models on the generated adversarial examples, they minimize the probability of the existence of adversarial examples near the input data. This process also implicitly minimizes the model's overall robust error. Thus, in this section we study whether this certified defense will have robust fairness issues. As a representative, we implement Randomized Smoothing (Cohen et al., 2019), which is one state-of-the-art methods to certifiably defense against l_2-norm adversarial attacks. In this experiment, we run Randomized Smoothing against l_2-norm 0.5 attacks in CIFAR10 dataset and report its class-wise certified standard error and certified robust error under different intensities.

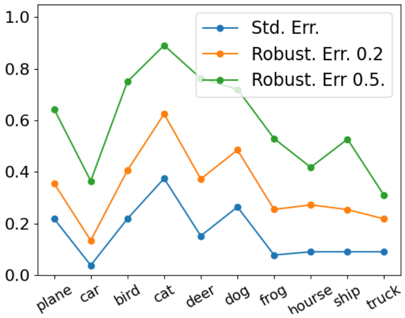


Figure 4. Randomized Smoothing on CIFAR10

The results also suggest that the Randomized Smoothing certified defense method presents the similar disparity phenomenon as the traditional adversarial training methods. Moreover, it also preserves the similar classwise performance relationship, i.e., it has both high standard & robustness error on the classes "cat" and "dog", but has relatively low errors on "car" and "ship".

A.1.4. Robust Fairness on SVHN Dataset

Figure 7 presents the robust fairness issues of adversarial training methods in SVHN dataset under PreAct ResNet18 model. From the experimental results, we also observe the strong disparity of classwise standard errors and robust errors, which do not exist in natural training. In particular, the classes "3" and "8" have the largest standard error in a naturally trained model. After adversarial training, these two classes also have the largest standard error increases among all classes, as well as the least robust error decreases. As a result, there is also a significant disparity of the standard / robustness performance among the classes. The full results are shown in Table 6.

A.2. Theoretical Proof of Section 3

In this section, we formally calculate the classwise standard & robust errors in an optimal linear classifier and an optimal linear robust classifier. Then, we present our main conclusion that robust optimization will unequally influence the performance of the two classes and therefore result in a severe performance disparity.

A.2.1. Proof of Theorem 1

In this subsection, we study an optimal linear classifier which minimizes the average standard error. By calculating its standard errors, we can get the conclusion that the class "+1" in distribution $\mathcal{D}$ is indeed harder than class "-1". We first start from a lemma to calculate the weight vector of an optimal linear model.

Lemma 1 (Weight Vector of an Optimal Classifier) *For the data following the distribution $\mathcal{D}$ defined in Eq. 2, an optimal linear classifier f_{nat} which minimizes the average standard classification error:*

$$\begin{aligned} f_{nat}(x) &= sign(\langle w_{nat}, x\rangle + b_{nat}) \\ where \;\; w_{nat}, b_{nat} &= \arg\min_{w,b} \; Pr.(sign(\langle w, x\rangle + b) \neq y) \end{aligned}$$

has the optimal weight that satisfy: $w_{nat} = \mathbf{1}$.

Proof 2 (Proof of Lemma 2) *In the proof we will use $w = w_{nat}$ and $b = b_{nat}$ for simplicity. Next, we will prove $w_1 = w_2 = \cdots = w_d$ by contradiction. We define $G = \{1, 2, \ldots, d\}$ and make the following assumption: for the optimal w and b, we assume if there exist $w_i < w_j$ for $i \neq j$ and $i, j \in G$. Then we obtain the following standard errors for two classes of this classifier with weight w:*

$$\begin{aligned} \mathcal{R}_{nat}(f; -1) = Pr\{ &\sum_{k\neq i, k\neq j} w_k \mathcal{N}(-\eta, \sigma_{-1}^2) + b \\ &+ w_i \mathcal{N}(-\eta, \sigma_{-1}^2) + w_j \mathcal{N}(-\eta, \sigma_{-1}^2) > 0\} \\ \mathcal{R}_{nat}(f; +1) = Pr\{ &\sum_{k\neq i, k\neq j} w_k \mathcal{N}(+\eta, \sigma_{+1}^2) + b \\ &+ w_i \mathcal{N}(+\eta, \sigma_{+1}^2) + w_j \mathcal{N}(+\eta, \sigma_{+1}^2) < 0\} \end{aligned} \tag{12}$$

