The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

Improving Robust Fairness via Balance Adversarial Training

Chunyu Sun[1], **Chenye Xu**[1], **Chengyuan Yao**[1], **Siyuan Liang**[2], **Yichao Wu**[1], **Ding Liang**[1], **Xianglong Liu**[3, 4, 5], **Aishan Liu** [5*]

[1]SenseTime Research
[2]Institute of Information Engineering, Chinese Academy of Sciences
[3]Zhongguancun Laboratory, Beijing, China
[4]Institute of Dataspace, Hefei, Anhui, China
[5]NLSDE, Beihang University, Beijing, China
{sunchunyu,xuchenye,yaochengyuan,wuyichao,liangding}@sensetime.com,
liangsiyuan@iie.ac.cn, {xlliu, liuaishan}@buaa.edu.cn

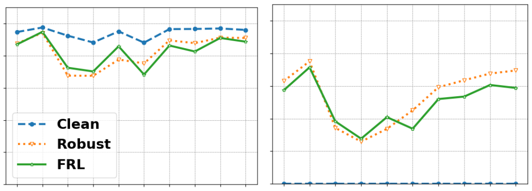
Abstract

Adversarial training (AT) methods are effective against adversarial attacks, yet they introduce severe disparity of accuracy and robustness between different classes, known as the *robust fairness problem*. Previously proposed Fair Robust Learning (FRL) adaptively reweights different classes to improve fairness. However, the performance of the better-performed classes decreases, leading to a strong performance drop. In this paper, we observed two unfair phenomena during adversarial training: different difficulties in generating adversarial examples from each class (source-class fairness) and disparate target class tendencies when generating adversarial examples (target-class fairness). From the observations, we propose Balance Adversarial Training (BAT) to address the robust fairness problem. Regarding source-class fairness, we adjust the attack strength and difficulties of each class to generate samples near the decision boundary for easier and fairer model learning; considering target-class fairness, by introducing a uniform distribution constraint, we encourage the adversarial example generation process for each class with a fair tendency. Extensive experiments conducted on multiple datasets (CIFAR-10, CIFAR-100, and ImageNette) demonstrate that our BAT can significantly outperform other baselines in mitigating the robust fairness problem (+5-10% on the worst class accuracy)(Our codes can be found at https://github.com/silvercherry/Improving-Robust-Fairness-via-Balance-Adversarial-Training).

Introduction

Deep neural networks (DNNs) are vulnerable to adversarial attacks (Szegedy et al. 2014; Goodfellow, Shlens, and Szegedy 2015a) which fool model predictions by adding imperceptible perturbations to natural examples. To defend against adversarial attacks, many defense techniques are designed (Xie et al. 2019; Cohen, Rosenfeld, and Kolter 2019; Jeong and Shin 2020). In particular, adversarial training (Madry et al. 2018; Zhang et al. 2019) that injects adversarial examples during training has been proved to be the most effective methods against adversarial attacks.

However, adversarial training suffers from the *robust fairness problem*, where the adversarially trained models make a severe disparity in accuracy and robustness among different classes (Xu et al. 2021). For example, an adversarially trained ResNet-18 model on CIFAR-10 has significantly lower clean and robust accuracy on class `cat` than other classes; in contrast, each class has a similar accuracy during the standard training (see Figure 1). This phenomenon is firstly defined by (Xu et al. 2021) and further theoretically justified by studying a binary classification task under a Gaussian mixture distribution. To mitigate the robust fairness problem, they proposed Fair-Robust-Learning (FRL), which adaptively re-weights each class during training to balance the performance of each class. However, at a closer inspection, we found that this robust fairness is achieved by reducing the performance of other previously better performed classes, leading to a reduction in both clean and robust accuracy (Figure 1).



(a) Clean accuracy (b) Robust accuracy

Figure 1: AT suffers from robust fairness problem where adversarially trained models make a severe disparity in accuracy and robustness among different classes compared to the standard training. FRL improves the previously poor performance classes, but other classes are decreased.

In this paper, we conjecture that the mechanisms of the adversarial example generation process during adversarial training are related to the robust fairness problem, which cannot be mitigated by a class re-weighting scheme. More specifically, we found two key observations that are fundamental to the robust fairness during AT as (1) *source-class fairness*: samples from different classes have different difficulties and require different perturbation budgets for adversarial example generation; (2) *target-class fairness*: the targets of the generated adversarial examples are biased and yield a clear tendency towards specified classes. Motivated by the above observation, we propose the Balance Adversarial Training

[*]*Corresponding author

(BAT) framework to mitigate the robust fairness problem by simultaneously addressing source-class and target-class fairness issues. To mitigate source-class fairness, we balance the strength of adversarial attacks on each class with adaptive perturbations so that we could bring samples to decision boundaries which would be easier and fairer for models to learn; to balance target-class fairness, we force the generated adversarial examples to follow a uniform distribution towards target classes, so that we could yield a fairer classifier not influenced by the tendency of adversarial targets. Extensive experiments have been conducted on CIFAR-10, CIFAR-100, and ImageNette, demonstrating that BAT improves robust fairness while preserving both accuracy and robustness. In particular, our method outperforms other baselines by large margins and improves the worst class error rate of 6.31% on average. Our **contributions** can be summarized as:

- We discover the source-class and target-class fairness phenomena as the related cause of the robust fairness problem for AT.
- Based on our observation, we propose a novel AT framework named BAT to mitigate the fairness problem, where we balance the source-class and target-class fairness.
- Extensive experiments on several datasets have been conducted, which demonstrate the superiority of our approach compared to other baselines.

