CFA: Class-wise Calibrated Fair Adversarial Training

Zeming Wei[1], Yifei Wang[1], Yiwen Guo[2], Yisen Wang[3,4*]
[1]School of Mathematical Sciences, Peking University [2]Independent Researcher
[3]National Key Lab of General Artificial Intelligence
School of Intelligence Science and Technology, Peking University
[4]Institute for Artificial Intelligence, Peking University

Abstract

*Adversarial training has been widely acknowledged as the most effective method to improve the adversarial robustness against adversarial examples for Deep Neural Networks (DNNs). So far, most existing works focus on enhancing the overall model robustness, treating each class equally in both the training and testing phases. Although revealing the disparity in robustness among classes, few works try to make adversarial training fair at the class level without sacrificing overall robustness. In this paper, we are the first to theoretically and empirically investigate the preference of different classes for adversarial configurations, including perturbation margin, regularization, and weight averaging. Motivated by this, we further propose a **C**lass-wise calibrated **F**air **A**dversarial training framework, named CFA, which customizes specific training configurations for each class automatically. Experiments on benchmark datasets demonstrate that our proposed CFA can improve both overall robustness and fairness notably over other state-of-the-art methods. Code is available at* https://github.com/PKU-ML/CFA.

1. Introduction

Deep Neural Networks (DNNs) have achieved remarkable success in a variety of tasks, but their vulnerability against adversarial examples [11, 20] have caused serious concerns about their application in safety-critical scenarios [6, 15]. DNNs can be easily fooled by adding small, even imperceptible perturbations to the natural examples. To address this issue, numerous defense approaches have been proposed [2, 9, 17, 18, 28], among which Adversarial Training (AT) [16, 25] has been demonstrated as the most effective method to improve the model robustness against such attacks [1, 27]. Adversarial training can be formulated as the following min-max optimization problem:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \max_{\|x'-x\|\le\epsilon} \mathcal{L}(\boldsymbol{\theta}; x', y), \tag{1}$$

where $\mathcal{D}$ is the data distribution, ϵ is the margin of perturbation and $\mathcal{L}$ is the loss function, *e.g.* the cross-entropy loss. Generally, Projected Gradient Descent (PGD) attack [16] has shown satisfactory effectiveness to find adversarial examples in the perturbation bound $\mathcal{B}(x,\epsilon) = \{x' : \|x'-x\| \le \epsilon\}$, which is commonly used in solving the inner maximization problem in (1):

$$x^{t+1} = \Pi_{\mathcal{B}(x,\epsilon)}(x^t + \alpha \cdot \mathrm{sign}(\nabla_{x^t}\mathcal{L}(\boldsymbol{\theta}; x^t, y))), \tag{2}$$

where Π is the projection function and α controls the step size of gradient ascent. TRADES [30] is another variant of AT, which adds a regularization term to adjust the trade-off between robustness and accuracy [22, 24]:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \{\mathcal{L}(\boldsymbol{\theta}; x, y) + \beta \max_{\|x'-x\|\le\epsilon} \mathcal{K}(f_{\boldsymbol{\theta}}(x), f_{\boldsymbol{\theta}}(x'))\}, \tag{3}$$

where $\mathcal{K}(\cdot)$ is the KL divergence and β is the *robustness regularization* to adjust the robustness-accuracy trade-off.

Although certain robustness has been achieved by AT and its variants, there still exists a stark difference among class-wise robustness in adversarially trained models, *i.e.*, the model may exhibit strong robustness on some classes while it can be highly vulnerable on others, as firstly revealed in [4, 21, 29]. This disparity raises the issue of robustness fairness, which can lead to further safety concerns of DNNs, as the models that exhibit good overall robustness may be easily fooled on some specific classes, *e.g.*, the stop sign in automatic driving. To address this issue, Fair Robust Learning (FRL) [29] has been proposed, which adjusts the margin and weight among classes when fairness constraints are violated. However, this approach only brings limited improvement on robust fairness while causing a drop on overall robustness.

In this paper, we first present some theoretical insights on how different adversarial configurations impact class-wise robustness, and reveal that strong attacks can be detrimental to the *hard* classes (classes that have lower clean accuracy). This finding is further empirically confirmed through evaluations of models trained under various adversarial configurations. Additionally, we observe that the worst robust-

*Corresponding Author: Yisen Wang (yisen.wang@pku.edu.cn)

ness among classes fluctuates significantly between different epochs during the training process. It indicates that simply selecting the checkpoint with the best overall robustness like the previous method [19] may result in poor robust fairness, *i.e.*, the worst class robustness may be extremely low.

Inspired by these observations, we propose to dynamically customize different training configurations for each class. Note that unlike existing instance-wise customized methods that aim to enhance overall robustness [3,7,10,26,31], we also focus on the fairness of class-wise robustness. Furthermore, we modify the weight averaging technique to address the fluctuation issue during the training process. Overall, we name the proposed framework as **C**lass-wise calibrated **F**air **A**dversarial training (CFA).

Our contributions can be summarized as follows:

- We show both theoretically and empirically that different classes require appropriate training configurations. In addition, we reveal the fluctuating effect of the worst class robustness during adversarial training, which indicates that selecting the model with the best overall robustness may result in poor robust fairness.
- We propose a novel approach called Class-wise calibrated Fair Adversarial training (CFA), which dynamically customizes adversarial configurations for different classes during the training phase, and modifies the weight averaging technique to improve and stabilize the worst class robustness.
- Experiments on benchmark datasets demonstrate that our CFA outperforms state-of-the-art methods in terms of both overall robustness and fairness, and can be also easily incorporated into other adversarial training approaches to further improve their performance.

2. Theoretical Class-wise Robustness Analysis

In this section, we present our theoretical insights on the influence of different adversarial configurations on class-wise robustness.