However, if we define a new classier $\tilde{f}$ whose weight vector $\tilde{w}$ uses w_j to replace w_i, we obtain the errors for the new classifier:

$$\begin{aligned} \mathcal{R}_{nat}(\tilde{f}; -1) = Pr\{ &\sum_{k\neq i, k\neq j} w_k \mathcal{N}(-\eta, \sigma_{-1}^2) + b \\ &+ w_j \mathcal{N}(-\eta, \sigma_{-1}^2) + w_j \mathcal{N}(-\eta, \sigma_{-1}^2) > 0\} \\ \mathcal{R}_{nat}(\tilde{f}; +1) = Pr\{ &\sum_{k\neq i, k\neq j} w_k \mathcal{N}(+\eta, \sigma_{+1}^2) + b \\ &+ w_j \mathcal{N}(+\eta, \sigma_{+1}^2) + w_j \mathcal{N}(+\eta, \sigma_{+1}^2) < 0\}. \end{aligned} \tag{13}$$

By comparing the errors in Eq 12 and Eq 13, it can imply the classifier $\tilde{f}$ has smaller error in each class. Therefore, it contradicts with the assumption that f is the optimal classifier with least error. Thus, we conclude for an optimal linear classifier in natural training, it must satisfies $w_1 = w_2 = \cdots = w_d$ and $w = \mathbf{1}$.

Given the results in Lemma 1, we can calculate the errors of classifiers by only calculating the interception term b_{nat} and b_{rob}. Recall Theorem 1, we explicitly calculate the classwise errors of an optimal classifier which minimizes the average standard error.

Theorem 1 *For a data distribution $\mathcal{D}$ in Eq. 2, for the optimal linear classifier f_{nat} which minimizes the average standard classification error, it has the intra-class standard error for the two classes:*

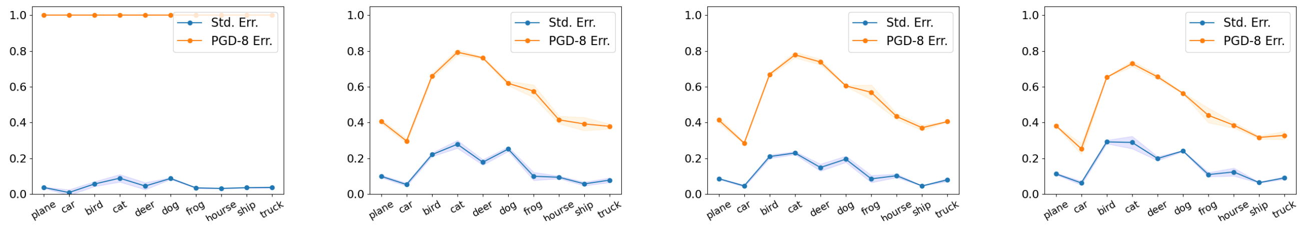
$$\begin{aligned} \mathcal{R}_{nat}(f_{nat}, -1) &= Pr.\{\mathcal{N}(0,1) \leq A - K \cdot \sqrt{A^2 + q(K)}\} \\ \mathcal{R}_{nat}(f_{nat}, +1) &= Pr.\{\mathcal{N}(0,1) \leq -K \cdot A + \sqrt{A^2 + q(K)}\} \end{aligned}$$

where $A = \frac{2}{K^2-1}\frac{\sqrt{d}\eta}{\sigma}$ and $q(K) = \frac{2\log K}{K^2-1}$ which is a positive constant and only depends on K, As a result, the class "+1" has a larger standard error:

$$\mathcal{R}_{nat}(f_{nat}, -1) < \mathcal{R}_{nat}(f_{nat}, +1).$$

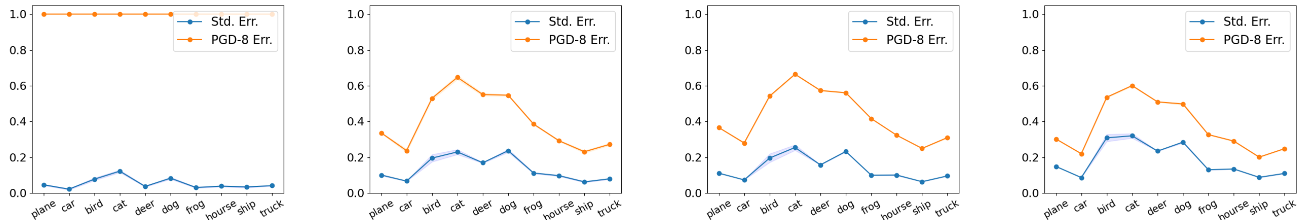
Proof 3 (Proof of Theorem 1) *From the results in Lemma 1, we define our optimal linear classifier to be $f_{nat}(x) = sign(\sum_{i=1}^{d} x_i + b_{nat})$. Now, we calculate the optimal b_{nat} which can minimize the average standard error:*

$$\begin{aligned} R_{nat}(f) &= Pr.\{f(x) \neq y\} \\ &\propto Pr\{f(x) = 1 | y = -1\} + Pr\{f(x) = -1 | y = 1\} \\ &= Pr.\{\sum_{i=1}^{d} x_i + b_{nat} > 0 | y = -1\} \\ &+ Pr\{\sum_{i=1}^{d} x_i + b_{nat} < 0 | y = +1\} \\ &= Pr.\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{\sigma} + \frac{1}{\sqrt{d}\sigma} \cdot b_{nat}\} \\ &+ Pr.\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{K\sigma} - \frac{1}{K\sqrt{d}\sigma} \cdot b_{nat}\} \end{aligned} \tag{14}$$



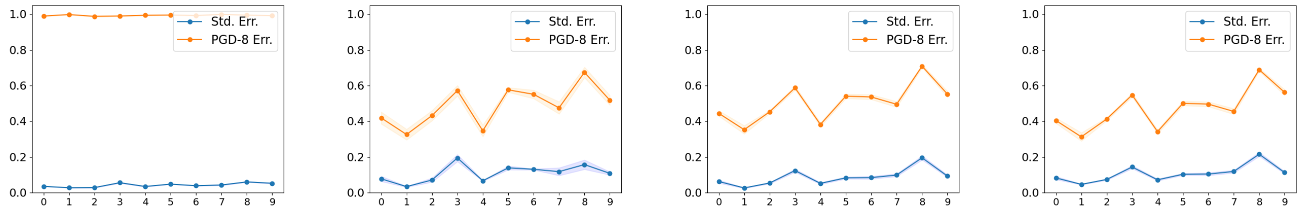
(a) Natural Training (b) PGD Adversarial Training (c) TRADES $(1/\lambda = 1)$ (d) TRADES $(1/\lambda = 6)$

Figure 5. The class-wise performance of natural / adversarial training (target on l_∞-norm 8/255 attack) on CIFAR10 under WRN28.



(a) Natural Training (b) PGD Adversarial Training (c) TRADES $(1/\lambda = 6)$ (d) TRADES $(1/\lambda = 6)$

Figure 6. The class-wise performance of natural / adversarial training (target on l_2-norm 0.5 attack) on CIFAR10 under PreActResNet18



(a) Natural Training (b) PGD Adversarial Training (c) TRADES $(1/\lambda = 1)$ (d) TRADES $(1/\lambda = 6)$

Figure 7. The class-wise performance of natural / adversarial training (target on l_∞-norm 8/255 attack) on SVHN under PreActResNet18.

