Adversarial Training with Anti-adversaries

Xiaoling Zhou, Ou Wu, and Nan Yang

Abstract—Adversarial training is effective in improving the robustness of deep neural networks. However, existing studies still exhibit significant drawbacks in terms of the robustness, generalization, and fairness of models. In this study, we validate the importance of different perturbation directions (i.e., adversarial and anti-adversarial) and bounds from both theoretical and practical perspectives. The influence of adversarial training on deep learning models in terms of fairness, robustness, and generalization is theoretically investigated under a more general perturbation scope that different samples can have different perturbation directions and varied perturbation bounds. Our theoretical explorations suggest that combining adversaries and anti-adversaries with varied bounds in training can be more effective in achieving better fairness among classes and a better tradeoff among robustness, accuracy, and fairness in some typical learning scenarios compared with standard adversarial training. Inspired by our theoretical findings, a more general learning objective that combines adversaries and anti-adversaries with varied bounds on each training sample is presented. To solve this objective, two adversarial training frameworks based on meta-learning and reinforcement learning are proposed, in which the perturbation direction and bound for each sample are determined by its training characteristics. Furthermore, the role of the combination strategy with varied bounds is explained from a regularization perspective. Extensive experiments under different learning scenarios verify our theoretical findings and the effectiveness of the proposed methodology.

Index Terms—Adversarial training, anti-adversary, robustness, generalization, fairness.

1 Introduction

In addition to the standard generalization error (also known as natural error), robust generalization error (also known as robust error) has attracted great attention from researchers in recent years. A deep neural network with a low robust error can deal well with adversarial attacks. Adversarial training is an effective technique to reduce the robust errors of deep learning models [1], [2]. Given a model $f(\cdot)$ and a sample $\boldsymbol{x}$ associated with a label y, classical adversarial training methods [3], [4] first generate an adversary (i.e., adversarial example) $\boldsymbol{x}_{\text{adv}}$ for $\boldsymbol{x}$ with the following optimization:

$$\boldsymbol{x}_{\text{adv}} = \boldsymbol{x} + \arg\max_{\|\boldsymbol{\delta}\| \leq \epsilon} \ell(f(\boldsymbol{x}+\boldsymbol{\delta}), y), \tag{1}$$

where $\ell(\cdot,\cdot)$ is a loss function, $\boldsymbol{\delta}$ and ϵ are the perturbation term and bound, respectively. Adversaries are then leveraged as training data to learn a more robust model. A number of variations for adversarial training have been proposed in recent literature. Zhang et al. [5] decomposed the robust error into natural and boundary errors. They developed a new method, TRADES, to obtain a better tradeoff between standard generalization and robustness. Wang et al. [6] proposed a misclassification-aware adversarial training method to focus on the misclassified examples.

Apart from the design of new methods, theoretical studies have been conducted to explore the effectiveness and ineffectiveness of adversarial training [2]. Yang et al. [7] concluded that existing adversarial training methods could not achieve an ideal tradeoff between accuracy and robustness due to the insufficient smoothness [8] and generalization properties of classifiers trained by these methods. They pointed out that customized optimization methods or better network architectures should be proposed. The unsatisfied tradeoff of adversarial training may lead to the robust overfitting phenomenon [9]. Dong et al. [10] argued that this phenomenon is due to memorization in adversarial training. Xu et al. [11] revealed that adversarial training introduces severe unfairness among different categories. Thus, they further required the classifier to satisfy two fairness constraints, and set a varied perturbation bound for each class, resulting in better fairness. Different from these studies, we conjectured that one possible reason leading to the drawbacks of adversarial training is that not all training samples should be perturbed equally, including both perturbation directions and bounds. For instance, adversaries of noisy samples may harm the model's performance [12], and

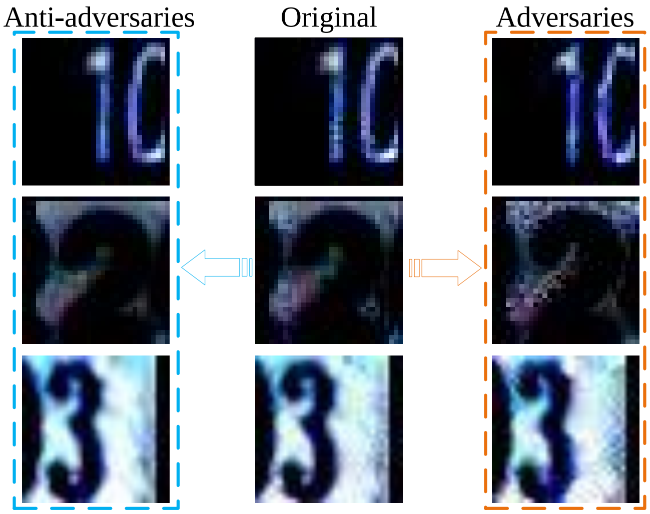


Fig. 1. Examples of adversaries and anti-adversaries for samples in the SVHN dataset. Adversarial perturbation makes samples harder, while anti-adversarial perturbation makes samples easier.

- *Xiaoling Zhou, Ou Wu, and Nan Yang are in the Center for Applied Mathematics, Tianjin University, Tianjin, China. Ou Wu is the corresponding author. Email addresses: xiaolingzhou@tju.edu.cn (Xiaoling Zhou), wuou@tju.edu.cn (Ou Wu), yny@tju.edu.cn (Nan Yang).*

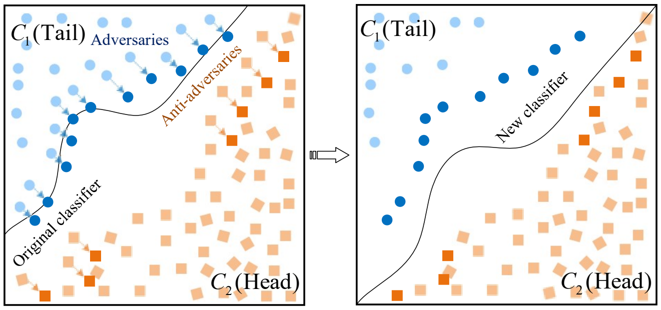


Fig. 2. Illustration for adversaries and anti-adversaries with varied bounds in imbalance learning. Anti-adversaries (adversaries) are obtained by perturbing the samples along the direction away from (close to) the classifier. Samples in the head/tail category are anti-adversarially/adversarially perturbed. Besides, boundary samples are perturbed with small bounds. Thus, fairness between classes can be improved, and a better tradeoff between the robustness and accuracy of the model can be attained.

these samples should be perturbed in the anti-adversarial direction to reduce their negative influence on model optimization. Zhu et al. [13] re-annotated pseudo labels for possible noisy samples before generating adversaries for them. The generated adversaries are actually perturbed anti-adversarially in binary classification tasks. In this study, samples with anti-adversarial perturbations are called anti-adversaries[1] ($\boldsymbol{x}_{\text{at-adv}}$):

$$\boldsymbol{x}_{\text{at-adv}} = \boldsymbol{x} + \arg\min_{\|\boldsymbol{\delta}\| \leq \epsilon} \ell(f(\boldsymbol{x} + \boldsymbol{\delta}), y). \tag{2}$$

The examples of adversaries and anti-adversaries for three instances in the SVHN [15] dataset are shown in Fig. 1, in which adversarial (anti-adversarial) perturbation makes samples harder (easier) than the original ones. In addition, different samples are supposed to have varied perturbation bounds. For instance, several previous studies [16], [17], [18] have argued that samples close to the boundary should be assigned with small bounds to avoid the model giving up these samples.

This study conducts comprehensive theoretical analyses of adversarial training with two different perturbation directions (adversarial and anti-adversarial) and varied bounds. Several typical learning scenarios are considered, including classes with different learning difficulties, imbalance learning, noisy label learning, and classes with skewed distributions. Our theoretical findings reveal that the perturbation directions and bounds can remarkably influence the model training. The combination of two perturbation directions and varied bounds can improve fairness among classes and achieve a better tradeoff among accuracy, robustness, and fairness. We illustrate the imbalanced learning occasion in Fig. 2, in which better fairness and tradeoff are attained by combining adversaries and anti-adversaries in training with varied perturbation bounds. Accordingly, a novel objective that combines adversaries and anti-adversaries with varied perturbation bound for each sample is constructed for adversarial training. In this objective, individual samples possess distinct perturbation strategies, incorporating both directions and bounds. These strategies ought to undergo dynamic optimization throughout the training process instead of being manually specified.

Accordingly, two algorithms based on meta-learning and reinforcement learning are further proposed to optimize the objective, in which the perturbation directions and bounds of samples are determined by their training characteristics, such as training loss and margin. Then, the role of the combination strategy with varied bounds is analyzed from a regularization aspect, demonstrating that training with anti-adversaries is a form of anti-regularization, whose strength is controlled by the perturbation bound. Thus, over-regularization caused by standard adversarial training can be overcome and better fairness among classes can be achieved. Extensive experiments are conducted under various learning scenarios, including standard learning, noisy label learning, and imbalance learning, demonstrating that the proposed **C**ombining **A**dversaries and **AnT**i-adversaries (CAAT) framework outperforms state-of-the-art adversarial training methods. In addition, our experimental observations are in accordance with our theoretical findings.

The contributions of our study are as follows:

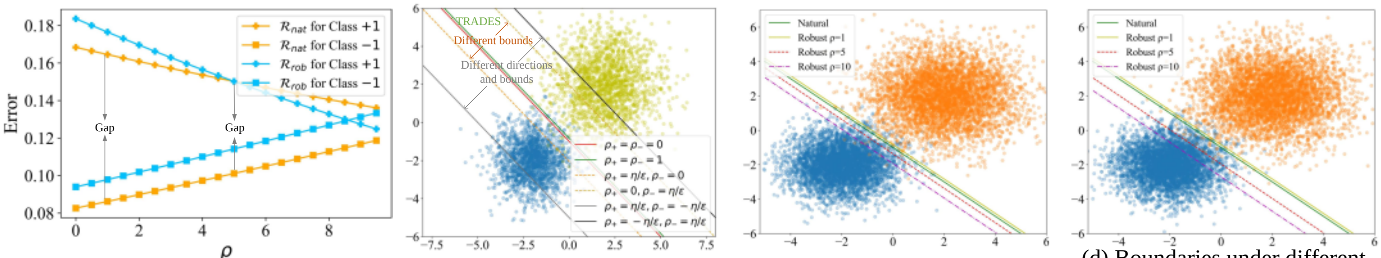
- To the best of our knowledge, this is the first work that combines adversaries and anti-adversaries in training with varied perturbation bounds. A comprehensive theoretical analysis is conducted for the role of different perturbation directions and varied bounds[2] under four typical learning scenarios.
- A new objective is established for adversarial training by combining adversaries and anti-adversaries with varied perturbation bounds. Meta-learning and reinforcement learning are utilized to solve the optimization. The perturbation direction and bound for each sample are determined in accordance with its learning characteristics, such as loss and margin.
- The role of the combination strategy with varied bounds is explained from a regularization view, revealing that training with anti-adversaries is a form of anti-regularization, whose strength is controlled by the perturbation bound. This learning strategy facilitates preventing over-regularization and achieving better fairness among classes.
- Extensive experiments under different learning scenarios verify that the proposed CAAT can achieve state-of-the-art performance in attaining the tradeoff between robustness and accuracy and improving fairness among classes.

2 Related Work

Tradeoff and Fairness in Adversarial Training. Recent studies on adversarial training focus on the tradeoff between accuracy and robustness. Efforts [5], [19], [20], [21] have been made to reduce the natural errors of the adversarially trained models, such as adversarial training with semi/unsupervised learning and robust local feature [22]. Rice et al. [9] systematically investigated the role of various

[1] The anti-adversary defined by Alfarra et al. [14] is different from ours. Their perturbation term is obtained by optimizing $\text{argmin}_{\boldsymbol{\delta}} \ell(f(\boldsymbol{x} + \boldsymbol{\delta}), \hat{y})$, where $\hat{y}$ is the prediction of sample $\boldsymbol{x}$. In addition, they utilize anti-adversaries to deal with attacks, whereas we aim to improve the robustness, accuracy, and fairness of models.

[2] In contrast, existing *theoretical* studies presume that the perturbation directions and bounds are identical for all training samples.



(a) Performance gap (b) Boundary scope (c) Boundaries under different bounds (d) Boundaries under different directions and bounds

Fig. 3. (a) Variation of performance gaps between classes as ρ increases under adversarial training with varied bounds. (b) Scope of the classification boundary under different manners, including natural training (red line), standard adversarial training (green line), TRADES, adversarial training with different bounds, and adversarial training with different directions and bounds. The parameters are $K=2$, $\eta=2$, $\epsilon=0.2$, and $\sigma=1$. The bounds for class "$+1$" and "-1" are denoted as $\rho_+ \times \epsilon$ and $\rho_- \times \epsilon$ $(-\eta/\epsilon < \rho_+, \rho_- < \eta/\epsilon)$, respectively. $\rho_+(\rho_-) < 0$ means that class "$+1(-1)$" is anti-adversarially perturbed. The formulas of all boundaries are provided in the supplements (Eqs. (A.32)-(A.35)). (c) Logistic regression classifiers (natural and robust) on simulated data in Eq. (3). (d) Logistic regression classifiers (natural and robust with different directions) on simulated data.

techniques used in deep learning for achieving a better tradeoff, such as cutout, mixup, and early stopping, where early stopping is found to be the most effective. This investigation was also confirmed by Pang et al. [23]. Unfairness among classes is also a problem caused by adversarial training. Xu et al. [11] trained a robust classifier to minimize errors and stressed it to satisfy two fairness constraints. Several studies [16], [17], [18] adaptively tune the perturbation bound for each sample with the inspiration that samples near the decision boundary should have small bounds.

Meta-learning. Meta-learning has aroused great interest from researchers in recent years [24]. Existing meta-learning methods can be divided into three categories, namely, metric-based [25], [26], model-based [27], and optimizing-based [28], [29] algorithms. The method we adopted that is inspired by Model-Agnostic Meta-Learning [28] belongs to the optimizing-based methods. The data-driven manner of meta optimization is always utilized to learn the weights or the parameters of deep neural networks [30], [31].

Reinforcement Learning. Reinforcement learning [32], [33], [34] studies how natural and artificial systems learn to predict consequences and optimize their behavior in the environment from one state/situation to another. In this subsection, we mainly retrospect the policy-based algorithms [35], [36], which directly optimize the policy to obtain the optimal strategy. The specific mechanism is to design an objective function and utilize the gradient ascent to optimize the parameters and maximize the expected return [36], [37]. This manner is also adopted by our algorithm.