Related Work

Adversarial Attacks

Adversarial attacks are inputs intentionally designed to mislead deep learning models but are imperceptible to humans (Szegedy et al. 2014; Goodfellow, Shlens, and Szegedy 2015b). A long line of work has been proposed to attack deep learning models (Goodfellow, Shlens, and Szegedy 2015b; Kurakin, Goodfellow, and Bengio 2016a; Liu et al. 2019, 2020b,a). In general, it can be roughly divided into white-box attacks and black-box attacks. In the white-box scenario, attackers have complete knowledge of the target model and often generate attacks using the model gradient (Goodfellow, Shlens, and Szegedy 2015b; Madry et al. 2018; Carlini and Wagner 2017); as for the black-box scenario, attackers have limited model knowledge and could often only obtain the model output (Ilyas et al. 2018; Narodytska and Kasiviswanathan 2017; Andriushchenko et al. 2019). In this paper, we follow the commonly-studied setting (Zhang et al. 2019, 2020, 2021b) and mainly focus on defending the more challenging white-box adversarial attacks.

Adversarial Training

Among the adversarial defenses (Xie et al. 2019; Cohen, Rosenfeld, and Kolter 2019; Jeong and Shin 2020; Qin et al. 2019; Zhang et al. 2021a), adversarial training (Kurakin, Goodfellow, and Bengio 2016b) that injects adversarial examples during training has been proved to be one of the most effective methods against adversarial attacks. Madry et al. formulated the adversarial training as a min-max optimization issue and utilize PGD attack (Madry et al. 2018) to solve the inner maximization for generating adversarial examples. This method makes a notable advance, and many variants of adversarial training are based on a similar min-max framework (Zhang et al. 2019; Wang et al. 2020; Wu, Xia, and Wang 2020). Though promising, Xu et al. found that AT introduces severe disparity of clean and robust accuracy between different classes, which is formulated as the robustness fairness problem. As a preliminary study, they were motivated by (Buolamwini and Gebru 2018; Zafar et al. 2017; Agarwal et al. 2018) and used a re-weight and re-margin framework to finetune a robust model to improve the previously poor classes. However, they decrease the performance of other classes and make the overall accuracy (both clean and robustness) drop. In this paper, we primarily focus on better understanding and mitigating the robust fairness problem. Specifically, we discover the source-class and target-class fairness phenomena and further propose the BAT framework.

Methodology

In this section, we introduce the BAT framework to mitigate the robust fairness problem. We first clarify definitions and symbols in Section ; we then show the source-class and target class fairness phenomena during AT in Section ; finally, in Section , we propose novel and effective BAT methods against the robust fairness problem in AT.

Preliminaries and Notations

In this paper, we consider the image classification task. Let $f_\theta : \mathbf{x} \rightarrow \mathbb{R}^K$ represents a deep neural network classifier parameterized by θ, where K denotes the number of the output classes, θ denotes the model parameters.

Input Space. Let $\mathcal{D} \subset \mathbb{R}^d$ be the input space. Consider an input feature $\mathbf{x}_i \in \mathcal{D}$ and a label $\mathbf{y}_i$ is the input space $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$.

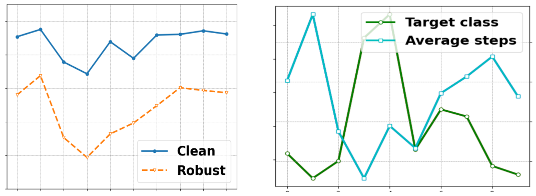
Adversarial Example. We use $\mathbf{x}_{adv} = \mathbf{x} + \delta$ to denote adversarial examples, where $||\delta||_p \leq \epsilon$. The added perturbation δ could make DNNs misclassify the input into wrong labels, *i.e.*, $f_\theta(\mathbf{x} + \delta) \neq f_\theta(\mathbf{x})$.

Adversarial Training. Given an input image $(\mathbf{x}_i, \mathbf{y}_i)$, a model f_θ and a loss function ℓ, we aim to build robust DNNs through the adversarial training scheme by solving the min-max optimization problem as

$$\min_\theta \sum_{i=1}^n \max_\delta \ell(f_\theta(\mathbf{x}_i + \delta, \mathbf{y}_i)). \tag{1}$$

Source-Class Fairness and Target-Class Fairness

In this section, we first illustrate the source-class and target-class fairness phenomena for adversarial training and then draw the relation between source&target-class fairness and robust fairness. For the adversarially-trained model, we select ResNet-18 (He et al. 2016a) on CIFAR-10 using PGD adversarial training (Madry et al. 2018); we use the untargeted PGD-ℓ_∞ attack with 10 steps under $\epsilon = 8/255$ budgets and the step size as $2/255$ for confusion matrices and 1000 steps under $\epsilon = 8/255$ budgets and the step size as $0.4/255$ for calculating the average of attack steps. *More details are in the supplementary materials.*



(a) (b)

Figure 2: Source-class fairness and Target-class fairness are both related to the robust fairness. (a): robust fairness of AT; (b): the average number of attack steps of source-class and the distribution probability of target-class.

Source-Class Fairness Source-class fairness is defined as the different difficulties of adversarial example generation from each class. There are two ways to measure it quantitatively. The first is to calculate the class-wise average number of attack steps required to cause misclassification. This is a natural way of measuring, and it reflects the distance of the decision boundary from the clean example. The second way is to fix the attack strength and count robust samples of each class, and we can use the diagonal of the confusion matrix to present the quantity. We can calculate the matrix after each attack step during the attack, and the matrix from the final attack step represents the very notion of robust fairness. We can see that the first measure integrates the second measure over the attack steps dimension. In Figure 2, we empirically observe the correlation between the average number of attack steps and class-wise robust accuracy.