The optimal b_{nat} to minimize $\mathcal{R}_{nat}(f)$ is achieved at the point that $\frac{\partial \mathcal{R}_{nat}(f)}{\partial b_{nat}} = 0$. Thus, we find the optimal b_{nat}:

$$b_{nat} = \frac{K^2+1}{K^2-1} \cdot d\eta - K\sqrt{\frac{4d^2\eta^2}{(K^2-1)^2} + q(K)d\sigma^2} \quad (15)$$

and $q(K) = \frac{2\log K}{K^2-1}$ which is a positive constant and only depends on K. By incorporating the optimal b_{nat} into Eq. 17, we can get the classwise standard errors for the two classes:

$$\mathcal{R}_{nat}(f_{nat}, -1) = Pr.\{\mathcal{N}(0,1) \leq A - K \cdot \sqrt{A^2 + q(K)}\}$$
$$\mathcal{R}_{nat}(f_{nat}, +1) = Pr.\{\mathcal{N}(0,1) \leq -K \cdot A + \sqrt{A^2 + q(K)}\}$$

where $A = \frac{2}{K^2-1}\frac{\sqrt{d}\eta}{\sigma}$. Since $q(K) > 0$, we have the direct conclusion that $\mathcal{R}_{nat}(f; -1) < \mathcal{R}_{nat}(f; +1)$.

A.2.2. Proof of Theorem 2

In this subsection, we focus on calculating the errors of robust classifiers which minimize the average robust errors of the model. By comparing natural classifiers and robust classifiers, we get the conclusion that robust classifiers can further exacerbate the model's performance on the "harder" class. Similar to Section A.2.1, we start from a Lemma to show an optimal robust classifier f_{rob} has a weight vector $w_{\text{rob}} = \mathbf{1}$.

Lemma 2 (Weight Vector of an Optimal Robust Classifier) *For the data following the distribution $\mathcal{D}$ defined in Eq. 2, an optimal linear classifier f_{nat} which minimizes the*

Table 4. The Changes of Standard & Robust Error in Natural & Adversarial Training in CIFAR10 on PreAct ResNet18.

Std. Error	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
Natural Training	4.0	1.8	6.4	11.3	6.1	10.0	5.1	5.2	3.5	4.3
PGD Adv. Training	11.7	6.1	23.3	34.8	20.8	26.9	12.6	9.8	6.4	9.5
Diff. (Adv. - Nat.)	7.7	4.3	16.9	23.5	14.6	16.9	7.5	4.6	2.9	5.2

Rob. Error	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
Natural Training	100	100	100	100	100	100	100	100	100	100
PGD Adv. Training	44.9	34.3	68.4	82.7	74.7	66.4	51.5	47.0	40.8	42.3
Diff. (Adv. - Nat.)	-55.1	-65.7	-31.6	-17.3	-25.3	-33.5	-48.5	-53.0	-59.2	-57.7

Table 5. The Changes of Standard & Robust Error in Natural & Adversarial Training in CIFAR10 on WRN28.

Std. Error	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
Natural Training	3.9	1.9	6.4	10.0	6.2	9.0	5.0	4.6	3.3	4.2
PGD Adv. Training	10.0	4.9	21.4	24.6	17.4	26.2	12.4	9.4	6.3	9.2
Diff. (Adv. - Nat.)	6.1	3.0	15.0	14.6	11.2	17.2	7.4	4.8	3.0	5.0

Rob. Error	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
Natural Training	100	100	100	100	100	100	100	100	100	100
PGD Adv. Training	41.4	29.3	65.8	77.2	75.5	61.6	60.9	40.7	40.8	39.4
Diff. (Adv. - Nat.)	-58.6	-70.7	-34.2	-22.8	-24.5	-38.4	-39.1	-59.3	-59.2	-60.6

Table 6. The Changes of Standard & Robust Error in Natural & Adversarial Training in SVHN in PreAct ResNet18.