2.1. Notations

For a K-classification task, we use $f : \mathcal{X} \to \mathcal{Y}$ to denote the classification function which maps from the input space $\mathcal{X}$ to the output labels $\mathcal{Y} = \{1, 2, \cdots, K\}$. For an example $x \in \mathcal{X}$, we use $\mathcal{B}(x, \epsilon) = \{x' | \|x' - x\| \leq \epsilon\}$ to restrict the perturbation. In this paper, we mainly focus on the l_∞ norm $\|\cdot\|_\infty$, and note that our analysis and approach can be generalized to other norms similarly.

We use $\mathcal{A}(f)$ and $\mathcal{R}(f)$ to denote the clean and robust accuracy of the trained model f:

$$\begin{aligned} \mathcal{A}(f) &= \mathbb{E}_{(x,y)\sim\mathcal{D}}\, \mathbf{1}(f(x) = y), \\ \mathcal{R}(f) &= \mathbb{E}_{(x,y)\sim\mathcal{D}}\, \mathbf{1}(\forall x' \in \mathcal{B}(x, \epsilon), f(x') = y). \end{aligned} \tag{4}$$

We use $\mathcal{A}_k(f)$ and $\mathcal{R}_k(f)$ to denote the clean and robust accuracy of the k-th class respectively to analyze the class-wise robustness.

2.2. A Binary Classification Task

We consider a simple binary classification task that is similar to the data model used in [22], but the properties (hard or easy) of the two classes are different.

Data Distribution. Consider a binary classification task where the data distribution $\mathcal{D}$ consists of input-label pairs $(x, y) \in \mathbb{R}^{d+1} \times \{-1, +1\}$. The label y is uniformly sampled, *i.e.*, $y \overset{\text{u.a.r.}}{\sim} \{-1, +1\}$. For input $x = (x_1, x_2, \cdots, x_{d+1})$, let $x_1 \in \{-1, +1\}$ be the *robust feature*, and $x_2, \cdots, x_{d+1}$ be the *non-robust features*. The robust feature x_1 is labeled as $x_1 = y$ with probability p and $x_1 = -y$ with probability $1 - p$ where $0.5 \leq p < 1$. For the non-robust features, they are sampled from $x_2, \cdots, x_{d+1} \overset{\text{i.i.d}}{\sim} \mathcal{N}(\eta y, 1)$ where $\eta < 1/2$ is a small positive number. Intuitively, as discussed in [22], x_1 is robust to perturbation but not perfect (as $p < 1$), and $x_2, \cdots, x_{d+1}$ are useful for classification but sensitive to small perturbation. In our model, we introduce some differences between the two classes by letting the probability of $x_1 = y$ correlate with its label y. Overall, the data distribution is

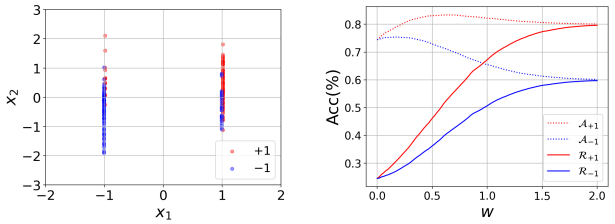
$$x_1 = \begin{cases} +y, & \text{w.p. } p_y \\ -y, & \text{w.p. } 1 - p_y \end{cases} \text{ and } \quad x_2, \cdots, x_{d+1} \overset{\text{i.i.d}}{\sim} \mathcal{N}(\eta y, 1). \tag{5}$$

We set $p_{+1} > p_{-1}$ in our model. Therefore, the robust feature x_1 is more reliable for class $y = +1$, while for class $y = -1$, the robust feature x_1 is noisier and their classification depends more on the non-robust features $x_2, \cdots, x_{d+1}$.

Hypothesis Space. Consider a SVM classifier (without bias term) $f(x) = \text{sign}(w_1 x_1 + w_2 x_2 + \cdots + w_{d+1} x_{d+1})$. For the sake of simplicity, we assume $w_1, w_2 \neq 0$, and $w_2 = w_3 = \cdots = w_{d+1}$ since $x_2, \cdots, x_{d+1}$ are equivalent. Then, let $w = \frac{w_1}{w_2}$, the model can be simplified as $f_w(x) = \text{sign}(x_1 + \frac{x_2 + \cdots + x_{d+1}}{w})$. Without loss of generality, we further assume $w > 0$ since $x_2, \cdots, x_{d+1} \sim \mathcal{N}(\eta y, 1)$ tend to share the same sign symbol with y.

2.3. Theoretical Insights

Illustration Example. An example of the data distribution for the case $d = 1$ is visualized in Fig. 1(a). The data points for class $y = +1$ are colored red and for $y = -1$ are colored blue. We can see that the robust feature x_1 of class $y = -1$ seems to be noisier than $y = +1$, since the frequency of blue dots appearing on the line $x_1 = 1$ is higher compared to the frequency of red dots appearing on the line $x_1 = -1$, with $p_{+1} > p_{-1}$. Therefore, class $y = -1$ might be more difficult to learn. Furthermore, we plot the clean and robust accuracy of the two classes of f_w for different w in Fig. 1(b). Implementation details of this example can be found in Appendix A. The parameter



(a) (b)

Figure 1. An visualization of the toy model for the case $d = 1$. (a): Sampled data from the distribution. Red dots are labeled $y = +1$ and blue dots are labeled $y = -1$. (b): Clean and robust accuracy of the two classes. Solid lines indicate robust accuracy and dotted lines indicate clean accuracy.

w can be regarded as the strength of adversarial attack in adversarial training, since larger w indicates the classifier f_w bias less weight on non-robust features $w_2, \cdots, w_{d+1}$ and pay more attention on robust feature w_1. We can see that as w increases, the clean accuracy of $y = -1$ drops significantly faster than $y = +1$, but the robustness improves slower. We formally prove this observation in the following.