3 Theoretical Investigation

This section conducts comprehensive theoretical analyses to assess the influence of two different perturbation directions and varied bounds on adversarial training in four typical binary classification cases. All proofs are presented in the supplementary material.

3.1 Notation

We denote the sample instance as $\boldsymbol{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$ as the label, where $\mathcal{X} \subseteq \mathbb{R}^d$ indicates the instance space, and $\mathcal{Y} = \{-1, +1\}$ indicates the label space. The classification model f maps the input data space $\mathcal{X}$ to the label space $\mathcal{Y}$. It can be parameterized by using linear classifiers or deep neural networks. The overall natural error of f is denoted as $\mathcal{R}_{\text{nat}}(f) := \Pr(f(\boldsymbol{x}) \neq y)$. The overall robust error is denoted as $\mathcal{R}_{\text{rob}}(f) := \Pr(\exists ||\boldsymbol{\delta}|| \leq \epsilon, \text{s.t.} f(\boldsymbol{x} + \boldsymbol{\delta}) \neq y)$.

3.2 Case I: Classes with Different Difficulties

In this case, the binary setting established by Xu et al. [11] is followed. The data from each class follow a Gaussian distribution $\mathcal{D}$ that is centered on $\boldsymbol{\theta}$ and $-\boldsymbol{\theta}$, respectively. A K-factor difference is found between two classes' variances: $\sigma_{+1} : \sigma_{-1} = K : 1$ and $K > 1$. Thus, the data follow

$$y \overset{u.a.r}{\sim} \{-1, +1\}, \boldsymbol{\theta} = (\eta, \ldots, \eta) \in \mathbb{R}^d, \eta > 0,$$
$$\boldsymbol{x} \sim \begin{cases} \mathcal{N}\left(\boldsymbol{\theta}, \sigma_{+1}^2 \boldsymbol{I}\right), & \text{if } y = +1, \\ \mathcal{N}\left(-\boldsymbol{\theta}, \sigma_{-1}^2 \boldsymbol{I}\right), & \text{if } y = -1. \end{cases} \tag{3}$$

Class "$+1$" is harder because the optimal linear classifier under natural training will give a larger error to class "$+1$" than to "-1". Xu et al. [11] proved that adversarial training with an equal bound would exacerbate the performance gap (including natural and robust errors) between classes and hurt the harder class. We prove that adversarial training with unequal bounds on two classes can tune the performance gap and the tradeoff between robustness and accuracy. Let $\sigma_{-1} = \sigma$. Theorem 1 calculates the errors of two classes utilizing adversarial training with unequal bounds.

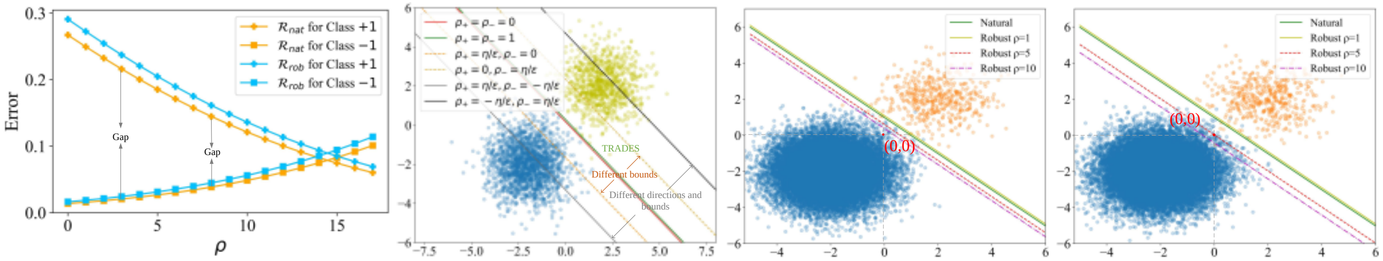
Theorem 1. *For a data distribution $\mathcal{D}$ in Eq. (3), assume that the perturbation bounds of class "-1" and class "$+1$" are ϵ and $\rho \times \epsilon$ $(0 \leq \epsilon, \rho\epsilon < \eta)$, respectively. The optimal robust linear classifier f_{rob} which minimizes the average robust error is*

$$\begin{aligned} f_{rob} = \arg\min_f \{ &\Pr(\exists \|\boldsymbol{\delta}\| \leq \epsilon,\ s.t. f(\boldsymbol{x} + \boldsymbol{\delta}) \neq y \mid y = -1) \\ &+ \Pr(\exists \|\boldsymbol{\delta}\| \leq \rho \times \epsilon,\ s.t. f(\boldsymbol{x} + \boldsymbol{\delta}) \neq y \mid y = +1)\}. \end{aligned} \tag{4}$$

It has natural errors for the two classes:

$$\begin{aligned} &\mathcal{R}_{nat}\left(f_{rob}, -1\right) \\ &= \Pr\left\{\mathcal{N}(0,1) \leq B - K \cdot \sqrt{B^2 + q(K)} - \frac{\sqrt{d}}{\sigma}\epsilon\right\}, \\ &\mathcal{R}_{nat}\left(f_{rob}, +1\right) \\ &= \Pr\left\{\mathcal{N}(0,1) \leq -K \cdot B + \sqrt{B^2 + q(K)} - \frac{\sqrt{d}\rho}{K\sigma}\epsilon\right\}, \end{aligned} \tag{5}$$

where $B = \frac{2}{K^2-1} \frac{\sqrt{d}(\eta - \frac{\epsilon(\rho+1)}{2})}{\sigma}$ and $q(K) = \frac{2\log K}{K^2-1}$. Additionally, there is $\mathcal{R}_{\text{nat}}(f_{\text{nat}}, +1) > \mathcal{R}_{\text{nat}}(f_{\text{nat}}, -1)$.



(a) Performance gap (b) Boundary scope (c) Boundaries under different bounds (d) Boundaries under different directions and bounds

Fig. 4. (a) Variation of performance gaps between classes as ρ increases under adversarial training with varied bounds on imbalanced data.(b): Scope of the classification boundary under different manners on imbalanced data. The parameters are $V = 2$, $\eta = 2$, $\epsilon = 0.8$, and $\sigma = 1$. The formulas of all boundaries are provided in the supplementary material (Eqs. (A.71)-(A.74)). (c): Logistic regression classifiers (natural and robust) on simulated data in Eq. (6). (d): Logistic regression classifiers (natural and robust with different directions). The imbalance factor V is set to 50.

The robust errors are shown in the supplements (Eq. (A.9)). The natural and robust errors change with different ρ values. A corollary depicting the variation of the performance gap between classes under adversarial training with unequal bounds is derived according to Theorem 1.

Corollary 1. *The data and perturbations in Theorem 1 are followed. When $K < \exp(d(\eta-\epsilon)^2/2\sigma^2)$, the performance gaps between classes (i.e., $|\mathcal{R}_{\mathrm{nat}}(f_{\mathrm{rob}}, +1) - \mathcal{R}_{\mathrm{nat}}(f_{\mathrm{rob}}, -1)|$ and $|\mathcal{R}_{\mathrm{rob}}(f_{\mathrm{rob}}, +1) - \mathcal{R}_{\mathrm{rob}}(f_{\mathrm{rob}}, -1)|$) decrease with the increase in ρ and thus better fairness can be attained. During this process, the scope of the decision boundary is $d\eta$.*

From Corollary 1, better inter-class fairness can be attained through adversarial training with varied bounds, as shown in Fig. 3(a). From Fig. 3(c), the boundary shifts towards the easy class ("-1") when the hard class ("$+1$") has a larger adversarial bound than that of the easy one. From Fig. 3(b), adversarial training with varied bounds contributes to the larger scope (i.e., $d\eta$) of the boundary compared with TRADES [5] and standard adversarial training. Thus, a better tradeoff among accuracy, robustness, and fairness can be attained. Next, anti-adversaries are considered, in which samples in class "-1" perform anti-adversarial perturbation. Similar to Theorem 1, a theorem is proposed to calculate the errors of classes when adversaries and anti-adversaries are combined in training with varied bounds, which is placed in the supplementary material (Theorem A.2). A corollary can then be derived.

Corollary 2. *For a data distribution $\mathcal{D}$ in Eq. (3), assume that class "-1" is anti-adversarially perturbed with the bound ϵ, and class "$+1$" is adversarially perturbed with the bound $\rho \times \epsilon$ $(0 \leq \epsilon, \rho\epsilon < \eta)$. When $K < \exp(d(\eta-\epsilon)^2/2\sigma^2)$, the performance gaps between classes (i.e., $|\mathcal{R}_{\mathrm{nat}}(f_{\mathrm{rob}}, +1) - \mathcal{R}_{\mathrm{nat}}(f_{\mathrm{rob}}, -1)|$ and $|\mathcal{R}_{\mathrm{rob}}(f_{\mathrm{rob}}, +1) - \mathcal{R}_{\mathrm{rob}}(f_{\mathrm{rob}}, -1)|$) decrease with the increase in ρ and thus better fairness can be attained. Additionally, the boundary scope during this process is $2d\eta$ which contains that of using only adversaries.*

In accordance with Corollaries 1 and 2, adversarial training with unequal bounds and the combination strategy with varied bounds can nearly attain the same performance. Nevertheless, the integration of anti-adversaries contributes to the largest scope (i.e., $2d\eta$) of the classification boundary, as shown in Fig. 3(b). Thus, it is more effective in achieving a better tradeoff among accuracy, robustness, and fairness theoretically. As shown in Figs. 3(c) and(d), the combination strategy has a more pronounced effect under the same bound (i.e., the same ρ) compared with using only adversaries, indicating that it needs smaller bounds when the same performance is achieved. Therefore, the combination strategy is more effective than using only adversarial perturbation, indicating that anti-adversaries are valuable.

3.3 Case II: Classes with Imbalanced Proportions

In this case, the two variances in Eq. (3) are assumed to be identical[3], that is, $\sigma_{+1} = \sigma_{-1} = \sigma$. However, $p(y = +1)$ (p_+) is no longer equal to $p(y = -1)$ (p_-). Without loss of generality, let $p_+ : p_- = 1 : V$ and $V > 1$. The data follow

$$\begin{aligned} &\Pr(y = +1) = p_+, \Pr(y = -1) = p_-, \\ &\boldsymbol{\theta} = (\overbrace{\eta, \ldots, \eta}^{\mathrm{dim}=d}), \\ &\boldsymbol{x} \sim \begin{cases} \mathcal{N}\left(\boldsymbol{\theta}, \sigma^2 I\right), & \text{if } y = +1, \\ \mathcal{N}\left(-\boldsymbol{\theta}, \sigma^2 I\right), & \text{if } y = -1. \end{cases} \end{aligned} \tag{6}$$

Class "-1" is the majority category, and an optimal linear classifier under natural training will give a smaller error to class "-1" than to "$+1$", as stated in Theorem 2.

Theorem 2. *For a data distribution $\mathcal{D}_V$ in Eq. (6) with the imbalance factor V, the optimal linear classifier under natural training f_{nat} which minimizes the average natural error is*

$$f_{nat} = \arg\min_f \Pr(f(\boldsymbol{x}) \neq y). \tag{7}$$

It has natural errors for the two classes:

$$\begin{aligned} \mathcal{R}_{nat}\ (f_{nat}, -1) &= \Pr\left\{\mathcal{N}(0,1) \leq -A - \frac{\log V}{2A}\right\}, \\ \mathcal{R}_{nat}\ (f_{nat}, +1) &= \Pr\left\{\mathcal{N}(0,1) \leq -A + \frac{\log V}{2A}\right\}, \end{aligned} \tag{8}$$

where $A = \frac{\sqrt{d}\eta}{\sigma}$. As a result, class "$+1$" has a larger natural error:

$$\mathcal{R}_{nat}\ (f_{nat}, -1) < \mathcal{R}_{nat}\ (f_{nat}, +1)\,. \tag{9}$$

Theorem 2 demonstrates that class "$+1$" which has a small prior probability is harder to be classified than the dominant class ("-1") under natural training. The class-wise difference is due to the prior probability ratio V. If

[3] The case with different variances can be explored similarly.

the two classes' prior probabilities are equal, i.e., $V = 1$, the natural errors for the two classes are the same.

Then we prove that standard adversarial training will exacerbate the class performance gap. Nevertheless, adversarial training with unequal bounds on the two classes can tune the performance gap and the tradeoff among robustness, accuracy, and fairness. Theorem 3 calculates the errors of two classes utilizing adversarial training with varied perturbation bounds.

Theorem 3. *For a data distribution $\mathcal{D}_V$ defined in Eq. (6) with the imbalance factor V, assume that the perturbation bounds of classes "-1" and "$+1$" are ϵ and $\rho \times \epsilon$ $(0 \leq \epsilon, \rho\epsilon < \eta)$, respectively. The natural errors of the optimal robust linear classifier f_{rob} for the two classes are*

$$\mathcal{R}_{nat}\left(f_{rob}, -1\right) = \Pr\left\{\mathcal{N}(0,1) \leq -A - \frac{\log V}{2A} - \frac{\sqrt{d}}{\sigma}\epsilon\right\},$$
$$\mathcal{R}_{nat}\left(f_{rob}, +1\right) = \Pr\left\{\mathcal{N}(0,1) \leq -A + \frac{\log V}{2A} - \frac{\sqrt{d}\rho}{\sigma}\epsilon\right\}, \tag{10}$$

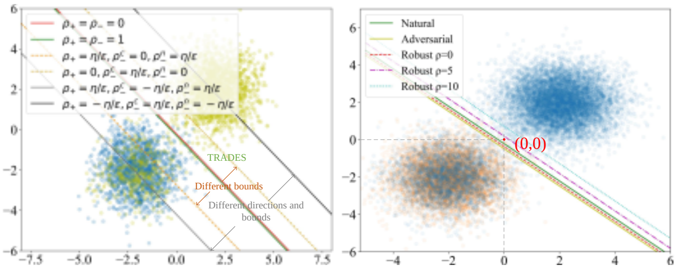
where $A = \frac{\sqrt{d}(\eta - \frac{\epsilon(\rho+1)}{2})}{\sigma}$.

The robust errors are presented in the supplements (Eq. (A.48)). In accordance with Theorems 2 and 3, when ρ in Eq. (10) equals 1, that is, utilizing standard adversarial training, the performance gap will be enlarged, as shown in Fig. 4(c). A corollary depicting the variation of the performance gap between classes under adversarial training with unequal bounds is then derived according to Theorem 3.