Std. Error	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
Natural Training	3.6	2.7	3.2	5.8	4.0	5.1	3.6	4.6	6.1	5.3
PGD Adv. Training	8.8	6.2	7.9	15.8	6.9	13.2	13.3	13.4	19.8	11.4
Diff. (Adv. - Nat.)	5.2	3.5	4.8	10.1	4.9	8.1	9.6	8.9	13.6	6.4

Std. Error	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
Natural Training	100	100	100	100	100	100	100	100	100	100
PGD Adv. Training	47.2	38.8	49.0	63.4	41.9	57.1	62.6	55.2	73.7	57.1
Diff. (Adv. - Nat.)	-52.8	-61.2	-51.1	-36.6	-58.1	-42.9	-37.4	-44.8	-26.3	-42.9

average standard classification error:

$$\begin{aligned} f_{rob}(x) &= sign(\langle w_{rob}, x\rangle + b_{rob}) \\ \textit{where}\ \ w_{rob}, b_{rob} &= \underset{w,b}{\arg\min}\ Pr.(\exists \delta, ||\delta|| \leq \epsilon, \\ &\qquad s.t.\ sign(\langle w, x+\delta\rangle + b) \neq y). \end{aligned}$$

has the optimal weight which satisfy: $w_{rob} = \mathbf{1}$.

We leave the detailed proof out in the paper because it can be proved in the similar way as the proof of Lemma 1. Recall Theorem 2, we formally calculate the standard errors of an optimal robust linear classifier.

Theorem 2 *For a data distribution $\mathcal{D}$ in Eq. 2, the optimal robust linear classifier f_{rob} which minimizes the average robust error with perturbation margin $\epsilon < \eta$, it has the intra-class standard error for the two classes:*

$$\begin{aligned} &\mathcal{R}_{nat}(f_{rob}, -1) \\ =&Pr.\{\mathcal{N}(0,1) \leq B - K\cdot\sqrt{B^2 + q(K)} - \frac{\sqrt{d}}{\sigma}\epsilon\} \\ &\mathcal{R}_{nat}(f_{rob}, +1) \\ =&Pr.\{\mathcal{N}(0,1) \leq -K\cdot B + \sqrt{B^2 + q(K)} - \frac{\sqrt{d}}{K\sigma}\epsilon\} \end{aligned} \tag{16}$$

where $B = \frac{2}{K^2-1}\frac{\sqrt{d}(\eta-\epsilon)}{\sigma}$ and $q(K) = \frac{2\log K}{K^2-1}$ is a positive constant and only depends on K,

Proof 4 (Proof of Theorem 2) *From the results in Lemma 2, we define our optimal linear robust classifier to be $f_{rob}(x) = sign(\sum_{i=1}^{d} x_i + b_{rob})$. Now, we calculate the*

Table 7. Average & worst-class standard error, boundary error and robust error for various algorithms on CIFAR10 under WRN28.

	Avg. Std.	Worst Std.	Avg. Bndy.	Worst Bndy.	Avg. Rob.	Worst Rob.
PGD Adv. Training	14.0	29.3	38.1	53.0	52.2	78.8
TRADES$(1/\lambda = 1)$	**12.6**	25.2	40.2	58.7	52.8	76.7
TRADES$(1/\lambda = 6)$	15.5	29.1	**31.8**	**45.7**	**47.3**	71.8
Baseline Reweight	14.2	26.3	38.6	53.7	52.8	77.9
FRL(Reweight, 0.05)	14.5	**23.2**	40.0	53.3	54.4	76.8
FRL(Remargin, 0.05)	15.4	24.9	38.1	49.6	53.5	70.5
FRL(Reweight+Remargin, 0.05)	15.4	25.0	37.8	46.7	53.2	**67.1**
FRL(Reweight, 0.07)	14.1	23.8	39.5	54.1	53.6	77.0
FRL(Remargin, 0.07)	14.8	24.3	39.5	50.6	54.3	73.0
FRL(Reweight+Remargin, 0.07)	14.9	24.7	37.8	48.3	52.7	68.2

Table 8. Average & worst-class standard error, boundary error and robust error for various algorithms on SVHN under WRN28.