The Intrinsically Hard Class. First we formally distinct class $y = -1, +1$ as the *hard* and *easy* class in Theorem 1.

Theorem 1 *For any $w > 0$ and the classifier $f_w = sign(x_1 + \frac{x_2+\cdots+x_{d+1}}{w})$, we have $\mathcal{A}_{+1}(f_w) > \mathcal{A}_{-1}(f_w)$ and $\mathcal{R}_{+1}(f_w) > R_{-1}(f_w)$.*

Theorem 1 shows that the class $y = -1$ is more difficult to learn than class $y = +1$ both in robust and clean settings. This reveals the potential reason why some classes are intrinsically difficult to learn in the adversarial setting, that is, their robust features are less reliable.

Relation Between w and Attack Strength. Consider the model is adversarially trained with perturbation margin ϵ. The following Theorem 2 shows using larger ϵ enlarges w.

Theorem 2 *For any $0 \le \epsilon \le \eta$, let $w^* = \arg\max_w \mathcal{R}(f_w)$ be the optimal parameter for adversarial training with perturbation bound ϵ, then w^* is monotone increasing at ϵ.*

Theorem 2 bridges the gap between model parameter and attack strength in adversarial training. Next, we can implicitly investigate the influence of attack strength on class-wise robustness by analyzing the parameter w.

Impact of Attack Strength on Class-wise Robustness. Here, we demonstrate how adversarial strength influences class-wise clean and robust accuracy.

Theorem 3 *Let $w_y^* = \arg\max_w \mathcal{A}_y(f_w)$ be the parameter for the best clean accuracy of class y, then $w_{+1}^* > w_{-1}^*$.*

Theorem 3 shows that the clean accuracy of the hard class $y = -1$ reaches its best performance *earlier* than $y = +1$. In other words, $\mathcal{A}_{-1}(f_w)$ starts dropping earlier than $\mathcal{A}_{+1}(f_w)$. As the model further distracts its attention from its clean accuracy to robustness by increasing the parameter w, the hard class $y = -1$ losses more clean accuracy yet gains less robust accuracy as shown in Theorem 4.

Theorem 4 *Suppose $\Delta_w > 0$, then for $\forall w > w_{+1}^*$, $\mathcal{A}_{-1}(f_{w+\Delta_w}) - \mathcal{A}_{-1}(f_w) < \mathcal{A}_{+1}(f_{w+\Delta_w}) - \mathcal{A}_{+1}(f_w) < 0$, and for $\forall w > 0$, $0 < \mathcal{R}_{-1}(f_{w+\Delta_w}) - \mathcal{R}_{-1}(f_w) < \mathcal{R}_{+1}(f_{w+\Delta_w}) - \mathcal{R}_{+1}(f_w)$.*

The proof of the theorems can be found in Appendix B. In this section, we demonstrate the unreliability of robust features is a possible explanation for the intrinsic difficulty in learning some classes. Then, by implicitly expressing the attack strength with parameter w, we analyze how adversarial configuration influence class-wise robustness. Theorems 3 and 4 highlight the negative impact of strong attack on the hard class $y = -1$.

3. Observations on Class-wise Robustness

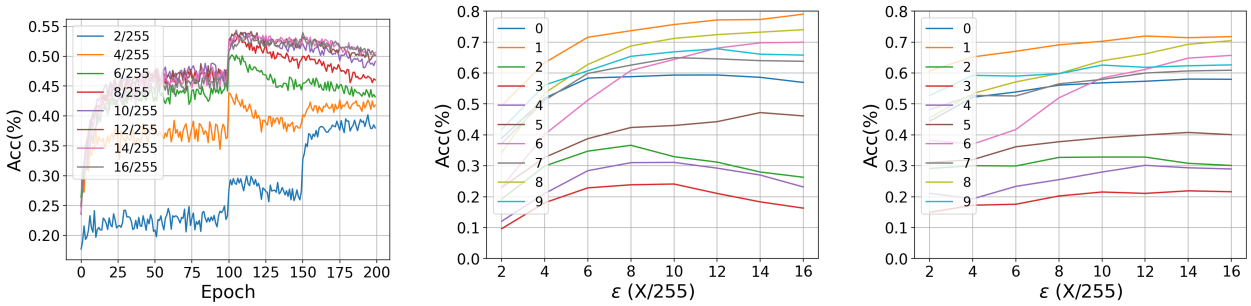
In this section, we present our empirical observations on the class-wise robustness of models adversarially trained under different configurations. Taking vanilla AT [16] and TRADES [30] as examples, we compare two key factors in the training configurations: the perturbation margin ϵ in vanilla AT and the regularization β in TRADES. We also reveal the fluctuation effect of the worst class robustness during the training process, which has a significant impact on the robust fairness in adversarial training.

3.1. Different Margins

Following the vanilla AT [16], we train 8 models on the CIFAR10 dataset [14] with l_∞-norm perturbation margin ϵ from $2/255$ to $16/255$ and analyze their overall and class-wise robustness.

The comparison of overall robustness is shown in Fig. 2(a). The robustness is evaluated under PGD-10 attack bounded by $\epsilon_0 = 8/255$, which is commonly used for robustness evaluation. Intuitively, using a larger margin can lead to better robustness. For $\epsilon < \epsilon_0$, the attack is too weak and hence the robust accuracy of the trained model is not comparable with $\epsilon \ge \epsilon_0$. However, for the three models trained with $\epsilon > \epsilon_0$, although their robustness outperforms the case of $\epsilon = \epsilon_0$ at the last epoch, they do not make significant progress on the best-case robustness (around 100-th epoch).