Corollary 3. *The data and perturbations in Theorem 3 are followed. When $V < \exp(d(\eta - \epsilon)^2/2\sigma^2)$, the performance gaps between classes (i.e., $|\mathcal{R}_{\mathrm{nat}}(f_{\mathrm{rob}}, +1) - \mathcal{R}_{\mathrm{nat}}(f_{\mathrm{rob}}, -1)|$ and $|\mathcal{R}_{\mathrm{rob}}(f_{\mathrm{rob}}, +1) - \mathcal{R}_{\mathrm{rob}}(f_{\mathrm{rob}}, -1)|$) decrease with the increase in ρ and thus better fairness can be attained. During this process, the scope of the decision boundary is $d\eta$.*

As stated in Corollary 3, the performance gaps can be decreased with different bounds, as shown in Fig. 4(a). In addition, the boundary can be moved with different ρ values within the scope (i.e., $d\eta$) that covers the boundaries of the standard adversarial training and TRADES, as shown in Fig. 4(b). From Fig. 4(c), adversarial training with varied bounds can attain a better tradeoff between robustness and accuracy as the obtained classifier is closer to the optimal classifier which passes through point (0,0). Next, anti-adversaries are considered, and a theorem calculating the errors when adversaries and anti-adversaries are combined in training with varied bounds is proposed, which is placed in the supplementary material (Theorem A.5). Then, a corollary regarding the combination strategy is derived.

Corollary 4. *For a data distribution $\mathcal{D}_V$ in Eq. (6), assume that class "-1" is anti-adversarially perturbed with the perturbation bound ϵ, and class "$+1$" is adversarially perturbed with the bound $\rho \times \epsilon$ $(0 \leq \epsilon, \rho\epsilon < \eta)$. When $V < \exp(d(\eta - \epsilon)^2/2\sigma^2)$, the performance gaps between classes (i.e., $|\mathcal{R}_{\mathrm{nat}}(f_{\mathrm{rob}}, +1) - \mathcal{R}_{\mathrm{nat}}(f_{\mathrm{rob}}, -1)|$ and $|\mathcal{R}_{\mathrm{rob}}(f_{\mathrm{rob}}, +1) - \mathcal{R}_{\mathrm{rob}}(f_{\mathrm{rob}}, -1)|$) decrease with the increase in ρ and thus better fairness can be attained. Additionally, the boundary scope during this process is $2d\eta$ which contains that of using only adversaries.*



(a) Boundary scope

(b) Boundaries under different directions and bounds

Fig. 5. (a): Scope of the classification boundary under different manners on noisy data. The parameters are $p = 0.2$, $\eta = 2$, $\epsilon = 0.2$, and $\sigma = 1$. The bounds for samples in class "+1", clean samples in class "-1", and noisy samples in class "-1" are denoted as $\rho_+ \times \epsilon$, $\rho_-^{\mathrm{c}} \times \epsilon$, $\rho_-^{\mathrm{n}} \times \epsilon$ $(-\eta/\epsilon < \rho_+, \rho_-^{\mathrm{c}}, \rho_-^{\mathrm{n}} < \eta/\epsilon)$, respectively. The formulas of all boundaries are provided in the supplementary material (Eqs. (A.113)-(A.116)). (b): Logistic regression classifiers (natural and robust with different directions) on simulated binary data in Eq. (11). Noisy samples are anti-adversarially perturbed with the perturbation bound $\rho \times \epsilon$, and clean samples are adversarially perturbed with the bound ϵ. The flipping rate p is set to 0.2.

From Corollary 4, better fairness can be attained by the combination strategy. As shown in Fig. 4(b), it can contribute to the largest boundary scope, i.e., $2d\eta$, compared with only adversaries. From the example in Figs. 4(c) and (d), classification boundaries that are closer to the Bayesian optimal classifier can be achieved through the combination strategy with varied bounds compared with using only adversaries. Thus, a superior tradeoff between accuracy and robustness can be attained. The above example also demonstrates that combining adversaries and anti-adversaries only requires a smaller bound than using only adversaries when the same performance is achieved. Therefore, the combination strategy is more effective than only adversarial perturbations in imbalance learning.

3.4 Case III: Classes with Noisy Labels

In this case, the two classes' variances and prior probabilities are assumed to be identical, that is, $\sigma_{+1} = \sigma_{-1}$ and $p_+ = p_-$. Without loss of generality, class "-1" is assumed to contain flipped noisy labels. The data follow

$$\begin{aligned}
&\tilde{y} \overset{u.a.r}{\sim} \{-1, +1\}, \\
&y = \begin{cases} +1 & \tilde{y} = +1, \\ +1 & \text{with a probability } p \text{ and } \tilde{y} = -1, \\ -1 & \text{with a probability } 1 - p \text{ and } \tilde{y} = -1, \end{cases} \\
&\boldsymbol{\theta} = (\overbrace{\eta, \ldots, \eta}^{\dim = d}), \\
&\boldsymbol{x} \sim \begin{cases} \mathcal{N}\left(\boldsymbol{\theta}, \sigma^2 I\right), & \text{if } y = +1, \\ \mathcal{N}\left(-\boldsymbol{\theta}, \sigma^2 I\right), & \text{if } y = -1, \end{cases}
\end{aligned} \tag{11}$$

where p (< 1) is the flipping rate for class "-1". Intuitively, class "-1" is harder than class "$+1$" as it contains noisy labels. Theorem 4 demonstrates that the error of class "-1" which contains label noise is larger than that of class "$+1$" under natural training.

Theorem 4. *For a data distribution $\mathcal{D}_N$ in Eq. (11) with the flipping rate p, the optimal linear classifier f_{nat} under natural training which minimizes the average natural error is*

$$f_{nat} = \arg\min_f \Pr(f(\boldsymbol{x}) \neq y). \tag{12}$$

It has natural errors for the two classes:

$$\begin{aligned}\mathcal{R}_{nat}\left(f_{nat},-1\right)&=\Pr\left\{\mathcal{N}(0,1)\le -A+\frac{\log\sqrt{\frac{1+p}{1-p}}}{A}\right\},\\ \mathcal{R}_{nat}\left(f_{nat},+1\right)&=\Pr\left\{\mathcal{N}(0,1)\le -A-\frac{\log\sqrt{\frac{1+p}{1-p}}}{A}\right\},\end{aligned}\tag{13}$$

where $A=\frac{\sqrt{d}\eta}{\sigma}$. *As a result, class "$-1$" has a larger natural error:*

$$\mathcal{R}_{nat}\ \left(f_{nat}\ ,+1\right)<\mathcal{R}_{nat}\ \left(f_{nat}\ ,-1\right).\tag{14}$$

We then prove that standard adversarial training exacerbates the performance gap between classes. The theorem is presented in the supplementary material (Theorem A.7). As shown in Figs. 5(a) and (b), standard adversarial training forces the classification boundary to move towards the noisy class ("-1"). Subsequently, we theoretically verify that adversarial training with unequal bounds for samples (0 for noisy samples and ϵ for clean ones) can decrease the performance gap between classes and improve the tradeoff between robustness and accuracy, compared with standard adversarial training, as illustrated in Fig. 5(b). However, the obtained classifier is still far from the optimal classifier (passing through point (0,0)), indicating that the balancing capability achieved by solely relying on adversaries is very limited. Corresponding theorem depicting adversarial training with varied bounds is presented in the supplementary material (Theorem A.8). Next, we prove that combining adversaries and anti-adversaries in training can tune the performance gap between classes and the tradeoff between robustness and accuracy more efficiently. Theorem 5 calculates the errors of two classes when adversaries and anti-adversaries are combined in training.

Theorem 5. *For a data distribution $\mathcal{D}_N$ in Eq. (11) with the flipping rate p, assume that clean samples $\boldsymbol{x}\in\boldsymbol{X}^c$ are adversarially perturbed with the perturbation bound ϵ, and noisy samples $\boldsymbol{x}\in\boldsymbol{X}^n$ are anti-adversarially perturbed with the bound $\rho\times\epsilon$ ($0\le\epsilon,\rho\epsilon<\eta$). The optimal robust linear classifier f_{rob} which minimizes the average robust error is*

$$\begin{aligned}f_{rob}=\arg\min_f\{&\Pr(\exists\|\boldsymbol{\delta}\|\le\epsilon,\ s.t. f(\boldsymbol{x}+\boldsymbol{\delta})\ne y\mid\boldsymbol{x}\in\boldsymbol{X}^c)\\&+\Pr(\exists\|\boldsymbol{\delta}\|\le\rho\times\epsilon,\ s.t. f(\boldsymbol{x}+\boldsymbol{\delta})\ne y\mid\boldsymbol{x}\in\boldsymbol{X}^n)\}.\end{aligned}\tag{15}$$

It has natural errors for the two classes:

$$\begin{aligned}&\mathcal{R}_{rob}\left(f_{nat},-1\right)\\&=\Pr\left\{\mathcal{N}(0,1)\le -A+\frac{\log\sqrt{\frac{1+p}{1-p}}}{A}-\frac{p(\rho-1)\sqrt{d}}{\sigma}\epsilon-\frac{\sqrt{d}\epsilon}{\sigma}\right\},\\&\mathcal{R}_{rob}\left(f_{nat},+1\right)=\Pr\left\{\mathcal{N}(0,1)\le -A-\frac{\log\sqrt{\frac{1+p}{1-p}}}{A}-\frac{\sqrt{d}\epsilon}{\sigma}\right\},\end{aligned}\tag{16}$$

where $A=\frac{\sqrt{d}(\eta-\epsilon-\frac{p(\rho-1)}{2}\epsilon)}{\sigma}$.

According to Theorem 5, Corollary 5 can be deduced, which portrays how the performance gaps change with the increase in ρ under the combination strategy.

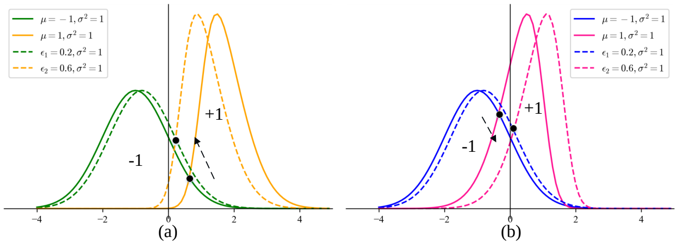


Fig. 6. (a): The occasion when the skewed distribution of class "$+1$" is far from the decision boundary ($\alpha>0$). The solid lines represent the distributions of the training set, and the dashed lines represent the perturbed distributions of the training set. The parameters are $\alpha=3$, $\eta=1.2$, and $\sigma=1$. (b): The occasion when the skewed distribution of class "$+1$" is close to the decision boundary ($\alpha<0$). The parameters are $\alpha=-3$, $\eta=2$, and σ =1.

Corollary 5. *The data and perturbations in Theorem 5 are followed. When $p<(e^{\frac{d(\eta-\epsilon)^2}{2\sigma^2}}-1)/(e^{\frac{d(\eta-\epsilon)^2}{2\sigma^2}}+1)$, the performance gaps between classes (i.e., $|\mathcal{R}_{\text{nat}}(f_{\text{rob}},+1)-\mathcal{R}_{\text{nat}}(f_{\text{rob}},-1)|$ and $|\mathcal{R}_{\text{rob}}(f_{\text{rob}},+1)-\mathcal{R}_{\text{rob}}(f_{\text{rob}},-1)|$) decrease with the increase in ρ and thus better fairness can be attained. During this process, the scope of the decision boundary is $2d\eta$.*

From Corollary 5, the fairness between classes can be tuned when adversaries and anti-adversaries are combined in training. Fig. 5(a) illustrates the boundary scope in different manners, in which the combination strategy has the largest scope (i.e., $2d\eta$). From Fig. 5(b), combining adversaries and anti-adversaries with varied perturbation bounds can make the classification boundary closer to the optimal classifier compared with using only adversaries. Thus, a better tradeoff among the robustness, accuracy, and fairness of the model can be attained. Our analysis also reveals that noisy samples should be perturbed anti-adversarially, manifesting that anti-adversaries are meaningful.

3.5 Case IV: Classes with Skewed Distributions

In this case, the two classes' variances and prior probabilities are assumed to be identical, i.e., $\sigma_{+1}=\sigma_{-1}=\sigma$ and $p_+=p_-$. Besides, there is no noisy sample in the two classes. However, due to some reasons such as improper data preprocessing or sampling, the data in a class follow a skewed distribution. To simplify the problem, we consider that the data are one-dimensional, i.e., $d=1$, which are assumed to be from two classes $\{-1,+1\}$. The data in class "-1" follow a Gaussian distribution $\mathcal{N}(-\theta,\sigma^2)$, while the training data of class "$+1$" follow a skewed distribution which is denoted as $S\mathcal{N}(\theta,\sigma^2,\alpha)$ [38], where α is the skew coefficient. The distribution is reduced to the normal distribution when $\alpha=0$. The probability density function of $S\mathcal{N}(\theta,\sigma^2,\alpha)$ [38] is $f(x;\theta,\sigma)=2\phi\left(x;\theta,\sigma\right)\Phi\left(\alpha(x-\theta)\right)$, where $\phi(x;\theta,\sigma)=\frac{1}{\sqrt{2\pi}}e^{-\frac{(x-\theta)^2}{2\sigma^2}}$ and $\Phi(x;\theta,\sigma)=\int_{-\infty}^{x}\phi(t;\theta,\sigma)\mathrm{d}t$. Thus, the data follow

$$\begin{aligned}&y\overset{u.a.r}{\sim}\{-1,+1\},\theta=\eta,\\&x\sim\begin{cases}S\mathcal{N}(\theta,\sigma^2,\alpha) & \text{if } y=+1,\\ \mathcal{N}\left(-\theta,\sigma^2\right) & \text{if } y=-1.\end{cases}\end{aligned}\tag{17}$$

We consider two occasions, including $\alpha>0$ and $\alpha<0$. Intuitively, under natural training, class "$+1$" is harder than class "-1" when $\alpha>0$, and class "$+1$" is easier than class

"-1" when $\alpha < 0$. Then, we prove that when $\alpha < 0$ ($\alpha > 0$), the error of class "$+1$" is smaller (larger) than that of class "-1" under natural training, as shown in Theorem 6.