	Avg. Std.	Worst Std.	Avg. Bndy.	Worst Bndy.	Avg. Rob.	Worst Rob.
PGD Adv. Training	8.1	16.8	38.5	57.3	46.7	71.2
TRADES$(1/\lambda = 1)$	8.0	19.6	40.1	60.0	48.1	73.3
TRADES$(1/\lambda = 6)$	10.6	23.1	**32.1**	52.5	**42.7**	70.6
Baseline Reweight	8.5	16.2	40.3	57.8	48.8	71.1
FRL(Reweight, 0.05)	**7.8**	13.4	38.9	56.9	46.7	70.7
FRL(Remargin, 0.05)	8.4	13.4	40.8	52.1	49.2	65.5
FRL(Reweight+Remargin, 0.05)	8.4	**13.2**	38.4	52.1	46.8	**63.1**
FRL(Reweight, 0.07)	8.2	13.5	41.2	56.3	49.4	69.8
FRL(Remargin, 0.07)	8.6	14.9	38.8	51.2	47.4	67.0
FRL(Reweight+Remargin, 0.07)	8.2	13.9	39.9	**50.2**	48.1	65.4

optimal b_{rob} which can minimize the average robust error:

$$\begin{aligned}
\mathcal{R}_{rob}(f) =& Pr.(\exists||\delta|| \le \epsilon \;\; s.t. \;\; f(x+\delta) \neq y) \\
=& \max_{||\delta|| \le \epsilon} Pr.(f(x+\delta) \neq y) \\
=& \frac{1}{2} Pr.(f(x+\epsilon) \neq -1 | y = -1) \\
&+ \frac{1}{2} Pr.(f(x-\epsilon) \neq +1 | y = +1) \\
=& Pr.\{\sum_{i=1}^{d} (x_i + \epsilon) + b_{rob} > 0 | y = -1\} \\
&+ Pr\{\sum_{i=1}^{d} (x_i - \epsilon) + b_{rob} < 0 | y = +1\} \\
=& Pr.\{\mathcal{N}(0,1) < -\frac{\sqrt{d}(\eta-\epsilon)}{\sigma} + \frac{1}{\sqrt{d}\sigma} \cdot b_{rob}\} \\
&+ Pr.\{\mathcal{N}(0,1) < -\frac{\sqrt{d}(\eta-\epsilon)}{K\sigma} - \frac{1}{K\sqrt{d}\sigma} \cdot b_{rob}\}
\end{aligned} \tag{17}$$

The optimal b_{rob} to minimize $\mathcal{R}_{rob}(f)$ is achieved at the point that $\frac{\partial \mathcal{R}_{rob}(f)}{\partial b_{rob}} = 0$. Thus, we find the optimal b_{rob}:

$$b_{rob} = \frac{K^2+1}{K^2-1} \cdot d(\eta-\epsilon) - K\sqrt{\frac{4d^2(\eta-\epsilon)^2}{(K^2-1)^2} + q(K)d\sigma^2} \tag{18}$$

and $q(K) = \frac{2\log K}{K^2-1}$ which is a positive constant and only depends on K. By incorporating the optimal b_{nat} into Eq. 17, we can get the classwise robust errors for the two classes:

$$\begin{aligned}
\mathcal{R}_{rob}(f_{rob}, -1) &= Pr.\{\mathcal{N}(0,1) \le B - K \cdot \sqrt{B^2 + q(K)}\} \\
\mathcal{R}_{rob}(f_{rob}, +1) &= Pr.\{\mathcal{N}(0,1) \le -K \cdot B + \sqrt{B^2 + q(K)}\}
\end{aligned}$$

where $B = \frac{2}{K^2-1}\frac{\sqrt{d}(\eta-\epsilon)}{\sigma}$. As a direct result, the classwise standard errors for the two classes:

$$\begin{aligned}
&\mathcal{R}_{nat}(f_{rob}, -1) \\
=& Pr.\{\mathcal{N}(0,1) \le B - K \cdot \sqrt{B^2 + q(K)} - \frac{\sqrt{d}}{\sigma}\epsilon\} \\
&\mathcal{R}_{nat}(f_{rob}, +1) \\
=& Pr.\{\mathcal{N}(0,1) \le -K \cdot B + \sqrt{B^2 + q(K)} - \frac{\sqrt{d}}{K\sigma}\epsilon\}.
\end{aligned}$$

A.2.3. Proof of Corollary 1

Giving the results in Theorem 1 and Theorem 2, we will show that a robust classifier will exacerbate the performance of the class "+1" which originally has higher error in a naturally trained model. In this way, we can get the conclusion that robust classifiers can cause strong disparity, because it exacerbates the "difficulty" difference among classes.