Target-Class Fairness Target-class Fairness is defined as the target class tendencies when generating adversarial examples. This quantity is calculated by the distance from the class distribution of generated adversarial examples to uniform distribution (as shown in Figure 2), and the class distribution can be calculated from the sum of the row columns in the confusion matrix. In Figure 2, we see the inverse correlation between the distribution probability density and the relative class robust performance. We conjecture that this quantity is closely tied to robust fairness, which should be addressed during adversarial training.

Here we establish the correlation between source/target class fairness and robust fairness. We conjecture that addressing the source/target fairness problems in adversarial training is important to robust fairness. Therefore, we propose a new adversarial training paradigm BAT that considers source-class and target-class fairness simultaneously.

Balance Adversarial Training

In this section, we introduce our proposed Balance Adversarial Training (BAT) framework as shown in Figure 3, where we balance both source-class and target-class fairness.

Balance Source-Class Fairness We attempt to balance the number of attack steps required to break the model by adjusting the attacking strength of each class with different perturbations. Studies (Rade and Moosavi-Dezfooli 2021; Zhang et al. 2021b) have revealed that excessive perturbations are difficult for models to fit and cause a performance drop. Therefore, we bring these samples to the decision boundaries, which would be easier and fairer for models to learn. Intuitively, for some classes that are difficult to generate adversarial examples, we should add stronger perturbations; conversely, classes that are easy to attack require fewer perturbations so that they would not be so far and "hard" to learn.

Based on the above analysis, we translate the difficulty of adversarial generation (*i.e.*, perturbation) to the distance to decision boundaries and define two types of boundary examples. Given sample $\mathbf{x}$, let $\Phi(\mathbf{x})$ denotes the maximum steps to the decision boundary, thus we have $\mathbf{x}^{\Phi}_{clean}$ as the **last clean example** and $\mathbf{x}^{\Phi}_{adv}$ as the **first adversarial example**. Specifically, $\mathbf{x}^{\Phi}_{clean}$ denotes the "last" instance that can be rightly classified by models after adding perturbations, and $\mathbf{x}^{\Phi}_{adv}$ denotes the "first" instance that is misclassified by models after perturbing. Therefore, these two types of perturbed examples are located close to the decision boundaries, which can be referred to as *boundary examples*. For each class, we adversarially perturb their samples with different strengths (perturbations) to generate the two types of boundary examples so that we could ensure that each class contains both misclassified and correctly classified samples with similar learning hardness for models. Therefore, based on the standard AT framework of TRADES (Zhang et al. 2019), we can improve the source-class fairness using $\mathcal{L}_{\texttt{source-class}}$ as

$$\mathcal{L}_{\texttt{source-class}} = \min_{\theta} \sum_{i=1}^{n} \{CE(f_\theta(\mathbf{x}^{\Phi}_{clean,i}), \mathbf{y}_i) + \beta \max KL(f_\theta(\mathbf{x}_i) f_\theta(\mathbf{x}^{\Phi}_{adv,i}))\}, \tag{2}$$

where CE is the cross-entropy loss, KL is the Kullback–Leibler (KL) divergence, and β is a balancing parameter. $\mathcal{L}_{\texttt{source-class}}$ could balance the difficulties of adversarial example generation from source classes by avoiding generating an excess of adversarial samples for easy-to-attack classes or too few adversarial samples for hard-to-attack classes.

Balance Target-Class Fairness After balancing the source class fairness, our next goal is to eliminate the biased tendencies of target classes for adversarial example generation. In other words, we need to generate adversarial examples with similar confidences or probabilities towards different classes. That is, we aim to learn a fair classifier not influenced by the tendency of adversarial targets. Formally, the generated adversarial examples should follow a uniform distribution

$$\min_{\theta} \sum_{i=1}^{n} \{KL(\mathcal{U} f_\theta(\mathbf{x}_i)) + KL(\mathcal{U} f_\theta(\mathbf{x}_{adv,i}))\}, \tag{3}$$

where $\mathcal{U}$ is a uniform distribution of samples. Inspired by fair adversarial training (Zafar et al. 2017), we have the following Lemma describing that the fairness of the AT process will be influenced by different adversarial perturbations.

Lemma 1. (Du and Wu 2021) *The fair classifier f that minimizes the cross entropy loss $CE(f_\theta(\mathbf{x}_i + \delta), \mathbf{y}_i)$ subject to $RD(\mathbb{D}) \leq \tau$, where $RD(\mathbb{D})$ is the risk difference over a biased distribution $\mathbb{D}$ of $X \times \Delta \times Y$.*