Theorem 6. *For a data distribution $\mathcal{D}_\alpha$ in Eq. (17), which is one-dimensional with the skew coefficient α, assume that the optimal linear classifier of the two classes is $-\eta < x = x^* < \eta$. When $\alpha < (>)0$, then the optimal classification boundary $x = x^* < (>)0$ under natural training.*

From Theorem 6, when $\alpha < (>)0$, the error of the optimal classifier for class "$+1$" is smaller (larger) than that for class "-1" under natural training as the optimal classifier is biased towards class "-1" ("$+1$"). The class-wise difference is only due to the skew coefficient α. If $\alpha = 0$, then the errors of the two classes are the same. Next, we prove that if $\alpha > (<)0$, then the adversaries (anti-adversaries) of samples in class "$+1$" can help tune the performance gap between classes and the tradeoff among generalization, robustness, and fairness. Corollary 6 manifests that when $\alpha > 0$, adversarial training with varied bounds can help tune the performance gap between classes and the tradeoff among generalization, robustness, and fairness.

Corollary 6. *The data in Theorem 6 are followed, in which $\alpha > 0$. Assume that the adversarial perturbation bounds for classes "$+1$" and "-1" are $\rho \times \epsilon$ and ϵ ($0 \leq \epsilon, \rho\epsilon < \eta$), respectively. The performance gaps between classes (i.e., $|\mathcal{R}_{\rm nat}(f_{\rm rob}, +1) - \mathcal{R}_{\rm nat}(f_{\rm rob}, -1)|$ and $|\mathcal{R}_{\rm rob}(f_{\rm rob}, +1) - \mathcal{R}_{\rm rob}(f_{\rm rob}, -1)|$) decrease with the increase in ρ and thus better fairness can be attained.*

From Corollary 6, performance gaps between classes can be tuned using adversarial training with varied bounds. Moreover, a larger scope of the decision boundary can be achieved, encompassing that attained by standard adversarial training. As shown in Fig. 6(a), the decision boundary achieved through adversarial training with varied bounds is more closely aligned with the optimal classifier when compared to natural training. Corollary 7 indicates that when $\alpha < 0$, combing adversaries and anti-adversaries in training can help attain a better tradeoff among generalization, robustness, and fairness.

Corollary 7. *The data in Theorem 6 are followed, in which $\alpha < 0$. Assume that class "$+1$" is anti-adversarially perturbed with the perturbation bound $\rho \times \epsilon$, and class "-1" is adversarially perturbed with the bound ϵ ($0 \leq \epsilon, \rho\epsilon < \eta$). The performance gaps between classes (i.e., $|\mathcal{R}_{\rm nat}(f_{\rm rob}, +1) - \mathcal{R}_{\rm nat}(f_{\rm rob}, -1)|$ and $|\mathcal{R}_{\rm rob}(f_{\rm rob}, +1) - \mathcal{R}_{\rm rob}(f_{\rm rob}, -1)|$) decrease with the increase in ρ and thus better fairness can be attained.*

Accordingly, combining adversaries and anti-adversaries in training effectively decreases the performance gap between classes. Thus, the fairness can be improved. Moreover, this combination strategy compels the decision boundary to shift towards the optimal classification boundary, as shown in Fig. 6(b). Therefore, a better tradeoff among robustness, accuracy, and fairness can be attained.

3.6 Summarization

Our theoretical analyses comprehensively reveal that the perturbation directions and bounds notably influence the generalization, robustness, and fairness of the robust model under four typical learning scenarios. The main findings are summarized as follows: 1) Adversarial training with varied bounds enhances fairness between classes and achieves a better tradeoff among robustness, accuracy, and fairness, compared with standard adversarial training, as demonstrated in Corollaries 1, 3, and 6. 2) Combining adversaries and anti-adversaries in training with varied bounds achieves a superior tradeoff among robustness, accuracy, and fairness compared to using only adversaries, as manifested by Corollaries 2, 4, 5, and 7. 3) The combination strategy requires smaller perturbation bounds to achieve the same performance compared to using only adversaries, making it a more efficient approach. Existing studies ignored the valuable anti-adversaries. Thus, a new optimized objective that combines adversaries and anti-adversaries with varied perturbation bounds is proposed.

4 Methodology

Illuminated by the theoretical findings, a new objective function is first established, which combines adversaries and anti-adversaries in training with a varied perturbation bound for each sample. Meta-learning and reinforcement learning are always utilized to select parameters in sample weighting and perturbation [31], [36]. Accordingly, two manners which are based on meta-learning and reinforcement learning, respectively, are proposed to solve the optimization. Their structures are shown in Fig. 7.

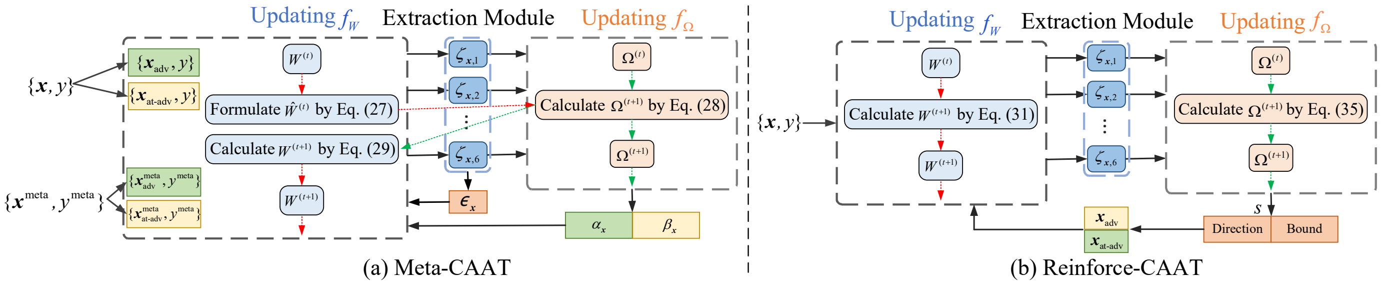
4.1 Proposed Objective Function

Inspired by our theoretical findings, the perturbation directions and bounds of samples are determined by the learning characteristics of samples ($\boldsymbol{\zeta}$), such as learning difficulty, imbalance ratio, noise degree, and skewness. Consequently, our proposed objective function that combines adversaries and anti-adversaries with varied bounds is formulated as

$$\begin{aligned}\min_{\boldsymbol{W}} \mathbb{E}_{\boldsymbol{x}} [& \max_{\boldsymbol{\Omega}} \mathbb{E}_{s^+ \sim p(s^+|\boldsymbol{\zeta}, \boldsymbol{\Omega})} \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}), y) \\ & + \min_{\boldsymbol{\Omega}} \mathbb{E}_{s^- \sim p(s^-|\boldsymbol{\zeta}, \boldsymbol{\Omega})} \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{at-adv}}), y)].\end{aligned} \tag{18}$$

The outermost optimization objective aims to minimize the loss of the classifier. The inner optimization objectives are designed to generate adversaries and anti-adversaries, respectively maximizing and minimizing the sample losses. s^+ and s^- refer to the perturbation strategies including directions and bounds for the adversaries $\boldsymbol{x}_{\text{adv}}$ and anti-adversaries $\boldsymbol{x}_{\text{at-adv}}$, respectively. These perturbation strategies are designed to be generated using a network parameterized by $\boldsymbol{\Omega}$, based on the training characteristics of samples $\boldsymbol{\zeta}$. $f_{\boldsymbol{W}}$ is the classifier with the parameter $\boldsymbol{W}$. The robust error $\ell(f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}), y)$ is then divided into the natural error $\ell(f_{\boldsymbol{W}}(\boldsymbol{x}), y)$ and the boundary error $\ell(f_{\boldsymbol{W}}(\boldsymbol{x}), f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}))$ to help achieve a better tradeoff between the accuracy and robustness [5]. To improve the fairness among classes, we further stress f to satisfy two fairness constraints following the manner in [11]. Thus, our objective is

$$\begin{aligned}\min_{\boldsymbol{W}} \mathbb{E}_{\boldsymbol{x}} \{ & \max_{\boldsymbol{\Omega}} \mathbb{E}_{s^+ \sim p(s^+|\boldsymbol{\zeta}, \boldsymbol{\Omega})} [\ell(f_{\boldsymbol{W}}(\boldsymbol{x}), y) + \lambda \ell(f_{\boldsymbol{W}}(\boldsymbol{x}), f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}))] \\ & + \min_{\boldsymbol{\Omega}} \mathbb{E}_{s^- \sim p(s^-|\boldsymbol{\zeta}, \boldsymbol{\Omega})} \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{at-adv}}), y) \}, \\ \text{s.t.} & \left\{ \begin{array}{l} \mathcal{R}_{\text{nat}}(f_{\boldsymbol{W}}, c) - \mathcal{R}_{\text{nat}}(f_{\boldsymbol{W}}) \leq \tau_1, \forall c \in \mathcal{Y}, \\ \mathcal{R}_{\text{bdy}}(f_{\boldsymbol{W}}, c) - \mathcal{R}_{\text{bdy}}(f_{\boldsymbol{W}}) \leq \tau_2, \forall c \in \mathcal{Y}, \end{array} \right.\end{aligned} \tag{19}$$



(a) Meta-CAAT

(b) Reinforce-CAAT

Fig. 7. The overall structure of Meta-CAAT (a) and Reinforce-CAAT(b). In Meta-CAAT, the red and green lines represent the learning loops of the classifier and the weighting network, respectively.

where $\mathcal{R}_{\text{bdy}}$ refers to the boundary error, denoted as $\mathcal{R}_{\text{bdy}}(f_{\boldsymbol{W}}) = \Pr(\exists \boldsymbol{x}_{\text{adv}} \in \mathbb{B}(\boldsymbol{x}, \epsilon), f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}) \neq f_{\boldsymbol{W}}(\boldsymbol{x}))$. Moreover, $\mathcal{R}_{\text{nat}}(f_{\boldsymbol{W}}, c) = \Pr(f_{\boldsymbol{W}}(\boldsymbol{x}) \neq y \mid y = c)$ and $\mathcal{R}_{\text{bdy}}(f_{\boldsymbol{W}}, c) = \Pr(\exists \boldsymbol{x}_{\text{adv}} \in \mathbb{B}(\boldsymbol{x}, \epsilon), f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}) \neq f_{\boldsymbol{W}}(\boldsymbol{x}) \mid y = c)$. $\lambda > 0$ is a regularization parameter that adjusts the influence of the natural and boundary errors; τ_1 and τ_2 are small and positive predefined parameters. The approach for solving the fairness constraints is the same as that of Xu et al. [11], where a Lagrangian is formed. Setting appropriate perturbation strategies for samples, including directions and bounds, is essentially a hyperparameter selection problem. Due to samples having their unique optimum values, conventional methods like grid search are impractical in such cases. Meta-learning and reinforcement learning have proven to be effective for hyperparameter selection [31], [36]. Consequently, we utilize these two methods to help optimize the hyperparameters of perturbations.

4.2 Extraction of Training Characteristics ($\zeta_{\boldsymbol{x}}$)

Accordingly, six training characteristics of samples are extracted from the classifier and input into the strategy network to generate the perturbation directions and bounds for training samples, as shown in the extraction module in Fig. 7. The six training characteristics are loss ($\boldsymbol{\zeta}_{\boldsymbol{x},1}$), margin ($\boldsymbol{\zeta}_{\boldsymbol{x},2}$), the norm of loss gradient for the logit vector ($\boldsymbol{\zeta}_{\boldsymbol{x},3}$), the information entropy of the softmax output ($\boldsymbol{\zeta}_{\boldsymbol{x},4}$), class proportion ($\boldsymbol{\zeta}_{\boldsymbol{x},5}$), and the average loss of each class ($\boldsymbol{\zeta}_{\boldsymbol{x},6}$), which are detailed below:

- Loss ($\boldsymbol{\zeta}_{\boldsymbol{x},1}$) is the most widely used factor to reflect the training behavior of samples [31].
- Margin ($\boldsymbol{\zeta}_{\boldsymbol{x},2}$) refers to the distance from the sample to the classification boundary [39], which is always utilized to measure the learning difficulty of samples. It is calculated by
$$\boldsymbol{\zeta}_{\boldsymbol{x},2} = f_{\boldsymbol{W}}(\boldsymbol{x})_{y_{\boldsymbol{x}}} - \max_{j \neq y_{\boldsymbol{x}}}(f_{\boldsymbol{W}}(\boldsymbol{x})_j), \tag{20}$$
where $f_{\boldsymbol{W}}(\boldsymbol{x})$ is the output of the softmax layer.
- The norm of loss gradient ($\boldsymbol{\zeta}_{\boldsymbol{x},3}$) is another commonly used characteristic [40]. As the cross-entropy loss is adopted, it can be calculated by
$$\boldsymbol{\zeta}_{\boldsymbol{x},3} = ||\boldsymbol{y}_{\boldsymbol{x}} - f_{\boldsymbol{W}}(\boldsymbol{x})||_2, \tag{21}$$
where $\boldsymbol{y}_{\boldsymbol{x}}$ is the one-hot label vector of sample $\boldsymbol{x}$.
- Information entropy ($\boldsymbol{\zeta}_{\boldsymbol{x},4}$) of $f_{\boldsymbol{W}}(\boldsymbol{x})$ is used to measure the uncertainty of training samples [41]. Its calculation is
$$\boldsymbol{\zeta}_{\boldsymbol{x},4} = -\sum\nolimits_{j=1}^{|\mathcal{Y}|} f_{\boldsymbol{W}}(\boldsymbol{x})_j \log_2(f_{\boldsymbol{W}}(\boldsymbol{x})_j), \tag{22}$$
where $\mathcal{Y}$ refers to the label set.
- Class proportion ($\boldsymbol{\zeta}_{\boldsymbol{x},5}$) is commonly used to handle imbalanced class distribution [42]. Its calculation is
$$\boldsymbol{\zeta}_{\boldsymbol{x},5} = N_{y_{\boldsymbol{x}}}/N, \tag{23}$$
where $N_{y_{\boldsymbol{x}}}$ and N are the numbers of samples in class $y_{\boldsymbol{x}}$ and in the entire training set, respectively.
- Average loss of each category ($\boldsymbol{\zeta}_{\boldsymbol{x},6}$) is another class-level factor indicating the average learning difficulty of samples in each class. Its calculation is
$$\boldsymbol{\zeta}_{\boldsymbol{x},6} = \bar{\ell}_{y_{\boldsymbol{x}}}, \tag{24}$$
where $\bar{\ell}_{y_{\boldsymbol{x}}}$ is the average loss of samples in class $y_{\boldsymbol{x}}$.

The above six characteristics will be input into the strategy network to generate the perturbation strategies of samples.