Corollary 1 *Adversarially Trained Models on $\mathcal{D}$ will increase the standard error for class "+1" and reduce the standard error for class "-1":*

$$\begin{aligned}
\mathcal{R}_{nat}(f_{rob}, -1) < \mathcal{R}_{nat}(f_{nat}, -1). \\
\mathcal{R}_{nat}(f_{rob}, +1) > \mathcal{R}_{nat}(f_{nat}, +1).
\end{aligned}$$

Proof 5 (Proof of Corollary 1) *From the intermediate results in Eq.15 and Eq. 18 in the proofs of Theorem 1 and Theorem 2, we find the only difference between a naturally trained model f_{nat} and a robust model f_{rob} is about the interception term b_{nat} and b_{rob}. Specifically, we denote $g(\cdot)$ is the function of the interception term, then we have the results:*

$$b_{nat} = \frac{K^2+1}{K^2-1}\cdot d\eta - K\sqrt{\frac{4d^2\eta^2}{(K^2-1)^2}+q(K)d\sigma^2} := g(\eta)$$

$$\begin{aligned} b_{rob} = \frac{K^2+1}{K^2-1}\cdot d(\eta-\epsilon) - K\sqrt{\frac{4d^2(\eta-\epsilon)^2}{(K^2-1)^2}+q(K)d\sigma^2} \\ := g(\eta-\epsilon) \end{aligned}$$

Next, we show that the function $g(\cdot)$ is a monotone increasing function between 0 and η:

$$\frac{dg(\eta)}{d\eta} \geq \frac{K^2+1}{K^2-1}d - K\frac{\frac{4}{(K^2-1)^2}d^2\cdot 2\eta}{2\sqrt{\frac{4}{(K^2-1)^2}d^2\eta^2}} = \frac{K-1}{K+1}d > 0$$

As a direct results, we have the interception terms: $b_{nat} > b_{rob}$. This will result a linear classifier $f(x) = sign(\langle \mathbf{1}^T, x\rangle + b)$ present more samples in the overall distribution to be class "-1". As a result, we have the conclusion:

$$\mathcal{R}_{nat}(f_{rob}, -1) < \mathcal{R}_{nat}(f_{nat}, -1).$$
$$\mathcal{R}_{nat}(f_{rob}, +1) > \mathcal{R}_{nat}(f_{nat}, +1).$$

A.3. Robust / Non-Robust Features

In Section 3.1, we discussed a theoretical example where adversarial training will unequally treat the two classes in the distribution $\mathcal{D}$, which will increase the standard error of one class and decrease the error for the other one. However, in the real applications of deep learning models, we always observe that each class's error will increase after adversarial training. In this subsection, motivated by the work (Tsipras et al., 2018; Ilyas et al., 2019), we extend the definition of $\mathcal{D}$, to split the features into two categories: robust features and non-robust features, where adversarial training will increase the standard errors for both classes. Formally, the data distribution $\mathcal{D}'$ is defined as as following:

$$\begin{aligned} & y \overset{u.a.r}{\sim} \{-1,+1\}, \quad \theta = (\overbrace{\eta,...,\eta}^{\text{dim = d}}, \overbrace{\gamma,...,\gamma}^{\text{dim = m}}), \\ & x \sim \begin{cases} \mathcal{N}(\theta, \sigma_{+1}^2 I) & \text{if } y=+1 \\ \mathcal{N}(-\theta, \sigma_{-1}^2 I) & \text{if } y=-1 \end{cases} \end{aligned} \tag{19}$$

where in the center vector θ, it includes robust features with scale $\eta > \epsilon$, and non-robust features with scale $\gamma < \epsilon$. Here we specify that non-robust feature space has much higher dimension than robust feature space ($m >> d$) and there is a K-factor difference between the variance of two classes: $\sigma_{+1} = K \cdot \sigma_{-1}$. From the main results in the work (Tsipras et al., 2018), it is easy to get that each class's standard error will increase after adversarial training. In the following theory, we will show that in distribution $\mathcal{D}'$, adversarial training will increase the error for the class "+1" by a larger rate than the class "-1".