We take a closer look at this phenomenon by investigating their class-wise robustness in Fig. 2(b) and Fig. 2(c). For each class, we calculate the average class-wise robust accuracy among the $101-120$-th epochs (where the model performs the best robustness) and $181-200$-th epochs, respectively. From Fig. 2(b), we can see that a larger training



(a) (b) (c)

Figure 2. Comparison of overall and class-wise robustness of models adversarially trained on CIFAR10 with different perturbation margin ϵ. (a): Overall robust accuracy with different perturbation margin ϵ from 2/255 to 16/255. (b): Average class-wise robust accuracy at epoch $101-120$ (each line represents a class). (c): Average class-wise robust accuracy at epoch $181-200$ (each line represents a class).

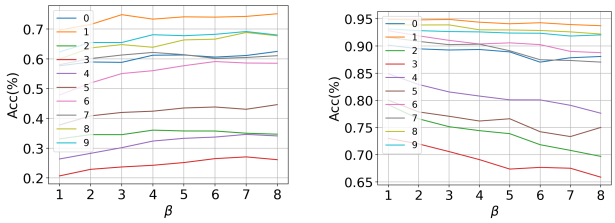
margin ϵ does not necessarily result in better class-wise robustness. For the *easy* classes which perform higher robustness, their robustness monotonously increase as ϵ enlarges from $2/255$ to $16/255$. By contrast, for the *hard* classes (especially class $2, 3, 4$), their robustness drop when ϵ enlarges from $8/255$. However, for the last several checkpoints in Fig. 2(c), we can see a consistent increase on class-wise robustness when the ϵ enlarges. Revisiting the overall robustness, we can summarize that the class-wise robustness is boosted mainly by reducing the robust over-fitting problem in the last checkpoint. This can explain why Fair Robust Learning (FRL) [29] can improve robust fairness by enlarging the margin for the hard classes, since the model reduces the over-fitting problem on these classes. Considering the overall robustness is lower in the last checkpoint (robust fairness is better though), we hope to improve the best-case robust fairness in the situation of a relatively high overall robustness.

In summary, larger perturbation is harmful to the hard classes in the best case, while can marginally improve the class-wise robustness in the later stage of training. For easy classes, larger perturbation is useful at whatever the best and last checkpoints. Therefore, a specific and proper perturbation margin is needed for each class.

3.2. Different Regularizations

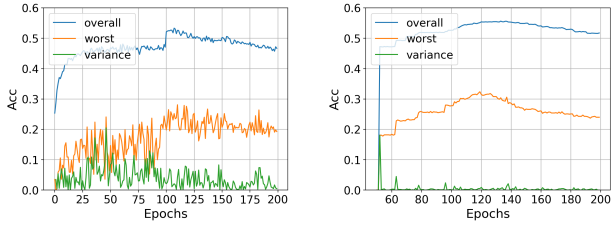
In this section, we also conduct a similar experiment on the selection of *robustness regularization* β in TRADES. We compare models trained on CIFAR10 with β from 1 to 8, and plot the average class-wise robust and clean accuracy among the $151-170$-th epochs (where TRADES performs the best performance) in Fig. 3. We can see that bias more weight on robustness (use larger β) cause different influences among classes. Specifically, for *easy classes*, improving β can improve their robustness at the cost of little clean accuracy reduction, while for *hard classes* (*e.g.*, classes $2, 3, 4$), improving β can only obtain limited robustness improvement but drop clean accuracy significantly.

This result is consistent with the Theorem 4. Recall that



(a) (b)

Figure 3. Comparison of class-wise robustness trained by TRADES with different robustness regularization parameters β. (a) Class-wise robust accuracy. (b) Class-wise clean accuracy.



(a) (b)

Figure 4. Comparison of overall robustness, the worst class robustness, and the absolute variation of the worst class robustness between adjacent checkpoints. (a): Vanilla AT. (b): AT with fairness aware weight averaging (FAWA), start from epoch 50.

in the toy model, hard class $y = -1$ costs more clean accuracy to exchanges for little robustness improvement than easy class $y = +1$. Therefore, similar to the analysis on perturbation margin ϵ, we also point out that there exists a proper β_y for each class.

3.3. Fluctuation Effect

In this section, we reveal an intriguing property regarding the fluctuation of class-wise robustness during adversarial training. In Fig. 4(a), we plot the overall robustness, the worst class robustness, and the variance of the worst robustness between adjacent epochs in vanilla adversarial training. While the overall robustness tends to be more sta-

ble between adjacent checkpoints (except when the learning rate decays), the worst class robustness fluctuates significantly. Particularly, many adjacent checkpoints between the $101-120$-th epochs exhibit a nearly 10% difference in the worst class robustness, while changes in overall robustness are negligible (less than 1%). Therefore, previously widely used selecting the best checkpoint based on overall robustness may result in an extremely unfair model. Taking the plotted training process as an example, the model achieves the highest robust accuracy of 53.2% at the 108-th epoch, which only has 23.5% robust accuracy on the worst class. In contrast, the checkpoint at epoch 110, which has 52.6% overall and 28.1% worst class robust accuracy, is preferred when considering fairness.

4. Class-wise Calibrated Fair Adversarial Training

With the above analysis, we introduce our proposed **C**lass-wise calibrated **F**air **A**dversarial training (CFA) framework in this section. Overall, the CFA framework consists of three main components: Customized Class-wise perturbation Margin (CCM), Customized Class-wise Regularization (CCR), and Fairness Aware Weight Averaging (FAWA). The CCM and CCR customize appropriate training configurations for different classes, and FAWA modifies weight averaging to improve and stabilize fairness.