4.3 Meta-learning-based Manner

To solve the objective in Eq. (19), we first propose a meta-learning-based algorithm. In this manner, the strategy network is a weighting network, which generates the weights ($\alpha_{\boldsymbol{x}}$ and $\beta_{\boldsymbol{x}}$) of the losses for each sample's adversary and anti-adversary. The values of the weights ($\alpha_{\boldsymbol{x}}$ and $\beta_{\boldsymbol{x}}$) are supposed to be selected in $\{0, 1\}$ and their sum is 1. In this way, a sample is either adversarially or anti-adversarially perturbed. Thus, the objective is

$$\begin{aligned}
&\min_{\boldsymbol{W},\boldsymbol{\Omega}} \mathbb{E}_{\boldsymbol{x}}\{\alpha_{\boldsymbol{x}}[\ell(f_{\boldsymbol{W}}(\boldsymbol{x}), y) + \lambda\ell(f_{\boldsymbol{W}}(\boldsymbol{x}), f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}))] \\
&\quad + \beta_{\boldsymbol{x}}\ell(f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{at-adv}}), y)\}, \\
&\quad \text{s.t.} \begin{cases} [\alpha_{\boldsymbol{x}}, \beta_{\boldsymbol{x}}] = f_{\boldsymbol{\Omega}}(\boldsymbol{\zeta}_{\boldsymbol{x}}), \forall \boldsymbol{x} \in \mathcal{X}, \\ \mathcal{R}_{\text{nat}}(f_{\boldsymbol{W}}, c) - \mathcal{R}_{\text{nat}}(f_{\boldsymbol{W}}) \leq \tau_1, \forall c \in \mathcal{Y}, \\ \mathcal{R}_{\text{bdy}}(f_{\boldsymbol{W}}, c) - \mathcal{R}_{\text{bdy}}(f_{\boldsymbol{W}}) \leq \tau_2, \forall c \in \mathcal{Y}, \end{cases}
\end{aligned} \tag{25}$$

where $f_{\boldsymbol{\Omega}}$ is a multilayer perception (MLP) network with a hidden layer and a τ-softmax layer: Softmax$((\boldsymbol{h\omega} + \boldsymbol{b})/\tau)$, which can generate approximated one-hot vectors. As the gradient cannot be backpropagated to the values of the bound through the adversaries and anti-adversaries, we adopted two varied perturbation bounds to generate the bounds for samples, which are stated in the next subsection.

4.3.1 *Perturbation Bound ($\epsilon_{\boldsymbol{x}}$)*

The varied perturbation bound for each sample is calculated in the following two manners. Following Xu et al. [11], the class-wise perturbation bound named ReMargin, suitable for imbalanced data, is utilized. In addition, we propose a sample-wise bound to handle noise. It is inspired by the intuition that noisy samples generally have a large

Algorithm 1: Meta-CAAT

```
Input: Iteration $T$, step sizes $\eta_0$, $\eta_1$, and $\eta_2$, batch size $n$,
  meta batch size $m$, bound $\epsilon$, #iterations $K$ in inner
  optimization, classifier network $f_{\boldsymbol{W}}$, weighting
  network $f_{\boldsymbol{\Omega}}$, $D^{\text{train}}$, $D^{\text{meta}}$.
Output: Trained robust network $f_{\boldsymbol{W}}$.

Initialize networks $f_{\boldsymbol{W}}$ and $f_{\boldsymbol{\Omega}}$;
for $t = 1$ to $T$ do
    Sample $n$ and $m$ samples from $D^{\text{train}}$ and $D^{\text{meta}}$;
    for $i = 1$ to $n$ (in parallel) do
        $\boldsymbol{x}_i^{\text{adv}} = \boldsymbol{x}_i + 0.001\mathcal{N}(0,I)$ and $\boldsymbol{x}_i^{\text{at-adv}} = \boldsymbol{x}_i + 0.001\mathcal{N}(0,I)$,
        where $\mathcal{N}(0, I)$ is the Gaussian distribution;
        Calculate the perturbation bound $\epsilon_i$ for sample $\boldsymbol{x}_i$;
        for $k = 1$ to $K$ do
            $\boldsymbol{x}_i^{\text{adv}} \leftarrow$
            $\Pi_{\mathbb{B}(\boldsymbol{x}_i, \epsilon_i)}(\eta_0 \operatorname{sign}(\nabla_{\boldsymbol{x}_i^{\text{adv}}} \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i), f_{\boldsymbol{W}}(\boldsymbol{x}_i^{\text{adv}}))) + \boldsymbol{x}_i^{\text{adv}})$,
            where $\Pi$ is the projection operator;
            $\boldsymbol{x}_i^{\text{at-adv}} \leftarrow$
            $\Pi_{\mathbb{B}(\boldsymbol{x}_i, \epsilon_i)}(-\eta_0 \operatorname{sign}(\nabla_{\boldsymbol{x}_i^{\text{at-adv}}} \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i^{\text{at-adv}}), y_i)) +$
            $\boldsymbol{x}_i^{\text{at-adv}})$;
        end for
    end for
    Formulate $\hat{\boldsymbol{W}}^{(t)}(\boldsymbol{\Omega})$ by Eq. (27);
    Update $\boldsymbol{\Omega}^{(t+1)}$ by Eq. (28) and update $\boldsymbol{W}^{(t+1)}$ by
      Eq. (29);
end for
```

norm of loss gradient, and these samples should exhibit the greatest degree of anti-adversarial perturbation. Thus, the grad-based bound can be calculated as

$$\epsilon_{\boldsymbol{x}} = (\alpha_{\boldsymbol{x}} \bar{\boldsymbol{g}}_{\boldsymbol{x}_{\text{adv}}} + \beta_{\boldsymbol{x}} \bar{\boldsymbol{g}}_{\boldsymbol{x}_{\text{at-adv}}} + \varepsilon) \times \epsilon, \tag{26}$$

where $\bar{\boldsymbol{g}}_{\boldsymbol{x}_{\text{adv}}}$ and $\bar{\boldsymbol{g}}_{\boldsymbol{x}_{\text{at-adv}}}$ are the normalized $||\frac{\partial \ell(f_{\boldsymbol{W}}(\boldsymbol{x}), f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}))}{\partial \boldsymbol{x}_{\text{adv}}}||_2$ and $||\frac{\partial \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{at-adv}}), y)}{\partial \boldsymbol{x}_{\text{at-adv}}}||_2$, respectively. ϵ is a predefined bound, and ε is a hyperparameter that is set to 0.9 in our experiments. This bound is also effective on imbalanced data because samples in tail classes have large norms of loss gradient, and they should do the greatest degree of adversarial perturbation.

4.3.2 Training with Meta-learning

On the basis of the extracted training characteristics and calculated bounds, an online meta-learning-based learning strategy is adopted to alternatively update $\boldsymbol{W}$ and $\boldsymbol{\Omega}$ using a single optimization loop, as shown in Fig. 7(a). Assume that we have a small amount of unbiased meta data $D^{\text{meta}} = \{(\boldsymbol{x}_i^{\text{meta}}, y_i^{\text{meta}})\}_{i=1}^{M}$, where $M \ll N$. Even if meta data are lacking, they can be compiled from the training data D^{train} [43]. There are three main steps in this algorithm. We ignore the regularization terms introduced by the fairness constraints to facilitate writing. The supplementary material provides the complete formulas (Eqs. (A.119)-(A.121)).

First, $\boldsymbol{\Omega}$ is regarded as the to-be-updated parameter, and the parameter $\boldsymbol{W}$, which is a function of $\boldsymbol{\Omega}$, is formulated. Stochastic gradient descent (SGD) is utilized to optimize the training loss. Specifically, a batch of training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ is selected in each iteration, where n is the batch size. Then, the updating of $\boldsymbol{W}$ can be formulated as

$$\hat{\boldsymbol{W}}^{(t)}(\boldsymbol{\Omega}) = \boldsymbol{W}^{(t)} - \eta_1 \frac{1}{n} \sum\nolimits_{i=1}^{n} \nabla_{\boldsymbol{W}} \{\alpha_i [\ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i), y_i) + \lambda \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i), f_{\boldsymbol{W}}(\boldsymbol{x}_i^{\text{adv}}))] + \beta_i \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i^{\text{at-adv}}), y_i)\}|_{\boldsymbol{W}^{(t)}}, \tag{27}$$

where η_1 is the step size. The parameter of the weighting network $\boldsymbol{\Omega}$ after receiving feedback from the classifier can be updated on a batch of meta data as follows:

$$\boldsymbol{\Omega}^{(t+1)} = \boldsymbol{\Omega}^{(t)} - \eta_2 \frac{1}{m} \sum\nolimits_{i=1}^{m} \nabla_{\boldsymbol{\Omega}} \{\alpha_i [\ell^{\text{meta}}(f_{\hat{\boldsymbol{W}}^{(t)}(\boldsymbol{\Omega})}(\boldsymbol{x}_i), y_i) + \lambda \ell^{\text{meta}}(f_{\hat{\boldsymbol{W}}^{(t)}(\boldsymbol{\Omega})}(\boldsymbol{x}_i), f_{\hat{\boldsymbol{W}}^{(t)}(\boldsymbol{\Omega})}(\boldsymbol{x}_i^{\text{adv}}))] + \beta_i \ell^{\text{meta}}(f_{\hat{\boldsymbol{W}}^{(t)}(\boldsymbol{\Omega})}(\boldsymbol{x}_i^{\text{at-adv}}), y_i)\}|_{\boldsymbol{\Omega}^{(t)}}, \tag{28}$$

where m and η_2 are the batch size of meta data and the step size, respectively. The parameters of the classifier are finally updated with the obtained weights by fixing the parameters of the weighting network as $\boldsymbol{\Omega}^{(t+1)}$:

$$\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \eta_1 \frac{1}{n} \sum\nolimits_{i=1}^{n} \nabla_{\boldsymbol{W}} \{\alpha_i [\ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i), y_i) + \lambda \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i), f_{\boldsymbol{W}}(\boldsymbol{x}_i^{\text{adv}}))] + \beta_i \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i^{\text{at-adv}}), y_i)\}|_{\boldsymbol{W}^{(t)}}. \tag{29}$$

Our Meta-CAAT algorithm is shown in Algorithm 1.

4.4 Reinforcement-learning-based Manner

As the perturbation bounds of samples can only be calculated by predefined varied bounds in Meta-CAAT, we further propose a reinforcement-learning-based algorithm, in which both the perturbation direction and bound are generated by a strategy network $f_{\boldsymbol{\Omega}}$, as shown in Fig. 7(b). As with Meta-CAAT, $f_{\boldsymbol{\Omega}}$ is an MLP with a hidden layer. The strategy network captures the conditional distribution $p(s|\boldsymbol{\zeta}, \boldsymbol{\Omega})$ of the given training characteristics $\boldsymbol{\zeta}$ and $\boldsymbol{\Omega}$. The parameters of the classifier and the strategy network are updated iteratively. Omitting the fairness constraints, given $\boldsymbol{\Omega}$, the subproblem of optimizing the classifier is defined as

$$\min_{\boldsymbol{W}} \mathbb{E}_{\boldsymbol{x}} \{\mathbb{E}_{s^+ \sim p(s^+|\boldsymbol{\zeta}, \boldsymbol{\Omega})} [\ell(f_{\boldsymbol{W}}(\boldsymbol{x}), y) + \lambda \ell(f_{\boldsymbol{W}}(\boldsymbol{x}), f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}))] + \mathbb{E}_{s^- \sim p(s^-|\boldsymbol{\zeta}, \boldsymbol{\Omega})} \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{at-adv}}), y)\}. \tag{30}$$

We randomly sample a strategy from the conditional distribution $p(s|\boldsymbol{\zeta}, \boldsymbol{\Omega})$ which can be divided into $p(s^+|\boldsymbol{\zeta}, \boldsymbol{\Omega})$ and $p(s^-|\boldsymbol{\zeta}, \boldsymbol{\Omega})$. After collecting the adversaries and anti-adversaries for a batch of samples, we can update the parameters of the classifier through gradient descent:

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta_1 \frac{1}{n} \sum_{i=1}^{n} \nabla_{\mathbf{W}} \ell\left(f\left(\tilde{\boldsymbol{x}}_i\right), y_i\right)|_{\boldsymbol{W}^{(t)}}, \tag{31}$$

where $\tilde{\boldsymbol{x}}_i$ is the perturbed sample of $\boldsymbol{x}_i$. When $\tilde{\boldsymbol{x}}_i$ is the adversary of sample $\boldsymbol{x}_i$, $\ell\left(f\left(\tilde{\boldsymbol{x}}_i\right), y_i\right) = \ell(f\left(\boldsymbol{x}_i\right), y_i) + \lambda \ell(f\left(\boldsymbol{x}_i^{\text{adv}}\right), f\left(\boldsymbol{x}_i\right))$. When $\tilde{\boldsymbol{x}}_i$ is the anti-adversary of $\boldsymbol{x}_i$, $\ell\left(f\left(\tilde{\boldsymbol{x}}_i\right), y_i\right) = \ell(f\left(\boldsymbol{x}_i^{\text{at-adv}}\right), y_i)$. n is the number of samples in a mini-batch and η_1 is the learning rate.