Theorem 3. *For a data distribution $\mathcal{D}'$ in Eq. 19, the robust optimizer f_{rob} increases the standard error of class "+1" by a larger rate than the increase of the standard error of class "-1":*

$$\begin{aligned} \mathcal{R}_{nat}(f_{rob}; +1) - \mathcal{R}_{nat}(f_{nat}; +1) > \\ \mathcal{R}_{nat}(f_{rob}; -1) - \mathcal{R}_{nat}(f_{nat}; -1) \end{aligned} \tag{20}$$

Proof 6 (Proof of Theorem 3) *The proof of Theorem 3 resembles the process of the proofs of Theorem 1 and Theorem 2, where we first calculate the classwise standard errors for each model. Note that from the work (Tsipras et al., 2018), an important conclusion is that a natural model f_{nat} uses both robust and non-robust features for prediction. While, a robust model f_{rob} only uses robust features for prediction (a detailed proof can be found in Section 2.1 in (Tsipras et al., 2018)). Therefore, we can calculate the classwise standard errors for both classes of a natural model f_{nat}:*

$$\mathcal{R}_{nat}(f_{nat}, -1) = Pr.\{\mathcal{N}(0,1) \leq A - K\cdot\sqrt{A^2+q(K)}\}$$
$$\mathcal{R}_{nat}(f_{nat}, +1) = Pr.\{\mathcal{N}(0,1) \leq -K\cdot A + \sqrt{A^2+q(K)}\}$$

where $A = \frac{2}{\sigma(K^2-1)}\sqrt{m\gamma^2+d\eta^2}$. The classwise standard errors of a robust model f_{rob} are:

$$\begin{aligned} & \mathcal{R}_{nat}(f_{rob}, -1) \\ =& Pr.\{\mathcal{N}(0,1) \leq B - K\cdot\sqrt{B^2+q(K)} - \frac{\sqrt{d}}{\sigma}\epsilon\} \\ & \mathcal{R}_{nat}(f_{rob}, +1) \\ =& Pr.\{\mathcal{N}(0,1) \leq -K\cdot B + \sqrt{B^2+q(K)} - \frac{\sqrt{d}}{K\sigma}\epsilon\} \end{aligned}$$

where $B = \frac{2}{\sigma(K^2-1)}\sqrt{d(\eta-\epsilon)^2}$. Next, we compare the standard error increase after adversarial training between the two classes. We have the results:

$$\begin{aligned} & (\mathcal{R}_{nat}(f_{rob}; +1) - \mathcal{R}_{nat}(f_{nat}; +1)) - (\mathcal{R}_{nat}(f_{rob}; -1) - \mathcal{R}_{nat}(f_{nat}; -1)) \\ & > (K+1)((A-B) + (\sqrt{B^2+q(K)} - \sqrt{A^2+q(K)})) \\ & \propto (\sqrt{B^2+q(K)} - B) - (\sqrt{A^2+q(K)} - A) \end{aligned}$$

because A includes high dimensional non-robust feature and $A >> B$, the equation above is positive and we get the conclusion as in Eq. 20.

A.4. Fairness Performance on WRN28

Table 7 and Table 8 presents the empirical results to validate the effectiveness of FRL algorithms under the WRN 28 model. The implementation details resemble those in Section 5.1. In the training, we start FRL from a pre-trained robust model (such as PGD-adversarial training), and run FRL with model parameter learning rate 1e-3 and hyperparameter learning rate $\alpha_1 = \alpha_2 = 0.05$ in the first 40 epochs. Then we decay the model parameter learning rate and the hyperparameter learning rate by 0.1 every 40 epochs. From the results, we have the similar observations as these for PreAct ResNet18 models, which is that FRL can help to improve the worst-class standard performance and robustness performance, such that the unfairness issue is mitigated. In particular, FRL (Reweight) is usually the most effective way to equalize the standard performance, but not sufficient to equalize the boundary errors and robust errors. FRL (Reweight + Remargin) is usually the most effective way to improve robustness for the worst class.