4.1. Class-wise Calibrated Margin (CCM)

In Sec. 3.1, we have demonstrated that different classes prefer specific perturbation margin ϵ in adversarial training. However, it is impractical to directly find the optimal class-wise margin. Inspired by a series of instance-wise adaptive adversarial training approaches [3, 10, 26], which customize train setting for each instance according to the model performance on current example, we propose to leverage the class-wise training accuracy as the measurement of difficulty.

Suppose the k-th class achieved train robust accuracy $t_k \in [0, 1]$ in the last training epoch. In the next epoch, we aim to update the margin ϵ_k for class k based on t_k. Based on our analysis in Sec. 3.1, we consider using a relatively smaller margin for the hard classes which are more vulnerable to attacks, and identify the *difficulty* among classes by the train robust accuracy tracked from the previous epoch. To avoid ϵ_k too small, we add a hyper-parameter λ_1 (called *base perturbation budget*) on all t_k and set the calibrated margin ϵ_k by multiply the coefficient on primal margin ϵ:

$$\epsilon_k \leftarrow (\lambda_1 + t_k) \cdot \epsilon, \tag{6}$$

where ϵ is the original perturbation margin, *e.g.*, $8/255$ that is commonly used for CIFAR-10 dataset. Note that the calibrated margin ϵ_k can adaptively converge to find the proper range during the training phase, for example, if the margin is too small for class k, the model will perform high train robust accuracy t_k and then increase ϵ_k by schedule (6).

4.2. Class-wise Calibrated Regularization (CCR)

We further customize different robustness regularization β of TRADES for different classes. Recall the objective function (3) of TRADES, we hope the hard classes tend to bias more weight on its clean accuracy. Still, we measure the difficulty by the train robust accuracy t_k for class k, and propose the following calibrated robustness regularization β_k:

$$\beta_k \leftarrow (\lambda_2 + t_k) \cdot \beta. \tag{7}$$

where β is the originally selected parameter. The objective function (3) can be rewritten as:

$$\mathcal{L}_{\boldsymbol{\theta}}(\beta; x, y) = \frac{\mathcal{L}(\boldsymbol{\theta}; x, y) + \beta_y \max\limits_{\|x'-x\| \le \epsilon} \mathcal{K}(f_{\boldsymbol{\theta}}(x), f_{\boldsymbol{\theta}}(x'))}{1 + \beta_y}. \tag{8}$$

To balance the weight between different classes, we add a denominator $1 + \beta_y$ since β_y is distinct among classes. Therefore, for the hard classes which have lower β_y tend to bias higher weight $\frac{1}{1+\beta_y}$ on its natural loss $\mathcal{L}(\boldsymbol{\theta}; x, y)$. Note that simply replacing ϵ in (8) with ϵ_k can combine the calibrated margin with this calibrated regularization. On the other hand, for general adversarial training algorithms, our calibrated margin schedule (6) can also be combined.

4.3. Fairness Aware Weight Average (FAWA)

As plotted in Fig. 4(a), the worst class robustness changes largely, among which part of checkpoints performs extremely poor robust fairness. Previously, there are a series of weight averaging methods to make the model training stable, *e.g.*, exponential moving average (EMA) [13, 23], thus we hope to further improve the worst class robustness by fixing the weight average algorithm.

Inspired by the large fluctuation of the robustness fairness among checkpoints, we consider eliminating the *unfair* checkpoints out in the weight averaging process. To this end, we propose a *Fairness Aware Weight Average (FAWA)* approach, which sets a threshold δ on the worst class robustness of the new checkpoint in the EMA process. Specifically, we extract a validation set from the dataset, and each checkpoint is adopted in the weight average process if and only if its worst class robustness is higher than δ. Fig. 4(b) shows the effect of the proposed FAWA. The difference between adjacent epochs is extremely small (less than 1%), and the overall robustness also outperforms vanilla AT.

4.4. Discussion

Overall, by combining the above components, we accomplish our CFA framework. An illustration of incorporating CFA to TRADES is shown in Alg. 1. Note that for

Algorithm 1: TRADES with CFA

Input: A DNN classifier $f_{\boldsymbol{\theta}}(\cdot)$ with parameter $\boldsymbol{\theta}$; Train dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$; Batch size m; Initial perturbation margin ϵ and robustness regularization β; Train epochs N; Batch size m; Learning rate η; Weight average decay rate α; Fairness threshold δ

Output: A fair and robust DNN classifier $\bar{f}_{\bar{\boldsymbol{\theta}}}(\cdot)$

```
/* Initialize parameters and
   datasets                                   */
```

Initialize $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0, \bar{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}$;

Split $D = D_{\text{train}} \cup D_{\text{valid}}$;

for $y \in \mathcal{Y}$ **do**

```
    /* Initialize ϵ_y and β_y                 */
```

    $\epsilon_y \leftarrow \epsilon, \beta_y \leftarrow \beta$;

for $T \leftarrow 1, 2, \cdots N$ **do**

  for *Every minibatch* (x, y) *in* D_{train} **do**

```
      /* Use ϵ_y and β_y to train             */
```

    $x' \leftarrow \arg \max_{x' \in \mathcal{B}(x, \epsilon_y)} \mathcal{K}(f_{\boldsymbol{\theta}}(x), f_{\boldsymbol{\theta}}(x'))$;

    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\theta}}(\beta_y; x, y)$;

  for $y \in \mathcal{Y}$ **do**

    $t_y \leftarrow Train\_Acc(f_\theta, T)$;

```
      /* Update ϵ_y, β_y with t_y             */
```

    $\epsilon_y \leftarrow (\lambda_1 + t_k) \cdot \epsilon$;

    $\beta_y \leftarrow (\lambda_2 + t_k) \cdot \epsilon$;

```
  /* Fairness Aware Weight Average
     */
```

  if $\min_{y \in \mathcal{Y}} \mathcal{R}_y(f_{\boldsymbol{\theta}}, D_{valid}) \geq \delta$ **then**

    $\bar{\boldsymbol{\theta}} \leftarrow \alpha \bar{\boldsymbol{\theta}} + (1 - \alpha)\boldsymbol{\theta}$;

return $\bar{f}_{\bar{\boldsymbol{\theta}}}$;

other methods like AT, we can still incorporate CFA by removing the CCR schedule specified for TRADES. Moreover, we discuss the difference between our proposed CFA and other works.