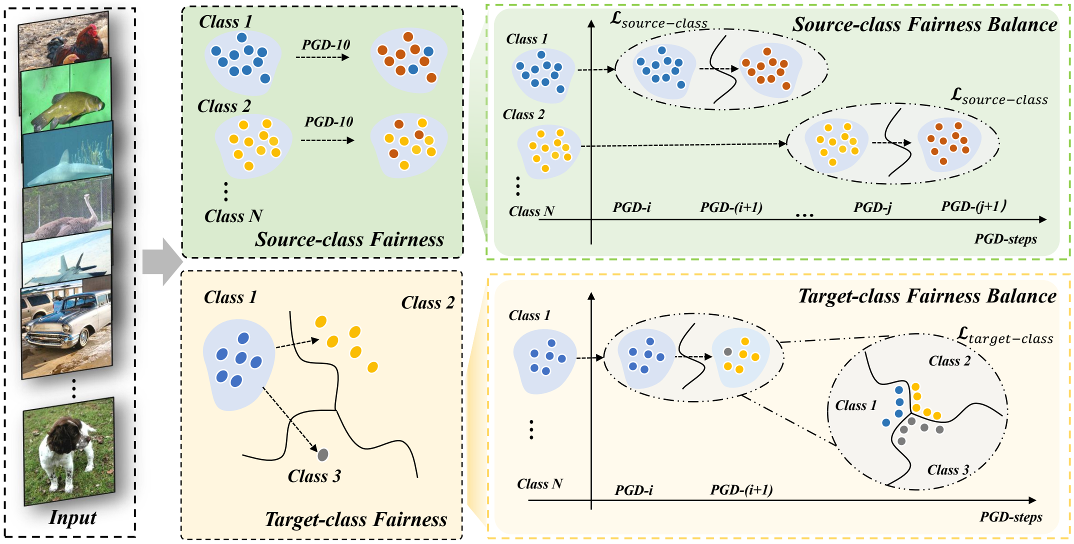


Figure 3: Framework overview. To mitigate source-class fairness, we generate adaptive perturbations where we balance the strength of adversarial attack on each class, so that we could bring these samples to the decision boundaries which would be easier and fairer for models to learn. To balance target-class fairness, we force the generated adversarial examples follow a uniform distribution, so that we could yield a fairer classifier not influenced by the tendency of adversarial targets.

In Lemma 1, X denotes the input features, Δ denotes the set of different adversarial perturbations, and Y denotes the label set. To make Lemma 1 reach the optimal solution (*i.e.*, training a fair classifier f), we can approximate the fairness constraints RD by using the boundary fairness as follows:

$$C_{\mathbb{D}}(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\delta_i - \hat{\delta}_i)d_\theta(\mathbf{x}_i), \tag{4}$$

where δ_i is the adversarial perturbation of $\mathbf{x}_i$, $\hat{\delta}_i$ denotes the mean value of the different adversarial perturbations (different steps of attacks) added on $\mathbf{x}_i$, and $d_\theta(\mathbf{x}_i)$ indicates the distance of $\mathbf{x}_i$ to the classifier boundary of f.

According Lemma 1 and the boundary fairness, the overall function can be written as

$$\mathcal{L}_{\mathtt{total}} = \mathcal{L}_{\mathtt{source-class}} + \alpha(\frac{1}{n}\sum_{i=1}^{n}(\delta_i - \hat{\delta}_i)d_\theta(\mathbf{x}_i) - \tau)^2, \tag{5}$$

where α is the trade off parameter. Since the distance between $\mathbf{x}^{\Phi}_{clean}$ and $\mathbf{x}^{\Phi}_{adv}$ to $\hat{\delta}$ is closer than the distance between the clean example x and the maximum adversarial example $\mathbf{x}_{adv}$ (generated by the fixed and largest perturbations), the value of the fairness loss is less.

In this way, we notice the target-class fairness is related to the boundary samples and Eq.(3) can be rewritten as

$$\mathcal{L}_{\mathtt{target-class}} = \min_{\theta}\sum_{i=1}^{n}\{KL(\mathcal{U}f_\theta(\mathbf{x}^{\Phi}_{clean,i}) + KL(\mathcal{U}f_\theta(\mathbf{x}^{\Phi}_{adv,i})\}. \tag{6}$$

To sum up, by uniforming the distribution of boundary examples, we could improve the target-class fairness, so that the target tendency of poor-performing classes could be reduced and the well-performing classes could be improved.

Overall Training Based on the above analysis, we then illustrate the overall training of our BAT framework (*c.f.* Algorithm 1). In particular, we dynamically adjust the perturbation size to the boundary samples to balance Source-class fairness; we adopt the standard min-max framework and uniform distribution constraint to the boundary samples to further balance Target-class fairness. The overall training objective is shown as

$$\mathcal{L}_{\mathtt{total}} = \mathcal{L}_{\mathtt{source-class}} + \alpha\mathcal{L}_{\mathtt{target-class}}, \tag{7}$$

where α is the balancing parameter. In Algorithm 1, $\mathcal{N}(\mathbf{0}, \mathbf{I})$ generates a random unit vector of d dimension, ξ is a small constant. For each min-batch of data $B = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{m}$, we use white-box untargeted PGD attacks to generate adversarial examples. Under the maximum PGD step K, we stop it when the samples are attacked successfully $(\arg\max_i f(\tilde{\mathbf{x}}_i) \neq \mathbf{y}_i)$ and get $\mathbf{x}^{\Phi}_{clean}$ and $\mathbf{x}^{\Phi}_{adv}$.

Experiments

In this section, we first illustrate our experimental setups; we then compare our method with other baselines; finally, we conduct ablation studies to better understand our framework.

Experimental Setups

Datasets and architectures. We choose the commonly-used datasets including CIFAR-10/100 and ImageNette. We use ResNet-18 (He et al. 2016b) and WRN-28-10 (Zagoruyko and Komodakis 2016) architectures in our experiments.

Compared Baselines. We compare the previously proposed method FRL (Xu et al. 2021) which is the only method that address robust fairness problem to the best of our knowledge. We also consider the standard adversarial training methods, including PGD adversarial training (PGD-AT) (Madry et al. 2018) and TRADES (Zhang et al. 2019).