Given $\boldsymbol{W}$, the subproblem of optimizing the strategy network can be written as

$$\max_{\boldsymbol{\Omega}} \mathbb{E}_{\boldsymbol{x}} \mathbb{E}_{s^+ \sim p(s^+|\boldsymbol{\zeta}, \boldsymbol{\Omega})} [\ell(f_{\boldsymbol{W}}(\boldsymbol{x}), y) + \lambda \ell(f_{\boldsymbol{W}}(\boldsymbol{x}), f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}))] + \min_{\boldsymbol{\Omega}} \mathbb{E}_{\boldsymbol{x}} \mathbb{E}_{s^- \sim p(s^-|\boldsymbol{\zeta}, \boldsymbol{\Omega})} \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{at-adv}}), y). \tag{32}$$

As the gradient cannot be backpropagated to the attack strategies through adversaries and anti-adversaries, we

Algorithm 2: Reinforce-CAAT

```
Input: Iteration $T$, step sizes $\eta_1$, and $\eta_2$, batch size $n$,
  interval $K$, classifier network $f_{\boldsymbol{W}}$, strategy network
  $f_{\boldsymbol{\Omega}}$, $D^{\text{train}}$.
Output: Trained robust network $f_{\boldsymbol{W}}$.

Initialize networks $f_{\boldsymbol{W}}$ and $f_{\boldsymbol{\Omega}}$;
for $t = 1$ to $T$ do
    Sample $n$ samples from $D^{\text{train}}$;
    Generate $\boldsymbol{x}_{\text{adv}}$ and $\boldsymbol{x}_{\text{at-adv}}$ for the $n$ samples using the
      strategies produced by $f_{\boldsymbol{\Omega}}$;
    Update $\boldsymbol{W}^{t+1}$ by Eq. (31);
    if $(t+1)\%K = 0$ then
        Update $\boldsymbol{\Omega}^{t+1}$ by Eq. (35);
    end if
end for
```

compute the derivative of the objective defined in Eq. (32) with respect to the parameters $\boldsymbol{\Omega}$ using the REINFORCE algorithm [37]. Denote $\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{s^+\sim p(s^+|\boldsymbol{\zeta},\boldsymbol{\Omega})}[\ell(f_{\boldsymbol{W}}(\boldsymbol{x}), y) + \lambda\ell(f_{\boldsymbol{W}}(\boldsymbol{x}), f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}))]$ as $J^+(\boldsymbol{\Omega})$. Its derivative is

$$\begin{aligned}\nabla_{\boldsymbol{\Omega}} J^+(\boldsymbol{\Omega}) &= \nabla_{\boldsymbol{\Omega}}\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{s^+\sim p(s^+|\boldsymbol{\zeta};\boldsymbol{\Omega})}\mathcal{L}_0(\boldsymbol{x}) \\ &= \mathbb{E}_{\boldsymbol{x}}\int_{s^+}\mathcal{L}_0(\boldsymbol{x})\cdot\nabla_{\boldsymbol{\Omega}} p(s^+ \mid \boldsymbol{\zeta};\boldsymbol{\Omega})ds^+ \\ &= \mathbb{E}_{\boldsymbol{x}}\int_{s^+}\mathcal{L}_0(\boldsymbol{x})\cdot p(s^+ \mid \boldsymbol{\zeta};\boldsymbol{\Omega})\nabla_{\boldsymbol{\Omega}}\log p(s^+ \mid \boldsymbol{\zeta};\boldsymbol{\Omega})ds^+ \\ &= \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{s^+\sim p(s^+|\boldsymbol{\zeta};\boldsymbol{\Omega})}[\mathcal{L}_0(\boldsymbol{x})\cdot\nabla_{\boldsymbol{\Omega}}\log p(s^+ \mid \boldsymbol{\zeta};\boldsymbol{\Omega})],\end{aligned} \tag{33}$$

where $\mathcal{L}_0(\boldsymbol{x}) = \ell(f_{\boldsymbol{W}}(\boldsymbol{x}), y) + \lambda\ell(f_{\boldsymbol{W}}(\boldsymbol{x}), f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}))$. In the same way, denote $\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{s^-\sim p(s^-|\boldsymbol{\zeta},\boldsymbol{\Omega})}\ell(f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{at-adv}}), y)$ as $J^-(\boldsymbol{\Omega})$. We obtain $\nabla_{\boldsymbol{\Omega}} J^-(\boldsymbol{\Omega}) = \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{s^-\sim p(s^-|\boldsymbol{\zeta};\boldsymbol{\Omega})}[\mathcal{L}_1(\boldsymbol{x})\cdot\nabla_{\boldsymbol{\Omega}}\log p(s^- \mid \boldsymbol{\zeta};\boldsymbol{\Omega})]$, where $\mathcal{L}_1(\boldsymbol{x}) = \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{at-adv}}), y)$.

Similar to solving Eq. (30), we sample the attack strategies from the conditional distribution of strategy i.e., $p(s|\boldsymbol{\zeta},\boldsymbol{\Omega})$, to generate adversaries and anti-adversaries. The gradient with respect to the parameters can be approximately computed as follows:

$$\begin{aligned}\nabla_{\boldsymbol{\Omega}} J^+(\boldsymbol{\Omega}) - \nabla_{\boldsymbol{\Omega}} J^-(\boldsymbol{\Omega}) \approx &\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_0(\boldsymbol{x}_i)\cdot\nabla_{\boldsymbol{\Omega}}\log p_{\boldsymbol{\Omega}}\left(s_i^+ \mid \boldsymbol{\zeta}_i\right) \\ &-\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_1(\boldsymbol{x}_i)\cdot\nabla_{\boldsymbol{\Omega}}\log p_{\boldsymbol{\Omega}}\left(s_i^- \mid \boldsymbol{\zeta}_i\right).\end{aligned} \tag{34}$$

Then, $\boldsymbol{\Omega}$ can be updated using

$$\boldsymbol{\Omega}^{(t+1)} = \boldsymbol{\Omega}^{(t)} + \eta_2[\nabla_{\boldsymbol{\Omega}} J^+(\boldsymbol{\Omega}) - \nabla_{\boldsymbol{\Omega}} J^-(\boldsymbol{\Omega})]|_{\boldsymbol{\Omega}^{(t)}}, \tag{35}$$

where η_2 is the step size. We update $\boldsymbol{\Omega}$ every K times of updating $\boldsymbol{W}$. The algorithm is presented in Algorithm 2.

4.5 Comparison of Two Manners

The advantages and disadvantages of the two algorithms are compared. The superiority of Meta-CAAT is that it can adjust the distribution information of the training set as the information in an additional high-quality dataset is utilized in each training epoch, which is beneficial for biased datasets. However, the construction of the meta dataset is sometimes challenging. In addition, the bounds of samples can only be calculated by predefined varied bounds as the gradient cannot be backpropagated to the perturbation bound through adversaries and anti-adversaries. Therefore, we further propose Reinforce-CAAT, in which both the perturbation directions and bounds can be generated by the strategy network. Furthermore, an additional meta dataset is not necessary. However, the defect is that it cannot adjust the distribution information of the training set.

5 Explanation from A Regularization View

We explain the role of the combination strategy with varied bounds from the regularization aspect. An existing study [44] has pointed out that adversarial training can be viewed as a special regularization manner. The optimized loss of adversarial training is

$$\tilde{\mathcal{L}}(\boldsymbol{x}, y) := \frac{1}{2}[\mathcal{L}(\boldsymbol{x}, y) + \mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta}, y)]. \tag{36}$$

If the perturbation term $\boldsymbol{\delta}$ is small enough, the above objective can be approximated using the first-order Taylor expansion, which becomes

$$\begin{aligned}\tilde{\mathcal{L}}(\boldsymbol{x}, y) \approx &\frac{1}{2}\left[\mathcal{L}(\boldsymbol{x}, y) + \mathcal{L}(\boldsymbol{x}, y) + \boldsymbol{\delta}\cdot\partial_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y)\right] \\ &= \mathcal{L}(\boldsymbol{x}, y) + \frac{1}{2}\boldsymbol{\delta}\cdot\partial_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y).\end{aligned} \tag{37}$$

The additional term introduced by sample perturbation is $\frac{1}{2}\boldsymbol{\delta}\cdot\partial_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y)$. For adversaries, we have

$$\begin{aligned}\boldsymbol{\delta}\cdot\partial_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y) &= \max_{\boldsymbol{\delta}:\|\boldsymbol{\delta}\|_p\le\epsilon}[\mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta}, y) - \mathcal{L}(\boldsymbol{x}, y)] \\ &\approx \max_{\boldsymbol{\delta}:\|\boldsymbol{\delta}\|_p\le\epsilon}[\partial_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y)\cdot\boldsymbol{\delta}] \\ &= \epsilon\left\|\partial_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y)\right\|_q,\end{aligned} \tag{38}$$

where $\frac{1}{p}+\frac{1}{q}=1$. Incorporating Eq. (38) into Eq. (37), we yield

$$\tilde{\mathcal{L}}(\boldsymbol{x}, y) \approx \mathcal{L}(\boldsymbol{x}, y) + \frac{\epsilon}{2}\left\|\partial_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y)\right\|_q. \tag{39}$$

Thus, adversarial training introduces a regularization term for the loss gradient of features, whose strength is controlled by the perturbation bound ϵ, helping the model to be insensitive to the perturbations. Alternatively, the predicted labels of samples in a small region (denoted as robust region) around a sample $\boldsymbol{x}_i$ are consistent with that of $\boldsymbol{x}_i$.

For anti-adversaries, we have

$$\begin{aligned}\boldsymbol{\delta}\cdot\partial_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y) &= \max_{\boldsymbol{\delta}:\|\boldsymbol{\delta}\|_p\le\epsilon}[\mathcal{L}(\boldsymbol{x}, y) - \mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta}, y)] \\ &\approx \min_{\boldsymbol{\delta}:\|\boldsymbol{\delta}\|_p\le\epsilon}[\partial_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y)\cdot\boldsymbol{\delta}] \\ &= -\epsilon\left\|\partial_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y)\right\|_q.\end{aligned} \tag{40}$$

Incorporating Eq. (40) into Eq. (37), we yield

$$\tilde{\mathcal{L}}(\boldsymbol{x}, y) \approx \mathcal{L}(\boldsymbol{x}, y) - \frac{\epsilon}{2}\left\|\partial_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y)\right\|_q. \tag{41}$$

Thus, training with anti-adversaries is a form of anti-regularization, which makes the model more sensitive to the perturbation of a sample. Alternatively, the robust regions of samples will be decreased compared with those under natural training. Thus, anti-regularization improves the accuracy of original samples. In our algorithm, both the regularization direction and strength can be adjusted adaptively, which is flexible.

Furthermore, we establish that distinct samples within a dataset should possess different perturbation directions and bounds. Let's consider a training set, denoted as

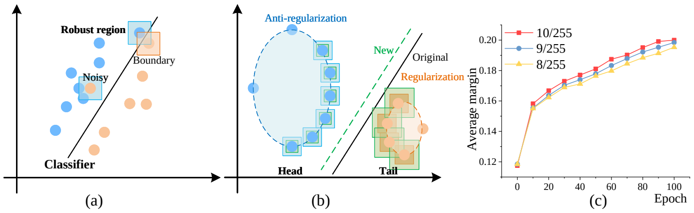


Fig. 8. (a): Illustration for noisy and boundary samples. (b): Illustration for imbalanced class distribution. The blue and orange boxes manifest the robust regions for samples in the blue and orange classes, respectively. The green boxes represent the adjusted robust regions of the two categories when adversaries and anti-adversaries are combined in training. The green line represents the new classification boundary using the combination strategy. (c): The average margins of samples with different perturbation bounds.

$D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{|D|}$, in which the samples are drawn *i.i.d* according to a distribution. The optimal model for D is attained through adversarial training with a perturbation bound of ϵ, whose objective can be expressed as follows:

$$\tilde{\mathcal{L}} \approx \sum\nolimits_{i=1}^{|D|} [\ell(\boldsymbol{x}_i, y_i) + \frac{\epsilon}{2} \left\|\partial_{\boldsymbol{x}_i} \ell(\boldsymbol{x}_i, y_i)\right\|_q]. \tag{42}$$

Let f^* be the optimal model obtained by minimizing $\tilde{\mathcal{L}}$ in Eq. (42). Assuming there is a biased dataset for a real-world task comprises the following four types of data. First, there are data points in D that undergo adversarial perturbation with a bound of ϵ, constituting a ratio of p_1. Second, there are data points in D that undergo adversarial perturbation with a bound of 2ϵ, comprising a ratio of p_2. Third, there are data points from D that remain unaltered, making up a ratio of p_3. Finally, there are data points in D that are subjected to anti-adversarial perturbation with a bound of ϵ, accounting for a ratio of p_4 ($p_1 + p_2 + p_3 + p_4 = 1$). Additionally, the perturbation bounds for samples with ratios of p_1, p_2, p_3, and p_4, are denoted as ϵ_1, ϵ_2, ϵ_3, and ϵ_4, respectively. These perturbation bounds can be positive (for adversarial perturbation) or negative (for anti-adversarial perturbation). Consequently, for the samples in the dataset with a ratio of p_1, their optimization objective can be expressed as follows:

$$\tilde{\mathcal{L}}_1 \approx p_1 \sum\nolimits_{i=1}^{|D|} [\ell(\boldsymbol{x}_i, y_i) + (\frac{\epsilon}{2} + \frac{\epsilon_1}{2}) \left\|\partial_{\boldsymbol{x}_i} \ell(\boldsymbol{x}_i, y_i)\right\|_q]. \tag{43}$$

The training objectives for the remaining three parts can be formulated in a similar manner. For the sake of simplicity in notation, let us denote $\sum_{i=1}^{|D|} \ell(\boldsymbol{x}_i, y_i)$ as R_1 and $\sum_{i=1}^{|D|} \|\partial_{\boldsymbol{x}_i} \ell(\boldsymbol{x}_i, y_i)\|$ as R_2. To obtain the same optimal model (f^*) obtained by Eq. (42), the following relationship is supposed to be held:

$$\begin{aligned} &p_1 R_1 + (\frac{\epsilon}{2} + \frac{\epsilon_1}{2}) p_1 R_2 + p_2 R_1 + (\epsilon + \frac{\epsilon_2}{2}) p_2 R_2 + p_3 R_1 \\ &+ \frac{\epsilon_3}{2} p_3 R_2 + p_4 R_1 + (\frac{\epsilon_4}{2} - \frac{\epsilon}{2}) p_4 R_2 = R_1 + \frac{\epsilon}{2} R_2. \end{aligned} \tag{44}$$

A feasible solution ensuring the hold of Eq. (44) is $\{\epsilon_1 = 0, \epsilon_2 = -\epsilon, \epsilon_3 = \epsilon, \epsilon_4 = 2\epsilon\}$. This solution implies that samples with a ratio of p_1 should remain unperturbed, samples with a ratio of p_2 should undergo anti-adversarial perturbation with a bound of ϵ, samples with a ratio of p_3 should be subjected to adversarial perturbation with a bound of ϵ, and samples with a ratio of p_4 should be adversarially perturbed with a bound of 2ϵ. Hence, it becomes evident that different samples necessitate distinct perturbation directions and bounds. Our approach determines perturbation directions and bounds of samples through meta-learning and reinforcement learning, guided by the training characteristics of samples.

Through our analysis, the benefits of training with varied bounds are summarized as follows. First, as boundary samples are easy to over-regularize, they should be perturbed with small bounds to avoid over-regularization. Alternatively, enforcing large margins on those examples will force the classifier to give up those examples, leading to a distorted decision surface, as shown in Fig. 8(a). Fig. 8(c) demonstrates that samples with smaller margins (boundary samples) do have smaller bounds in real applications. This experiment is conducted by Reinforce-CAAT on CIFAR10 using the PreAct-ResNet18 model. Second, the perturbation bound for each sample controls its regularization strength. It is natural that different samples should have diverse regularization strengths according to their characteristics. Consequently, better model performance can be attained under adversarial training with varied bounds.