Comparison with Fair Robust Learning (FRL) [29]. Here we highlight the differences between our CFA framework and Fair Robust Learning (FRL), the only existing adversarial training algorithm designed to improve the fairness of class-wise robustness. The FRL framework consists of two components: remargin and reweight. Initially, a robust model is trained, and a fairness constraint on the difference of robustness among classes is set. When the constraint is violated, the model is fine-tuned persistently by increasing the perturbation bound ϵ_k and weighting the loss of the hard classes. Although CFA also includes adaptive margin and regularization weight schedules, our work is fundamentally distinct from FRL. Firstly, as discussed in Sec. 3.1, a larger margin only mitigates the robust over-fitting problem but does not provide higher peak performance. In contrast, our approach aims to customize the proper margin for each class, which boosts the best performance. Secondly, FRL improves robust fairness at the cost of reducing overall robustness, which could be seen as *unfair* to other classes. However, our CFA framework improves both overall and worst class performance. In addition, FRL requires an initial robust model before fairness fine-tuning, resulting in extra computational burden. Finally, the fluctuation effect discussed in Sec. 3.3 is not considered in FRL.

Comparison with Instance-wise Adversarial Training. Though there exists a series of instance-wise adaptive adversarial training [3, 5, 7, 10, 25, 26, 31, 32] toward better robust generalization, to the best of our knowledge, we are the first work to pursue this from a class-wise perspective. Here we demonstrate several differences between our class-wise and other instance-wise adversarial training algorithms. First of all, CFA focuses on improve both overall and the worst class robust accuracy, while all existing instance-wise approaches only focus on overall robustness. Unfortunately, as shown in Sec. 5, the instance-wise ones are not comparable with our CFA from the perspective of fairness. In addition, instance-wise methods can be seen as to find the solution for each individual sample, while class-wise ones are to find the solution for multiple samples. Thus, class-wise methods can alleviate the frequent fluctuation while remaining the specificity (a class of samples) of configurations among training samples. Therefore, our class-wise calibration achieves a better trade-off between flexibility and stability. Finally, some instance-wise approaches can be well-combined with our CFA framework to further boost their performance.

5. Experiment

In this section, we demonstrate the effectiveness of our proposed CFA framework to improve both overall and class-wise robustness.

5.1. Experimental Setup

We conduct our experiments on the benchmark dataset CIFAR-10 [14] using PreActResNet-18 (PRN-18) [12] model. Experiments on Tiny-ImageNet can be found in Appendix C.1.

Baselines. We select vanilla adversarial training (AT) [16] and TRADES [30] as our baselines. Additionally, since our Fairness Aware Weight Average (FAWA) method is a variant of the weight average method with *Exponential Moving Average (EMA)*, we include baselines with EMA as well. For instance-wise adaptive adversarial training approaches, we include FAT [31], which adaptively adjusts attack strength on each instance. Finally, we compare our approach with FRL [29], the only existing adversarial training algorithm that focuses on improving the fairness of class-wise robustness.

Training Settings. Following the best settings in [19], we

Table 1. Overall comparison of our proposed CFA framework with original methods.

Method	Best (Avg. / Worst) Clean Accuracy	Best (Avg. / Worst) AA. Accuracy	Last (Avg. / Worst) Clean Accuracy	Last (Avg. / Worst) AA. Accuracy
AT	**82.3** ±0.8 / 63.9 ±1.6	46.7 ±0.5 / 20.1 ±1.3	84.1 ±0.2 / 65.1±2.4	43.0 ±0.4 / 15.5 ±1.8
AT + EMA	81.9 ±0.3 / 61.6 ±0.5	49.6 ±0.2 / 21.3 ±0.8	**84.8** ±0.1 / 67.7 ±0.7	44.3 ±0.5 / 18.1 ±0.5
AT + CFA	80.8 ±0.3 / **64.6** ±0.4	**50.1** ±0.3 / **24.4** ±0.3	83.6 ±0.2 / **68.7** ±0.7	**47.7** ±0.4 / **20.5** ±0.4
TRADES	**82.3** ±0.1 / **67.8** ±0.6	48.3 ±0.3 / 21.7 ±0.5	83.9 ±0.3 / 66.9 ±1.5	46.9 ±0.3 / 18.5 ±1.3
TRADES + EMA	81.2 ±0.4 / 65.0 ±0.7	49.7 ±0.3 / 24.2 ±0.6	**84.5** ±0.1 / 67.9 ±0.1	48.3 ±0.2 / 20.7 ±0.3
TRADES + CFA	80.4 ±0.2 / 66.2 ±0.5	**50.1** ±0.2 / **26.5** ±0.4	83.0 ±0.1 / **68.1** ±0.3	**49.3** ±0.1 / **21.5** ±0.3
FAT	84.6 ±0.4 / **69.2** ±0.8	45.7 ±0.6 / 17.2 ±1.3	85.4 ±0.2 / 70.8 ±1.9	42.1 ±0.1 / 14.8 ±1.6
FAT + EMA	**85.2** ±0.2 / 66.7 ±0.6	48.6 ±0.1 / 18.3 ±0.5	**85.7** ±0.2 / **71.2** ±0.4	43.2 ±0.1 / 15.7 ±0.7
FAT + CFA	82.1 ±0.3 / 64.7 ±0.9	**49.6** ±0.1 / **20.9** ±0.8	84.3 ±0.1 / 69.4 ±0.3	**45.1** ±0.2 / **16.7** ±0.2
FRL	82.8 ±0.1 / **71.4** ±2.4	45.9 ±0.3 / 25.4 ±2.0	**82.8** ±0.2 / 72.9 ±1.5	44.7 ±0.2 / 23.1 ±0.8
FRL + EMA	**83.6** ±0.3 / 69.5 ±0.7	**46.1** ±0.2 / **25.6** ±0.4	81.9 ±0.2 / **74.2** ±0.3	**44.9** ±0.2 / **24.5** ±0.3