Algorithm 1: Balance Adversarial Training

```
Input: training data $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$, batch of samples $B = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{m}$, model $f_\theta$, loss function $\ell_{KL}$, maximum PGD step $K$, perturbation $\epsilon$, step size $\alpha$, number of epochs $T$, learning rate $\eta$
Output: robustness network $f_\theta$
for epoch $= 1, \dots, T$ do
    Sample a mini-batch $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{m}$ from $B$
    $\mathbf{x}^{\Phi}_{clean,i} \leftarrow \mathbf{x}_i, \mathbf{x}^{\Phi}_{adv,i} \leftarrow \mathbf{x}_i$
    $\tilde{\mathbf{x}_i} \leftarrow \mathbf{x}_i + \xi\mathcal{N}(\mathbf{0}, \mathbf{I})$
    while $K > 0$ do
        if $\arg\max_i f(\tilde{\mathbf{x}_i}) \neq y_i$ then
            break
        else
            $\mathbf{x}^{\Phi}_{clean,i} \leftarrow \tilde{\mathbf{x}_i}$
            $\tilde{\mathbf{x}_i} \leftarrow \Pi_{\mathcal{B}[\mathbf{x}_i,\epsilon]}\left(\alpha(\nabla_{\tilde{\mathbf{x}_i}}\ell_{KL}(f(\tilde{\mathbf{x}_i}), f(\mathbf{x}_i)) + \tilde{\mathbf{x}_i}\right)$
            $\mathbf{x}^{\Phi}_{adv,i} \leftarrow \tilde{\mathbf{x}_i}$
            $K \leftarrow K - 1$
        end if
    end while
    $\theta \leftarrow \theta - \eta\frac{1}{m}\sum_{i-1}^{m}\nabla_\theta\left[\ell_{source-class}(\mathbf{x}^{\Phi}_{clean,i}, \mathbf{x}^{\Phi}_{adv,i}) + \alpha\ell_{target-class}(\mathbf{x}^{\Phi}_{clean,i}, \mathbf{x}^{\Phi}_{adv,i})\right]$
end for
```

Implementation Details. We use the published codes for TRADES (Zhang et al. 2019)[1], FRL (Xu et al. 2021)[2]. For FRL, we use the Reweight+Remargin under the $\tau = 0.05$ and $\tau = 0.07$ which perform best on its all settings; for the other adversarial training methods, we align our setting to the robustness benchmarks (Croce et al. 2020; Tang et al. 2021), and set the $\epsilon = 8/255$, step size $2/255$, and the maximum number of steps as 10. We keep the architecture and main hyper-parameters the same for BAT and other baselines.

Adversarial Attacks. In this paper, we follow (Xu et al. 2021) and use PGD attacks regarding cross entropy loss with 20 steps and step size of $2/255$ to evaluate the robust fairness in our main experiment. In addition, we also adopt AutoAttack (Croce and Hein 2020) to better evaluate the robustness of our method (*c.f.* supplementary material).

Evaluation Metrics. For fair comparisons, we follow (Xu et al. 2021) and use the average and worst-class error rate of standard (Avg. Std. & Worst Std.), boundary and robustness (Avg. Bndy. & Worst Bndy. and Avg. Rob. & Worst Rob.) to evaluate the robust fairness. These metrics comprehensively measure whether the model provides equal prediction quality among each class. For all these metrics, the lower the better. *The more details show on the supplementary materials.*

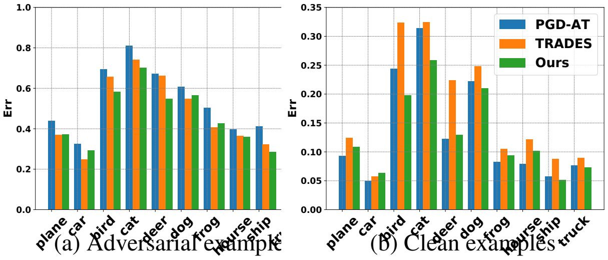
Comparison with Baseline Methods

In this section, we evaluate the robust fairness performance on ours and other baselines. Due to the space limitation, we only report the results of ResNet18 on CIFAR-10/100 and ImageNette in the main body of our paper. *More results of different model architectures can be found in the supplementary materials.* Based on the results shown in Table 1, we can draw the following **observations** and **conclusions**.

(1) For robust fairness (*i.e.*, Worst Std., Worst Bndy., Worst Rob.), our BAT consistently outperforms other baselines by large margins on all three datasets. Compared to PGD-AT, it has around **6%**, **10%** and **10%** reduction to the worst class standard error, boundary error, and robust error on CIFAR-10; for FRL, we improve the standard error, boundary error, and robust error on average **1.3%**, **6.8%**, **8.1%**. More specifically, we demonstrate the class-wise performance on clean and adversarial examples in Figure 4. We can observe that BAT could significantly improve the performance on `bird`, `cat` and `deer` (previously poor classes) while achieving better or similar performance on other classes. These results demonstrate the superiority of our BAT in mitigating robustness fairness problem during adversarial training.

(2) For accuracy and robustness (*i.e.*, Avg Std., Avg. Bndy., Avg. Rob.), our BAT achieves the best performance in almost all cases. The FRL framework improves the worst class errors compared to standard AT (PGD-AT and TRADES), but it decreases the average clean and robust accuracy. For example, considering FRL (Reweight+Remargin, 0.07), the worst standard, boundary, and robust errors both decline (*i.e.*, -3.9%, -6.2% and -5.0%), but the average standard and robust errors are improved (*i.e.*, +1.67% and +1.64%). Our BAT is able to avoid this drawback and leads to an overall improvement.

(3) Due to the trade-off between adversarial robustness and clean accuracy (Tsipras et al. 2019), our average clean accuracy (Avg. Std.) is slightly lower than TRADES($1/\lambda = 1$), which is designed to balance the clean/robust accuracy. However, our BAT maintains a comparatively high clean accuracy with fairer robust performance. For instance, compared to TRADES under $1/\lambda = 1$ which is used to focus on better clean accuracy, our method shows slightly lower clean accuracy (0.09%), but we achieve significantly higher robustness (**-8.42%** on average robust errors) and fairer results on both worst class standard error, boundary error and robust error (*i.e.*, **-1.9%**, **-13.4%** and **-9.4%**); compared to TRADES under $1/\lambda = 6$, our BAT outperforms it on all metrics.



(a) Adversarial example

(b) Clean examples

Figure 4: Errors (%) on each class of CIFAR-10 with towards clean or adversarial examples with ResNet-18 trained under PGD-AT, TRADES ($1/\lambda = 6$), and BAT.