Furthermore, the benefits of combining adversaries with anti-adversaries in training are as follows. First, noisy samples should perform anti-regularization as the expansion of their robust region will lead to the false prediction of their surrounding clean samples, as illustrated in Fig. 8(a). Second, anti-regularization (i.e., training with anti-adversaries) benefits fairness among classes. Standard adversarial training makes the classifier more inclined to the hard class, making the classifier more sensitive (insensitive) to the perturbations of samples in the hard (easy) category. Therefore, samples in the hard and easy categories should conduct regularization and anti-regularization, respectively. Accordingly, the robust regions for the samples in the hard and easy classes will be increased and decreased, as illustrated in Fig. 8(b). Thus, anti-regularization helps improve the fairness among classes as the classification boundary is adjusted to a proper position. Additionally, the combination strategy enables the application of diverse regularization directions and strengths tailored to individual samples, facilitating the enhancement of model performance.

The limitations of the proposed anti-adversarial perturbation are also discussed here. Firstly, given that anti-adversarial perturbations simplify samples, training exclusively with anti-adversaries may not result in a well-performing model in some learning scenarios. Therefore, our method advocates for integrating adversaries with anti-adversaries during training to effectively adjust the difficulty distribution of the training data. Second, applying anti-adversarial perturbations to samples may decrease the model's robustness against these samples. Therefore, it should be determined whether to apply adversarial or anti-adversarial perturbations to samples based on certain criteria. For example, our method judges through a network based on the training characteristics of samples.

6 Experiments

Experiments are conducted to verify our theoretical findings and the effectiveness of the proposed two algorithms (Meta-CAAT and Reinforce-CAAT) in improving the accuracy, robustness, and fairness of models. Three typical learning scenarios are considered, including training on standard datasets, imbalance learning, and noisy label learning.

6.1 Experimental Settings

Three benchmark adversarial learning datasets, including CIFAR10 [45], SVHN [15], and ImageNet [46], are adopted. The noisy and imbalanced versions of the CIFAR data [31] are also considered. For CIFAR10 and SVHN datasets, PreAct-ResNet18 [47] and Wide-ResNet28-10 (WRN28-10) [48] are adopted as the backbone classifiers. For ImageNet, the ResNet50 [47] model is utilized. Three popular adversarial training algorithms, namely, PGD [3], TRADES [5], and FRL [11], are compared. A debiasing method [49] is also compared, which is to upweight the loss of the category with the largest robust error in the training data. The results of TRADES and FRL are calculated by using the codes in their official repositories.

The training and testing configurations utilized by Xu et al. [11] are followed. 300 samples in each category with clean labels are selected as the meta dataset, helping us tune the hyperparameters and train the strategy network. Adversarial training is performed both on the PGD attack setting $\epsilon = 8/255$ with cross-entropy loss and on AutoAttack, which combines two new versions of PGD with FAB and Square Attack to form a parameter-free, computationally affordable and user-independent ensemble of complementary attacks. For the PGD attack, the number of iterations in an adversarial attack is set to 10 and 100; the predefined bound is set to $8/255$ for our Meta-CAAT and FRL (ReMargin). As for Reinforce-CAAT, the range of the perturbation bound is set from 3 to 15. All the models are optimized using SGD with the momentum 0.9 and a weight decay 5×10^{-4}. For Meta-CAAT, the values of λ is selected in $\{2/3, 1, 1.5, 6\}$. For Reinforce-CAAT, the values of λ for CIFAR10 and SVHN are 0.005 and 0.1, respectively. During the evaluation, we report each model's average and worst-class natural, boundary, and robust error rates. The calculation of the average and worst-class natural and robust errors are as follows:

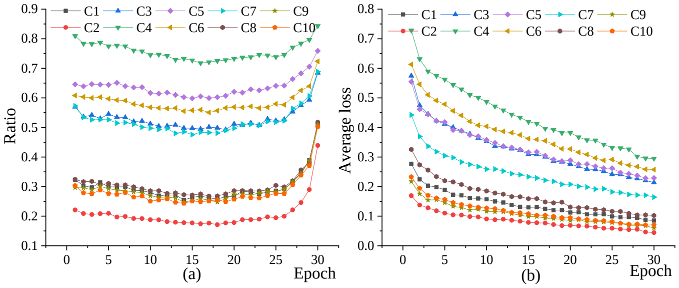


Fig. 9. (a): Ratio of adversaries in each class during training on CIFAR10. (b): Average loss of each class during training on CIFAR10.

$$\overline{R}_{\text{nat}} = \frac{1}{C}\sum\nolimits_{i=1}^{C}\frac{1}{B_i}\sum\nolimits_{j=1}^{B_i}\mathbb{I}(f(\boldsymbol{x}_j) \neq y_j), \tag{45}$$

$$\hat{R}_{\text{nat}} = \max\nolimits_{i=1}^{C}\frac{1}{B_i}\sum\nolimits_{j=1}^{B_i}\mathbb{I}(f(\boldsymbol{x}_j) \neq y_j), \tag{46}$$

$$\overline{R}_{\text{rob}} = \frac{1}{C}\sum_{i=1}^{C}\frac{1}{B_i}\sum_{j=1}^{B_i}\mathbb{I}(\exists\, \|\boldsymbol{\delta}_j\| \leq \epsilon, f(\boldsymbol{x}_j + \boldsymbol{\delta}_j) \neq y_j), \tag{47}$$

$$\hat{R}_{\text{nat}} = \max\nolimits_{i=1}^{C}\frac{1}{B_i}\sum_{j=1}^{B_i}\mathbb{I}(\exists\, \|\boldsymbol{\delta}_j\| \leq \epsilon, f(\boldsymbol{x}_j + \boldsymbol{\delta}_j) \neq y_j), \tag{48}$$

where $\mathbb{I}(\cdot)$ is a sign function, C and B_i refer to the size of the label set and the sample size of class i, respectively.

TABLE 1
Average and worst-class natural, boundary, and robust errors (%) for various algorithms on standard CIFAR10.

Method	Avg. Nat.	Worst Nat.	Avg. Bdy.	Worst Bdy.	Avg. Rob.	Worst Rob.
PGD Adv. Training	15.5	33.8	40.9	55.9	56.4	82.7
TRADES ($1/\lambda = 1$)	14.6	31.2	43.1	64.6	57.7	84.7
TRADES ($1/\lambda = 6$)	19.6	39.1	29.9	49.5	49.5	77.6
Baseline ReWeight	19.2	28.3	39.2	53.7	58.4	80.1
FRL (ReWeight)	16.0	**22.5**	41.6	54.2	57.6	73.3
FRL (ReMargin)	16.9	24.9	35.0	50.6	51.9	75.5
FRL (ReWeight+ReMargin)	17.0	26.8	35.7	44.5	52.7	69.5
Meta-CAAT (Grad-based)	14.6	23.6	**14.4**	**23.3**	29.0	48.1
Meta-CAAT (ReMargin)	13.9	24.3	15.4	24.9	29.3	**44.4**
Reinforce-CAAT	**13.7**	24.6	15.1	25.1	**28.8**	46.8

TABLE 2
Average and worst-class natural, boundary, and robust errors (%) for various algorithms on standard SVHN.

Method	Avg. Nat.	Worst Nat.	Avg. Bdy.	Worst Bdy.	Avg. Rob.	Worst Rob.
PGD Adv. Training	9.4	19.8	37.0	53.9	46.4	73.7
TRADES ($1/\lambda = 1$)	9.9	18.6	39.1	60.6	49.0	78.3
TRADES ($1/\lambda = 6$)	10.5	23.4	32.5	52.5	43.0	76.6
Baseline ReWeight	8.8	17.4	39.3	54.7	48.2	72.1
FRL (ReWeight)	7.9	13.3	38.2	56.4	46.1	69.7
FRL (ReMargin)	9.2	17.4	39.7	49.6	48.9	67.0
FRL (ReWeight+ReMargin)	7.7	12.8	36.2	51.2	43.9	64.0
Meta-CAAT (Grad-based)	6.3	**9.8**	27.2	35.4	33.5	**43.2**
Meta-CAAT (ReMargin)	6.6	10.8	24.7	**32.3**	31.3	43.9
Reinforce-CAAT	**6.2**	9.9	**24.5**	33.7	**30.7**	44.2

TABLE 3
Robust accuracy for AutoAttack on standard CIFAR10.

Method	AutoAttack
PGD Adv. Training	44.29
TRADES ($1/\lambda = 1$)	50.11
TRADES ($1/\lambda = 6$)	50.73
FRL (ReWeight)	51.34
FRL (ReMargin)	51.66
FRL (ReWeight+ReMargin)	52.43
Meta-CAAT (Grad-based)	60.45
Meta-CAAT (ReMargin)	60.76
Reinforce-CAAT	**61.54**

TABLE 4
Robust accuracy for AutoAttack on ImageNet.

Method	AutoAttack
PGD Adv. Training	37.51
TRADES ($1/\lambda = 1$)	40.33
TRADES ($1/\lambda = 6$)	40.18
FRL (ReWeight)	43.05
FRL (ReMargin)	43.12
FRL (ReWeight+ReMargin)	43.21
Meta-CAAT (Grad-based)	**49.87**
Meta-CAAT (ReMargin)	49.21
Reinforce-CAAT	49.69

ϵ is the perturbation bound. $\hat{R}_{\text{nat}}$ and $\overline{R}_{\text{nat}}$ refer to the worst-class and average natural errors, respectively. $\hat{R}_{\text{rob}}$ and $\overline{R}_{\text{rob}}$ refer to the worst-class and average robust errors, respectively. The average (worst-class) boundary error $\overline{R}_{\text{bdy}}$ ($\hat{R}_{\text{bdy}}$) equals the average (worst-class) robust error minus the average (worst-class) natural error, i.e., $\overline{R}_{\text{bdy}} = \overline{R}_{\text{rob}} - \overline{R}_{\text{nat}}$ ($\hat{R}_{\text{bdy}} = \hat{R}_{\text{rob}} - \hat{R}_{\text{nat}}$).

6.2 Experiments on Standard Datasets

The average and worst-class natural and robust errors of PreAct-ResNet18 on standard CIFAR10 and SVHN datasets for PGD with step 10 are shown in Tables 1 and 2, while those of Wide-ResNet28-10 (WRN28-10) and PGD with step 100 are presented in the supplementary material (Tables A-1, A-2, and A-3). As our training and testing configurations are the same as those in [11], the results of the competing methods in the FRL [11] paper are directly presented.

From the results, Meta-CAAT with two types of bound reduces the average natural and robust errors under different degrees, manifesting that it obtains better accuracy and robustness of the model. Compared with other methods, Meta-CAAT decreases the average and worst robust error rates by 21% and 25% on CIFAR10. It decreases the average and worst robust error rates by 13% and 21% on SVHN. In addition, Reinforce-CAAT achieves equivalent or even better performance compared with Meta-CAAT on standard datasets. It attains the lowest average natural and robust errors on CIFAR10 and has the minimum average natural, boundary, and robust errors on SVHN. Baseline ReWeight only decreases the worst intraclass natural error but cannot equalize boundary or robust errors. FRL [11] has limited ability to reduce the worst-class boundary and robust errors, leading to limited fairness among classes. Both algorithms of CAAT more effectively decrease the worst intraclass errors. Thus, it achieves better fairness among classes than other methods. Although FRL (ReWeight) obtains the lowest worst-class natural error, it has large average and worst robust errors, which is inferior to CAAT. Hard classes (classes with large average losses) have higher ratios of adversaries than easy ones during training, as illustrated in Figs. 9(a) and (b), helping improve the model performance on hard classes and enhance the fairness among classes. The same conclusions are also obtained on the SVHN dataset.

The robust accuracy on standard CIFAR10 and ImageNet datasets for AutoAttack are shown in Tables 3 and 4. From

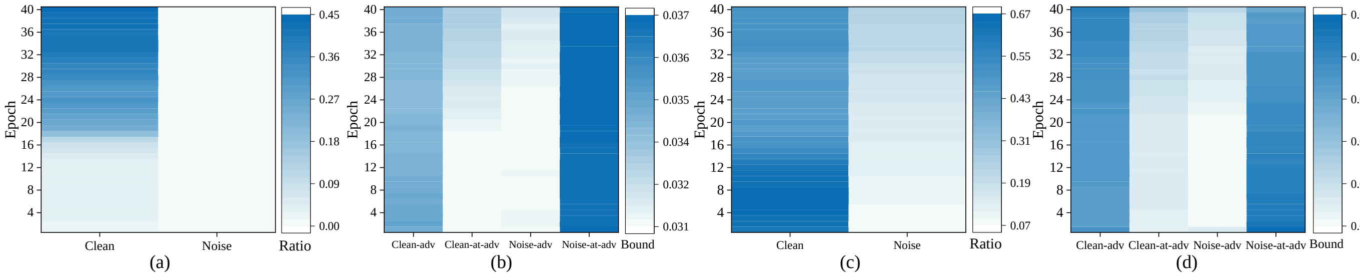


Fig. 10. (a): Ratio of adversaries for noisy and clean samples on CIFAR10 with 20% uniform noise during training. (b): Average adversarial and anti-adversarial perturbation bounds for clean and noisy samples on CIFAR10 with 20% uniform noise during training. (c): Ratio of adversaries for noisy and clean samples on CIFAR10 with 40% pair-flip noise during training. (d): Average adversarial and anti-adversarial perturbation bounds for clean and noisy samples on CIFAR10 with 40% pair-flip noise during training.

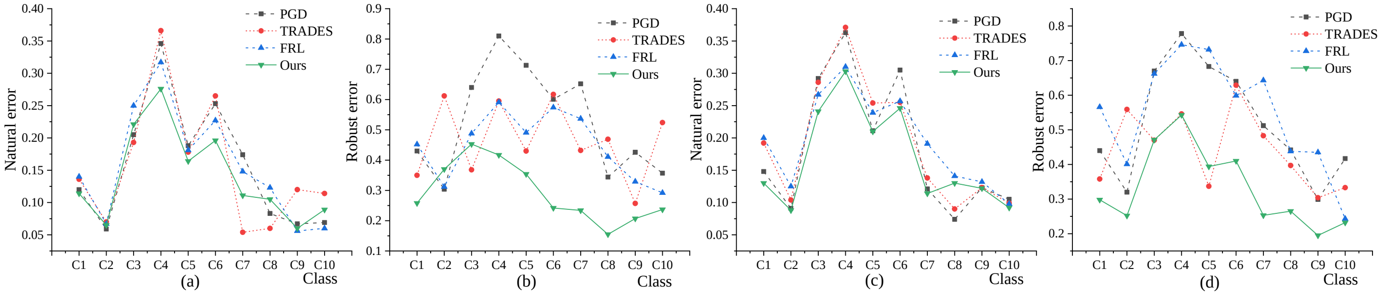


Fig. 11. (a) and (b): Natural and robust errors of four methods for each class on CIFAR10 with 20% pair-flip noise. (c) and (d): Natural and robust errors of four methods for each class on CIFAR10 with 40% pair-flip noise.