train a PRN-18 using SGD with momentum 0.9, weight decay 5×10^{-4}, and initial learning rate 0.1 for 200 epochs. The learning rate is divided by 10 after epoch 100 and 150. All experiments are conducted by default perturbation margin $\epsilon = 8/255$, and for TRADES, we initialize $\beta = 6$. For the base attack strength for Class-wise Calibrated Margin (CCM), we set $\lambda_1 = 0.5$ for AT and $\lambda_1 = 0.3$ for TRADES since the training robust accuracy of TRADES is higher than AT. For FAT, we set $\lambda_1 = 0.7$ to avoid the attack being too weak to hard classes. Besides, we set $\lambda_2 = 0.5$ for Class-wise Calibrated Regularization (CCR) in TRADES. For the weight average methods, the decay rate of FAWA and EMA is set to 0.85, and the weight average processes begin at the 50-th epoch for better initialization. We draw 2% samples from each class as the validation set for FAWA, and train on the rest of 98% samples, hence FAWA does not lead to extra computational costs. The fairness threshold for FAWA is set to 0.2.

Metrics. We evaluate the clean and robust accuracy both in average and the worst case among classes. The robustness is evaluated by **AutoAttack (AA)** [8], a well-known reliable attack for robustness evaluation. To perform the best performance during the training phase, we adopt early stopping in adversarial training [19] and present both the best and last results among training checkpoints. Further, as discussed in Sec. 3.3 that the worst class robust accuracy changes drastically, we select the checkpoint that achieves the highest sum of overall and the worst class robustness to report the results for a fair comparison.

5.2. Robustness and Fairness Performance

We implement our proposed training configuration schedule on AT, TRADES, and FAT. To evaluate the effectiveness of our approach, we conduct five independent experiments for each method and report the mean result and standard deviation.

As summarized in Table 1, CFA helps each method achieve a significant robustness improvement both in average and the worst class at the best and last checkpoints. Furthermore, when compared with baselines that use weight average (EMA), our CFA still achieves higher overall and the worst class robustness for each method, especially in the worst class at the best checkpoints, where the improvement exceeds 2%. Note that the vanilla FAT only achieves 17.2% the worst class robustness at the best checkpoint which is even lower than TRADES, which verifies the discussion in Sec. 4.4 that instance-wise adaptive approaches are not helpful for robustness fairness. We also visualize and compare the robustness for each class in Appendix C.2, which shows that CFA indeed reduces the difference among class-wise robustness and improves the fairness without harming other classes.

We also compare our approach with FRL [29]. However, since FRL also applies a remargin schedule, we cannot incorporate our CFA into FRL. Therefore, we only report results of FRL with and without EMA in Table 1. As FRL is a variant of TRADES that applies the loss function of TRADES, we compare the results of FRL with TRADES and TRADES+CFA. From Table 1, we observe that FRL and FRL+EMA show only marginal progress (less than 2%) in the worst class robustness as compared to TRADES+EMA, but at a expensive cost (about 3%) of reducing the average performance. As demonstrated in Sec. 3.1, larger margin which is adopted in FRL mainly mitigates the robust over-fitting issue but does not bring satisfactory best performance. This is further confirmed by the performance of final checkpoints of FRL, where FRL exhibits better performance in the worst class robustness. In contrast, we calibrate the appropriate margin for each class rather than simply enlarging them, thus achieving both better robustness and fairness at the best checkpoint, *i.e.*, our TRADES+CFA outperforms FRL+EMA in both average (about 4%) and the worst class (about 1%) robustness.

5.3. Ablation Study

In this section, we show the usefulness of each component of our CFA framework. Note that we still apply **AutoAttack (AA)** to evaluate robustness.

5.3.1 Effectiveness of Calibrated Configuration

First, we compare our calibrated adversarial configuration including CCM ϵ_y and CCR β_y with vanilla ones for AT, TRADES, and FAT. As Table 2 shows, both the average and worst class robust accuracy are improved for all three methods by applying CCM. Besides, CCR, which is customized for TRADES, also improves the performance of vanilla TRADES. All experiments verify that our proposed class-wise adaptive adversarial configurations are effective for robustness and fairness improvement.

We also investigate the influence of base perturbation budget λ_1 by conducting 5 experiments of AT incorporated CCM with λ_1 varies from 0.3 to 0.7. The comparison is plotted in Fig. 5(a). We can see that all models with different λ_1 show better overall and the worst class robustness than vanilla AT, among which $\lambda_1 = 0.5$ performs best. We can say that CCM has satisfactory adaptive ability on adjusting ϵ_k and is not heavily rely on the selection of λ_1. Fig. 5(b) shows the class-wise margin used in the training phase for $\lambda_1 = 0.5$. We can see the hard classes (class 2,3,4,5) use smaller ϵ_k than the original $\epsilon = 8/255$, while the easy classes use larger ones, which is consistent with our empirical observation on different margins in Sec. 3.1 and can explain why CCM is helpful to improve performance. We also present a similar comparison experiments on λ_2 for CCR in Appendix C.3.

