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 2, 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 5 and Table 6, 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.

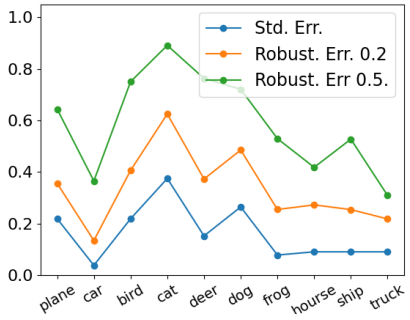


Figure 4. Randomized Smoothing on CIFAR10

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 7.

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:*

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

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 $\tilde{w}$ 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 calculate the classwise errors of an optimal classifier with minimal 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:*

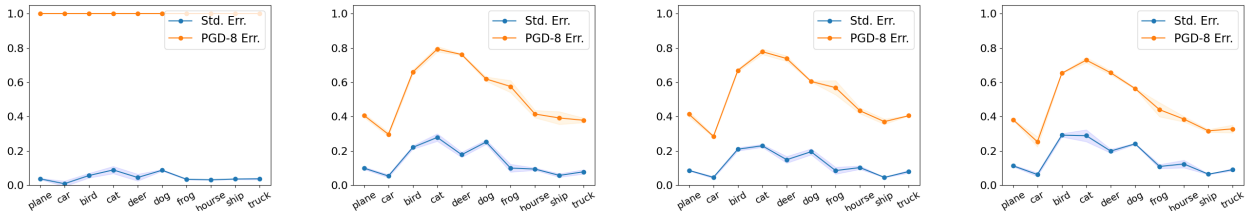
$$\begin{aligned}\mathcal{R}_{nat}(f_{nat},-1) &= Pr.\{\mathcal{N}(0,1) \leq A - K\cdot\sqrt{A^2+q(K)}\}\\ \mathcal{R}_{nat}(f_{nat},+1) &= Pr.\{\mathcal{N}(0,1) \leq -K\cdot A + \sqrt{A^2+q(K)}\}\end{aligned}$$

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:

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

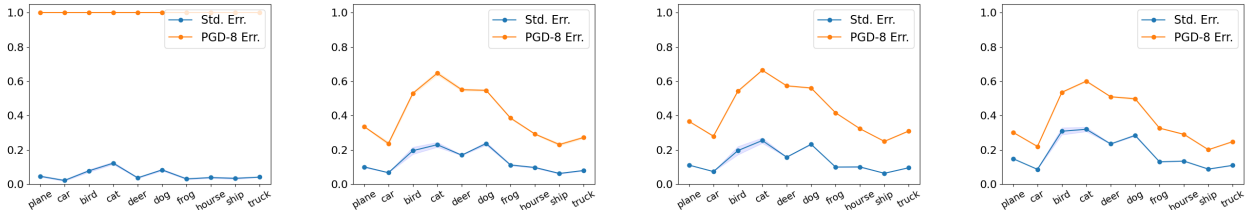
Proof 3 (Proof of Theorem 1) *From the results in Lemma 1, we define our optimal linear classifier to be $f_{nat}(x) = sign(\sum_{i=1}^d x_i + b_{nat})$. Now, we calculate the optimal b_{nat} which can minimize the average standard error:*

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



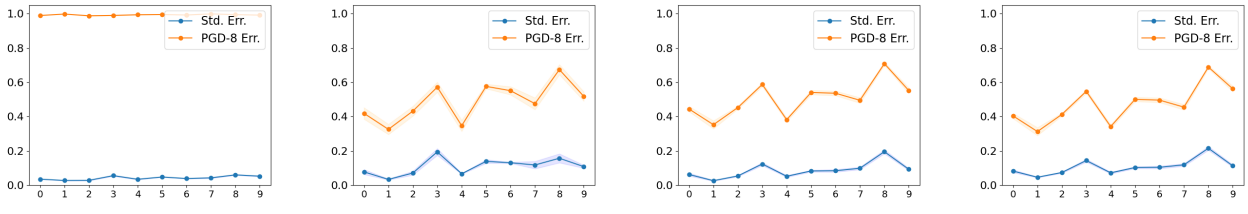
(a) Natural Training (b) PGD Adversarial Training (c) TRADES ($1/\lambda = 1$) (d) TRADES ($1/\lambda = 6$)

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 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 average standard classification error:*

$$f_{rob}(x) = sign(\langle w_{rob}, x\rangle + b_{rob})$$
$$where \;\; w_{rob}, b_{rob} = \arg\min_{w,b} \; Pr.(\exists \delta, ||\delta|| \leq \epsilon,$$
$$s.t.\; sign(\langle w, x+\delta\rangle + b) \neq y).$$

Table 5. 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 6. 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 7. 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

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

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

Table 8. 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 9. 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

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:

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

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":*

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

Proof 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:*

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

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

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

As a direct results, we have the interception terms: $b_{nat} > b_{rob}$. This will result a linear classifier $f(x) = sign(\langle \boldsymbol{I}^T, x\rangle + b)$ 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:

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

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":*

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

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 8 and Table 9 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.