Ablation Studies

In this section, we provide ablation studies on our BAT. We keep the same settings with Section and use CIFAR-10.

[1] https://github.com/yaodongyu/TRADES

[2] https://github.com/hannxu123/fair_robust

Dataset	Method	Avg. Std.	Worst Std.	Avg. Bndy.	Worst Bndy.	Avg. Rob.	Worst Rob.
CIFAR-10	PGD-AT	13.43	31.40	39.47	54.90	52.90	81.20
	TRADES$(1/\lambda = 1)$	**12.82**	27.80	38.91	57.80	51.73	79.70
	TRADES$(1/\lambda = 6)$	15.79	36.30	31.60	45.60	47.39	73.80
	FRL(Reweight+Remargin, 0.05)	14.79	26.90	38.76	53.70	53.55	80.60
	FRL(Reweight+Remargin, 0.07)	15.10	27.50	36.16	48.70	51.26	76.20
	Ours (BAT)	12.91	**25.90**	**31.57**	**44.40**	**44.48**	**70.30**
CIFAR-100	PGD-AT	40.40	80.00	36.95	57.00	77.35	98.00
	TRADES$(1/\lambda = 1)$	**40.14**	79.00	38.01	58.00	78.15	99.00
	TRADES$(1/\lambda = 6)$	44.14	81.00	28.75	53.00	72.89	97.00
	FRL(Reweight+Remargin, 0.05)	43.20	82.00	29.00	49.00	73.68	97.00
	FRL(Reweight+Remargin, 0.07)	46.50	83.00	28.88	51.00	75.38	97.00
	Ours (BAT)	40.25	**79.00**	**28.68**	**47.00**	**70.93**	**94.00**
ImageNette	PGD-AT	34.26	46.10	32.92	41.60	67.18	84.20
	TRADES$(1/\lambda = 1)$	27.22	39.60	36.55	50.20	63.77	83.40
	TRADES$(1/\lambda = 6)$	29.99	41.10	27.42	38.30	57.41	81.90
	FRL(Reweight+Remargin, 0.05)	28.05	40.90	32.00	42.30	60.05	81.30
	FRL(Reweight+Remargin, 0.07)	28.28	39.60	31.82	42.60	60.10	81.00
	Ours (BAT)	**26.80**	**39.30**	**30.24**	**37.60**	**57.04**	**80.10**

Table 1: Results on CIFAR-10/100 and Imagenette. Our BAT achieves the best robust fairness in almost all cases.

Source-class Loss	Target-class Loss	No.	Avg. Std.	Worst Std.	Avg. Bndy.	Worst Bndy.	Avg. Rob.	Worst Rob.
$\mathcal{L}_{\texttt{source-class}}(\mathbf{x}^{\Phi}_{clean}, \mathbf{x}^{\Phi}_{adv})$	*NA*	1	15.01	28.80	31.89	47.20	47.90	73.90
	$\mathcal{L}_{\texttt{Target-class}}(\mathbf{x}^{\Phi}_{clean})$	2	16.74	30.80	29.34	43.60	46.08	74.50
	$\mathcal{L}_{\texttt{Target-class}}(\mathbf{x}^{\Phi}_{adv})$	3	17.46	36.20	**26.23**	**41.30**	**43.69**	70.30
	$\mathcal{L}_{\texttt{Target-class}}(\mathbf{x}_{adv})$	4	17.51	37.20	26.59	42.80	44.10	72.80
	$\mathcal{L}_{\texttt{Target-class}}(\mathbf{x}^{\Phi}_{clean}, \mathbf{x}^{\Phi}_{adv})$	**5**	**12.91**	**25.90**	31.57	44.40	44.48	**70.30**
$\mathcal{L}_{\texttt{source-class}}(\mathbf{x}, \mathbf{x}_{adv})$	$KL(\mathcal{U} \| f_\theta(\mathbf{x}^{\Phi}_{clean})) + KL(\mathcal{U} \| f_\theta(\mathbf{x}^{\Phi}_{adv}))$	6	16.65	40.80	32.11	47.20	48.76	77.70

Table 2: Ablations on Source-class Loss and Target-class Loss. No. represent the number of experiment settings.

Source-Class Balance of BAT. Firstly, we study and ablate the Source-class Loss of our BAT framework. In our framework, we use the last clean sample $\mathbf{x}^{\Phi}_{clean}$ and the first adversarial sample $\mathbf{x}^{\Phi}_{adv}$ to conduct adversarial training for better source-class fairness balancing. Here, we use $\mathbf{x}$ and $\mathbf{x}_{adv}$ instead of our source-class loss, where $\mathbf{x}$ is the clean example and $\mathbf{x}_{adv}$ is the adversarial example generated with fixed stable steps (*i.e.*, 10 step numbers). Thus, the optimization objective of $\mathcal{L}_{\texttt{source-class}}$ can be changed as follows:

$$\min_{\theta} \sum_{i=1}^{n} \left\{ CE(f_\theta(\mathbf{x}_i), \mathbf{y}_i) + \beta \max KL(f_\theta(\mathbf{x}_i) \| f_\theta(\mathbf{x}_{adv,i})) \right\}. \tag{8}$$

The hyper-parameter β and other settings are kept the same as our main experiment. From Table 2, we can observe that our Source-class loss with decision boundary samples achieves the best performance on all evaluation metrics (No.5 vs. No.6 in Table 2), which indicates that the decision boundary samples play a critical role in mitigating the robust fairness problem. In addition, we found that our Source-class loss has a good behavior on both standard accuracy (on average **+0.48%** of four settings) and the robust accuracy (on average **+2.68%** of four settings). This shows that introducing fixed perturbations for each class is harmful to the overall performance.