5.3.2 FAWA Improves Worst Class Robustness

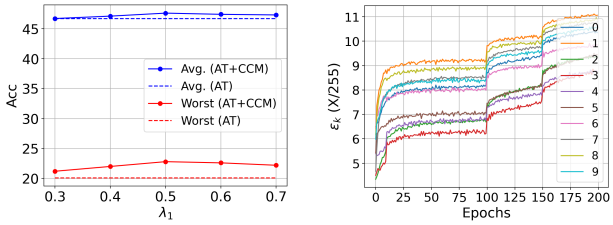
Here we present the results of our Fairness Aware Weight Averaging (FAWA) compared with the simple EMA method in Table 3. By eliminating the unfair checkpoints out, our FAWA achieves significantly better performance than EMA on the worst class robustness (nearly 2% improvement) with negligible decrease on the overall robustness (less than 0.3%). This verifies the effectiveness of FAWA on improving robustness fairness.

6. Conclusion

In this paper, we first give a theoretical analysis of how attack strength in adversarial training impacts the performance of different classes. Then, we empirically show the influence of adversarial configurations on class-wise robustness and the fluctuate effect of robustness fairness, and point out there should be some appropriate configurations for each class. Based on these insights, we propose a **C**lass-wise calibrated **F**air **A**dversarial training (CFA) framework to adaptively customize class-wise train configurations for improving robustness and fairness. Experiment shows our CFA outperforms state-of-the-art methods both in overall and fairness metrics, and can be easily incorporated into existing methods to further enhance their performance.

Table 2. Comparison of models with/without our class-wise calibrated configurations including margin ϵ and regularization β.

Method	Avg. Robust	Worst Robust
AT	46.7	20.1
+ CCM	**47.6**	**22.8**
TRADES	48.3	21.7
+ CCM	48.4	22.5
+ CCR	48.9	23.5
+ CCM + CCR	**49.2**	**23.8**
FAT	45.7	17.2
+ CCM	**46.8**	**18.9**



(a) (b)

Figure 5. Analysis on the base perturbation budget λ_1. (a): Average and the worst class robustness of models trained with different λ_1 (solid) and vanilla AT (dotted). (b): Class-wise calibrated margin ϵ_k in the training phase of $\lambda_1 = 0.5$.

Table 3. Comparison of simple EMA and our FAWA.

Method	Avg. Robust	Worst Robust
AT + EMA	**49.6**	21.3
AT + FAWA	49.3	**23.1**
TRADES + EMA	**49.7**	24.2
TRADES + FAWA	49.4	**25.1**
FAT + EMA	**48.6**	18.3
FAT + FAWA	48.5	**19.9**

Acknowledgement

Yisen Wang is partially supported by the National Key R&D Program of China (2022ZD0160304), the National Natural Science Foundation of China (62006153), and Open Research Projects of Zhejiang Lab (No. 2022RC0AB05).

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018. 1

[2] Yang Bai, Yan Feng, Yisen Wang, Tao Dai, Shu-Tao Xia, and Yong Jiang. Hilbert-based generative defense for adversarial examples. In *ICCV*, 2019. 1

[3] Yogesh Balaji, Tom Goldstein, and Judy Hoffman. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *arXiv preprint arXiv:1910.08051*, 2019. 2, 5, 6

[4] Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. Robustness may be at odds with fairness: An empirical study on class-wise accuracy. *arXiv preprint arXiv:2010.13365*, 2020. 1

[5] Qi-Zhi Cai, Min Du, Chang Liu, and Dawn Song. Curriculum adversarial training. *arXiv preprint arXiv:1805.04807*, 2018. 6

[6] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *CVPR*, 2015. 1

[7] Minhao Cheng, Qi Lei, Pin-Yu Chen, Inderjit Dhillon, and Cho-Jui Hsieh. Cat: Customized adversarial training for improved robustness. *arXiv preprint arXiv:2002.06789*, 2020. 2, 6

[8] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020. 7

[9] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E Kounavis, and Duen Horng Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900*, 2017. 1

[10] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. *arXiv preprint arXiv:1812.02637*, 2018. 2, 5, 6

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016. 6

[13] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *UAI*, 2018. 5

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3, 6

[15] Xingjun Ma, Yuhao Niu, Lin Gu, Yisen Wang, Yitian Zhao, James Bailey, and Feng Lu. Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, 2020. 1

[16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1, 3, 6

[17] Yichuan Mo, Dongxian Wu, Yifei Wang, Yiwen Guo, and Yisen Wang. When adversarial training meets vision transformers: Recipes from training to architecture. In *NeurIPS*, 2022. 1

[18] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *SP*, 2016. 1

[19] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *ICML*, 2020. 2, 6, 7

[20] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1

[21] Qi Tian, Kun Kuang, Kelu Jiang, Fei Wu, and Yisen Wang. Analysis and applications of class-wise robustness in adversarial training. In *KDD*, 2021. 1

[22] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018. 1, 2, 10

[23] Hongjun Wang and Yisen Wang. Self-ensemble adversarial training for improved robustness. In *ICLR*, 2022. 5

[24] Hongjun Wang and Yisen Wang. Generalist: Decoupling natural and robust generalization. In *CVPR*, 2023. 1

[25] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, 2019. 1, 6

[26] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2019. 2, 5, 6

[27] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, 2020. 1

[28] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019. 1

[29] Han Xu, Xiaorui Liu, Yaxin Li, Anil Jain, and Jiliang Tang. To be robust or to be fair: Towards fairness in adversarial training. In *ICML*, 2021. 1, 4, 6, 7

[30] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019. 1, 3, 6

[31] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *ICML*, 2020. 2, 6

[32] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *ICLR*, 2021. 6