Target-Class Balance of BAT. Moreover, we ablate the Target-class Loss. Specifically, we first remove the target loss (No.1 in Table 2), we then remove the uniform distribution regularization for the *first adversarial samples* $\mathbf{x}^{\Phi}_{adv}$ (No.2 in Table 2) and *last clean sample* $\mathbf{x}^{\Phi}_{clean}$ (No.3 in Table 2), respectively; finally, we use $\mathbf{x}^{adv}$ instead of our target-class loss (No.4 in Table 2). For the samples of Target-class loss, we found that uses fixed stable steps would significantly increase the robust fairness problem, which indicates the importance of our Target-class loss. More precisely, uniformally regularizing $\mathbf{x}^{\Phi}_{clean}$ increases the clean accuracy while decreases the robustness; while $\mathbf{x}^{\Phi}_{adv}$ shows the inverse phenomenon. This phenomenon demonstrates that $\mathbf{x}^{\Phi}_{clean}$ and $\mathbf{x}^{\Phi}_{adv}$ are suffered from the trade-off between clean and robust accuracy. Our Target-class loss with both boundary samples has an overall improvement. Moreover, we found that only introducing Source-class loss would improve the fairness on clean data, while our Target-class loss could improve the fairness on perturbed examples. This may demonstrate that *Source-class balance focuses on the performance of clean examples, and Target-class balance is more concentrated on the fairness of robustness.* We further verify this in Section .

Analysis and Discussion

In this section, we present some analyses and discussions to better understand our BAT and the robust fairness problem.

	Avg. Std.	Worst Std.	Avg. Bndy.	Worst Bndy.	Avg. Rob.	Worst Rob.
FAT	**12.54**	27.80	38.98	57.70	51.52	83.20
GAIRAT	16.74	34.90	32.42	46.90	49.16	79.30

Table 3: Comparison with FAT and GAIRAT on CIFAR10. Our BAT shows better performance on all metrhics, indicating the importance of addressing both source-class and target-class fairness.

	Avg. Std.	Worst Std.	Avg. Bndy.	Worst Bndy.	Avg. Rob.	Worst Rob.
Wo2,3,4,5	5.15	6.43	30.52	38.23	35.67	49.28
Wo1,6,8,9	17.83	32.64	40.09	52.59	57.92	80.52

Table 4: Training with all dataset, training without previously poor classes (*i.e.*, 2, 3, 4, and 5), and training without previously good classes (*i.e.*, 1, 6, 8, and 9).

Is Instance-Reweighting AT Helpful?

We have shown the class-reweighting scheme by FRL is not actually useful to robust fairness, but how instance-reweighting in adversarial training affect robust fairness?

There exist several studies that exploit the instance-reweighting technique to better balance the clean/robustness trade-off of AT, and here we examine Friendly Adversarial Training (FAT) (Zhang et al. 2020) and Geometry-aware Instance-Reweighted Adversarial Training (GAIRAT) (Zhang et al. 2021b). FAT uses friendly adversarial samples, which are the least misclassified generated by attacks, while GAIRAT up-weights the boundary instance during AT. From Table 3, our BAT achieves better performance on robust fairness than FAT and GAIRAT (**1.9%**, **13.3%**, **12.9%** and **9.0%**, **2.5%**, **9.0%**) in terms of Worst Std., Worst Bndy., and Worst Rob. Meanwhile, BAT also shows better clean and robustness trade-off performance than these two methods.

We provide a closer inspection of these results. The key observation is that FAT and GAIRAT re-weighting directions are different: the former decreases the loss by boundary examples, while the latter increases it. From Table 3, we can see the trade-off between worst clean and worst robustness. If we refer to Table 1, we can see the performance of FAT is close to TRADES $1/\lambda = 1$, which has weak regularization of AT, and that of GAIRAT is close to PGD-AT, which adds more perturbation to adversarial examples. We can conclude that excessive perturbations would cause a performance drop in clean accuracy. Thus, the source-class fairness term in BAT may play a similar role to the instance reweighting scheme of FAT, and we see the improvement in terms of clean fairness (Section). This also verifies our hypothesis on source-class fairness, which is more closely related to clean accuracy. Thus, FAT and GAIRAT could only improve robust fairness to some extent due to the ignorance of target-class fairness. Despite that target fairness cannot be used alone to get robustness, combining the terms are important to the robustness measure in fairness, which makes BAT better than all baselines and other instance-reweighting methods.

Does the Devil Exist in the Dataset?

We notice that the robust fairness problem relates to the inherent difficulty in robust learning. Since AT models often show weak performance on specific classes within a dataset, we try to remove these classes from the dataset and then re-train models. In particular, we adversarially train ResNet-18 models using PGD-AT on CIFAR-10, where we erase previously poor classes (*i.e.*, 2, 3, 4, and 5) or previously good classes (*i.e.*, 1, 6, 8, and 9), respectively. As shown in Table 4, models trained on datasets without previously poor classes show an obvious decrease in worst class metrics, which indicates that the robust fairness problem is somewhat mitigated by removing the "hard" classes.

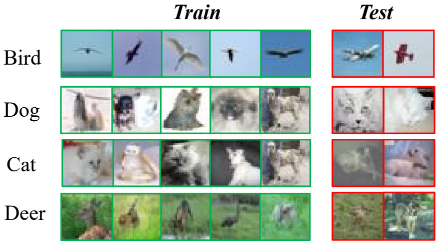


Figure 5: Bad case study. Test images that are failed by PGD-AT but correctly classified by our BAT are quite similar to the training examples of wrongly classified classes.

Some of the improvement in fairness comes from the mitigation of spurious correlation (Sagawa et al. 2020). We visualize some bad cases of these hard classes (*e.g.*, 2, 3, 4, and 5 in CIFAR-10) in Figure 5, where models trained by our BAT could correctly recognize these images while PGD-AT fails. We see PGD-AT suffers from the spurious correlation with the background. For example, the test image `Bird` has a similar blue background and two wings to the class `Plane`.

Conclusion

We find the correlation of robust fairness with source-class and target-class fairness. Based on the observation, we further propose Balance Adversarial Training to mitigate the robust fairness in adversarial training, where we simultaneously balance source-class and target-class fairness. Experiments demonstrate that BAT significantly improves robust fairness. In the future, we will study the generalization of robust fairness under more attacks and design metrics considering clean, robustness, and robust fairness.

Acknowledgements

This work was supported by The National Key Research and Development Program of China (2021YFF0602102), The National Key Research and Development Plan of China (2021ZD0110601, 2020AAA0103502), and National Natural Science Foundation of China (62022009, 61872021, and 62206009).

References

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

Andriushchenko, M.; Croce, F.; Flammarion, N.; and Hein, M. 2019. Square Attack: a query-efficient black-box adversarial attack via random search. *arXiv preprint arXiv:1912.00049*.

Buolamwini, J.; and Gebru, T. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*.

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

Cohen, J.; Rosenfeld, E.; and Kolter, Z. 2019. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*.

Croce, F.; Andriushchenko, M.; Sehwag, V.; Debenedetti, E.; Flammarion, N.; Chiang, M.; Mittal, P.; and Hein, M. 2020. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*.

Croce, F.; and Hein, M. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*.

Du, W.; and Wu, X. 2021. Robust fairness-aware learning under sample selection bias. *arXiv preprint arXiv:2105.11570*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015a. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations Workshop*.

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Identity mappings in deep residual networks. In *European Conference on Computer Vision*.

Ilyas, A.; Engstrom, L.; Athalye, A.; and Lin, J. 2018. Blackbox Adversarial Attacks with Limited Queries and Information. In *International Conference on Machine Learning*.

Jeong, J.; and Shin, J. 2020. Consistency Regularization for Certified Robustness of Smoothed Classifiers. In *Advances in Neural Information Processing Systems*.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016a. Adversarial examples in the physical world. arXiv:1607.02533.

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

Liu, A.; Huang, T.; Liu, X.; Xu, Y.; Ma, Y.; Chen, X.; Maybank, S.; and Tao, D. 2020a. Spatiotemporal Attacks for Embodied Agents. In *European Conference on Computer Vision*.

Liu, A.; Liu, X.; Fan, J.; Ma, Y.; Zhang, A.; Xie, H.; and Tao, D. 2019. Perceptual-sensitive gan for generating adversarial patches. In *AAAI Conference on Artificial Intelligence*.

Liu, A.; Wang, J.; Liu, X.; Cao, B.; Zhang, C.; and Yu, H. 2020b. Bias-based universal adversarial patch attack for automatic check-out. In *ECCV*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations Workshop*.

Narodytska, N.; and Kasiviswanathan, S. P. 2017. Simple Black-Box Adversarial Attacks on Deep Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*.

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

Rade, R.; and Moosavi-Dezfooli, S.-M. 2021. Helper-based adversarial training: Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *International Conference on Machine Learning*.

Sagawa, S.; Raghunathan, A.; Koh, P. W.; and Liang, P. 2020. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations Workshop*.

Tang, S.; Gong, R.; Wang, Y.; Liu, A.; Wang, J.; Chen, X.; Yu, F.; Liu, X.; Song, D.; Yuille, A.; et al. 2021. Robustart: Benchmarking robustness on architecture design and training techniques. *arXiv preprint arXiv:2109.05211*.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2019. Robustness May Be at Odds with Accuracy. In *International Conference on Learning Representations Workshop*.

Wang, Y.; Zou, D.; Yi, J.; Bailey, J.; Ma, X.; and Gu, Q. 2020. Improving Adversarial Robustness Requires Revisiting Misclassified Examples. In *International Conference on Learning Representations Workshop*.

Wu, D.; Xia, S.-T.; and Wang, Y. 2020. Adversarial Weight Perturbation Helps Robust Generalization. In *Advances in Neural Information Processing Systems*.

Xie, C.; Wu, Y.; van der Maaten, L.; Yuille, A. L.; and He, K. 2019. Feature Denoising for Improving Adversarial Robustness. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Xu, H.; Liu, X.; Li, Y.; Jain, A. K.; and Tang, J. 2021. To be Robust or to be Fair: Towards Fairness in Adversarial Training. In *International Conference on Machine Learning*.

Zafar, M. B.; Valera, I.; Gomez Rodriguez, M.; and Gummadi, K. P. 2017. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *International Conference on World Wide Web*.

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

Zhang, C.; Liu, A.; Liu, X.; Xu, Y.; Yu, H.; Ma, Y.; and Li, T. 2021a. Interpreting and Improving Adversarial Robustness of Deep Neural Networks with Neuron Sensitivity. *IEEE Transactions on Image Processing*.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; Ghaoui, L. E.; and Jordan, M. 2019. Theoretically Principled Trade-off between Robustness and Accuracy. In *International Conference on Machine Learning*.

Zhang, J.; Xu, X.; Han, B.; Niu, G.; Cui, L.; Sugiyama, M.; and Kankanhalli, M. S. 2020. Attacks Which Do Not Kill Training Make Adversarial Learning Stronger. In *International Conference on Machine Learning*.

Zhang, J.; Zhu, J.; Niu, G.; Han, B.; Sugiyama, M.; and Kankanhalli, M. S. 2021b. Geometry-aware Instance-reweighted Adversarial Training. In *International Conference on Learning Representations Workshop*.