TABLE 5
Average and worst-class natural, boundary, and robust errors (%) for various algorithms on CIFAR10 with 20% pair-flip noise.

Method	Avg. Nat.	Worst Nat.	Avg. Bdy.	Worst Bdy.	Avg. Rob.	Worst Rob.
PGD Adv. Training	15.6	34.6	37.1	52.5	52.8	81.0
TRADES ($1/\lambda = 1$)	15.6	36.6	31.0	54.2	46.5	61.7
TRADES ($1/\lambda = 6$)	16.4	34.2	21.0	38.9	37.4	59.9
FRL (ReWeight)	15.3	31.6	36.0	50.7	51.4	61.4
FRL (ReMargin)	15.2	31.3	36.0	51.9	51.1	79.1
FRL (ReWeight+ReMargin)	15.7	31.7	34.3	48.2	50.0	59.1
Meta-CAAT (Grad-based)	**14.6**	**25.2**	**13.9**	24.5	**28.5**	**45.4**
Meta-CAAT (ReMargin)	14.7	30.6	14.7	**24.0**	29.4	52.3
Reinforce-CAAT	14.9	29.8	15.2	26.7	30.1	52.8

TABLE 6
Average and worst-class natural, boundary, and robust errors (%) for various algorithms on CIFAR10 with 40% uniform noise.

Method	Avg. Nat.	Worst Nat.	Avg. Bdy.	Worst Bdy.	Avg. Rob.	Worst Rob.
PGD Adv. Training	17.8	35.3	32.9	51.3	50.7	74.0
TRADES($1/\lambda = 1$)	18.6	35.3	33.9	56.8	52.5	72.1
TRADES($1/\lambda = 6$)	23.6	38.8	28.7	55.8	52.3	69.7
FRL(ReWeight)	15.5	29.5	34.8	49.7	50.3	75.5
FRL(ReMargin)	15.7	32.1	33.3	50.0	49.0	75.8
FRL(ReWeight+ReMargin)	15.9	29.9	33.2	50.4	49.1	72.9
Meta-CAAT (Grad-based)	**13.6**	**28.6**	**19.3**	**31.3**	**32.9**	**57.9**
Meta-CAAT (ReMargin)	14.1	28.9	19.8	31.8	33.9	60.4
Reinforce-CAAT	15.3	28.9	19.4	32.0	34.7	59.2

TABLE 7
Average and worst-class natural, boundary, and robust errors (%) on CIFAR10 with an imbalance factor of 10.

Method	Avg. Nat.	Worst Nat.	Avg. Bdy.	Worst Bdy.	Avg. Rob.	Worst Rob.
PGD Adv. Training	20.1	49.4	42.8	49.6	62.9	93.0
TRADES ($1/\lambda = 1$)	16.8	40.5	32.3	57.7	49.1	75.5
TRADES ($1/\lambda = 6$)	23.6	54.0	23.8	46.9	47.4	79.0
FRL (ReWeight)	16.9	35.5	38.1	49.1	55.0	72.0
FRL (ReMargin)	17.5	49.8	35.6	51.7	53.1	88.8
FRL (ReWeight+ReMargin)	17.2	39.6	35.1	51.2	52.3	72.2
Meta-CAAT (Grad-based)	**15.8**	34.3	14.2	24.3	30.0	58.6
Meta-CAAT (ReMargin)	16.2	**34.2**	**13.7**	**23.2**	**29.9**	**57.4**
Reinforce-CAAT	16.3	34.6	15.9	25.1	32.2	58.8

the results, Meta-CAAT with two types of bound increases the robust accuracies under different degrees. Compared with other methods, Meta-CAAT increases the robust accuracy by 9% on CIFAR10. It increases the robust error accuracy by 6% on ImageNet. In addition, Reinforce-CAAT achieves equivalent or even better performance compared with Meta-CAAT on these two datasets. It attains the highest robust accuracy on CIFAR10 and has the second-best robust accuracy on ImageNet.

6.3 Experiments of Noisy Classifications

Two types of corrupted labels, including uniform and pair-flip noises, are adopted [31]. Uniform noise means that the label of each sample is independently changed to a random category, while flip noise means that the label of each sample is independently flipped to two similar categories. In the experiments, we randomly select two categories as similar categories. The noise ratios are set to 20% and 40%. The CIFAR10 dataset, which is popularly used for the evaluation of noisy labels, is adopted. The results of PreAct-ResNet18 [47] on CIFAR10 with 20% pair-flip and 40% uniform noises for PGD with step 10 are reported in Tables 5 and 6, while others are presented in the supplementary material (Tables A-4 and A-5). From the results, combining adversaries and anti-adversaries in training (i.e., Meta-CAAT and Reinforce-CAAT) achieves the lowest average and worst natural and robust errors, indicating that the best generalization, robustness, and fairness are obtained compared with other approaches. As Meta-CAAT utilizes an additional clean dataset to optimize the parameters in each epoch, its performance is generally better than that of Reinforce-CAAT on noisy datasets.

As illustrated in Figs. 10(a) and (c), almost all noisy samples are anti-adversarially perturbed during training, which is in accordance with our theoretical findings. The ratios of adversaries for clean samples increase as training progresses, indicating that clean samples play an increasingly important role during training. From Figs. 10(b) and (d), the average anti-adversarial perturbation bound for noisy samples is the largest, implying that noisy samples exhibit the largest degree of anti-adversarial perturbation. Thus, the negative influence of noisy samples can be deduced.

TABLE 8
Average and worst-class natural, boundary, and robust errors (%) for various algorithms on CIFAR10 with an imbalance factor of 100.

Method	Avg. Nat.	Worst Nat.	Avg. Bdy.	Worst Bdy.	Avg. Rob.	Worst Rob.
PGD Adv. Training	24.5	74.8	38.0	47.5	62.5	96.2
TRADES($1/\lambda = 1$)	30.8	68.3	29.9	64.2	60.7	83.8
TRADES($1/\lambda = 6$)	39.2	83.2	**16.7**	34.3	55.9	86.1
FRL(ReWeight)	19.8	42.6	36.9	50.8	56.7	86.9
FRL(ReMargin)	23.2	69.6	34.2	43.8	57.4	94.3
FRL(ReWeight+ReMargin)	19.2	48.3	36.8	52.5	56.0	89.4
Meta-CAAT (Grad-based)	18.8	**39.3**	16.8	27.5	**35.6**	**66.8**
Meta-CAAT (ReMargin)	**18.7**	41.5	17.6	**25.8**	36.3	72.9
Reinforce-CAAT	19.1	41.2	17.5	27.6	36.6	71.5

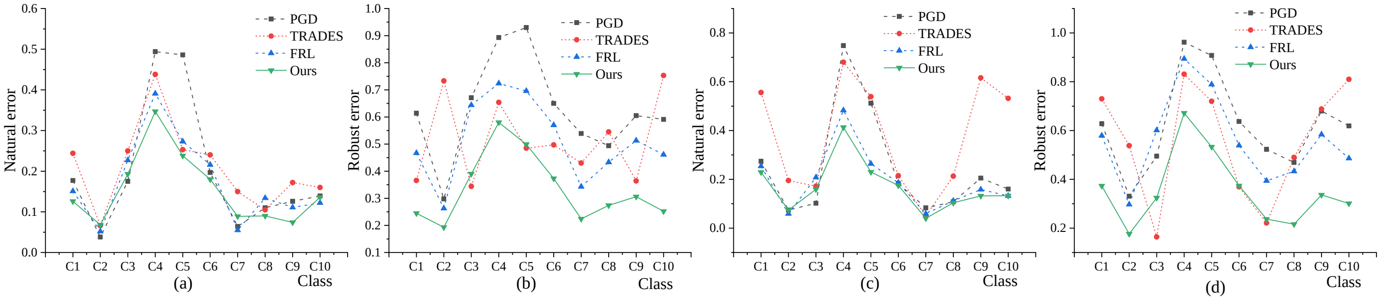


Fig. 12. (a) and (b): Natural and robust errors of four methods for each class on CIFAR10 with an imbalance factor of 10. (c) and (d): Natural and robust errors of four methods for each class on CIFAR10 with an imbalance factor of 100.

In addition, the average adversarial perturbation bound for clean samples is also large in training, manifesting that clean samples are adversarially perturbed to a high degree. Figs. 11(a)-(d) exhibit the natural and robust errors of each class on CIFAR10 with 20% and 40% pair-flip noise. Those with 20% and 40% uniform noise are shown in the supplementary material (Fig. A-19). Our method decreases the robust and natural errors of most categories. In addition, the gaps between the largest and the smallest errors are also narrowed, manifesting that the combination strategy achieves the best fairness compared with other methods.

6.4 Experiments of Imbalanced Classifications

The long-tailed version of CIFAR10 compiled by Cui et al. [42] is used. The values of the imbalance factor are set to 10 and 100. The results of PreAct-ResNet18 [47] for PGD with step 10 are presented in Tables 7 and 8. Combining adversaries with anti-adversaries in training with varied perturbation bounds achieves the minimum average and worst-class natural and robust errors compared with other algorithms. Compared with Reinforce-CAAT, Meta-CAAT utilizes an additional balanced meta dataset to optimize the parameters in each epoch, which obtains state-of-the-art performance. As shown in Figs. 12(a)-(d), CAAT decreases the natural and robust errors for most classes and achieves the smallest performance gap among different classes. Figs. 13(a) and (b) illustrate the ratio of adversaries and the average perturbation bound of each class during the training process. The first head class with the largest number of samples has the lowest ratio of adversaries, while the tail classes have high proportions of adversaries, which is in accordance with our theoretical analysis. In addition, the head and tail categories have large average perturbation bounds, indicating that the head classes exhibit a great degree of anti-adversarial perturbation, and the tail classes conduct a large extent of adversarial perturbation.

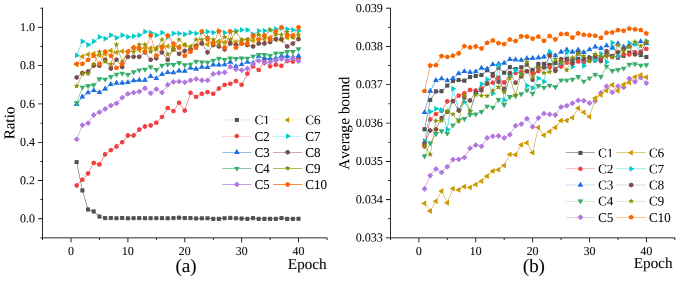


Fig. 13. (a): Ratio of adversaries in each class during training on CIFAR10 with an imbalance factor of 100. (b): Average bound of each class during training on CIFAR10 with an imbalance factor of 100. “C1” to “C10” is from the first head category to the last tail category.

6.5 Ablation Studies

This section conducts ablation studies for our CAAT. All experiments are conducted on PGD with step 10.

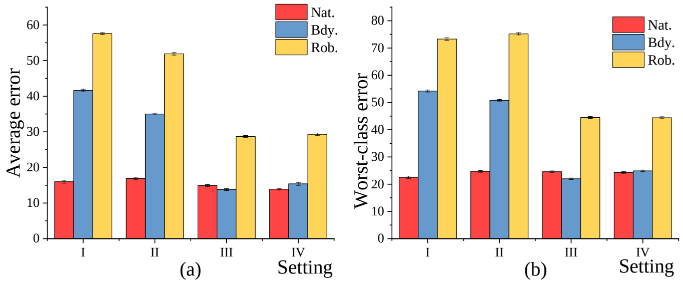


Fig. 14. Average (a) and worst-class (b) natural, boundary, and robust errors (%) for four variations of CAAT on standard CIFAR10.

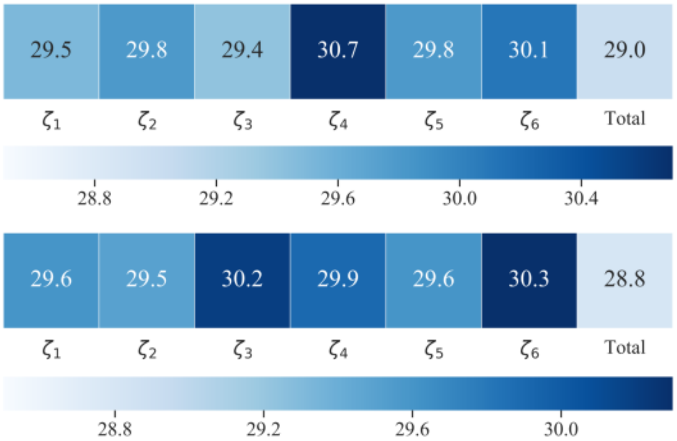


Fig. 15. Ablation studies for the six training characteristics of Meta-CAAT (top) and Reinforce-CAAT (bottom) on standard CIFAR10.

First, to validate that both perturbation directions and bounds are crucial, four variations of CAAT are considered, including adversarial training with the same perturbation direction and bound (Setting I), adversarial training with the same perturbation direction and different bounds (Setting II), adversarial training with different perturbation directions (adversaries and anti-adversaries) and the same bound (Setting III), and adversarial training with different perturbation directions and bounds (Setting IV). Meta-CAAT is adopted to realize the four variations. The PreAct-ResNet18 model is utilized as the backbone classifier. The results are presented in Figs. 14(a) and (b). Settings III and IV achieve lower average and worst-class natural and robust errors than Settings I and II. Thus, the combination strategy is more effective than using only adversaries. Compared with Setting III, Setting IV further decreases the average natural error. In addition, the average boundary and robust errors are reduced by Setting II compared with Setting I. Therefore, the varied bound is more valid in some cases.

Second, we evaluate the necessity of six adopted training characteristics, which include loss ($\zeta_{\boldsymbol{x},1}$), margin ($\zeta_{\boldsymbol{x},2}$), loss gradient ($\zeta_{\boldsymbol{x},3}$), information entropy ($\zeta_{\boldsymbol{x},4}$), class proportion ($\zeta_{\boldsymbol{x},5}$), and the average loss for each class ($\zeta_{\boldsymbol{x},6}$). The results are depicted in Fig. 15, indicating that the exclusion of any of these characteristics leads to a decrease in model performance. Consequently, all these characteristics are deemed both useful and necessary.

Third, we assess the necessity of employing meta- and reinforcement learning in determining the perturbation directions and bounds for samples within the context of imbalance learning. In the absence of both meta- and reinforcement learning, samples in tail categories are adversarially perturbed, while those in head categories are anti-adversarially perturbed to enhance the model's performance on tail categories. The perturbation bound is set at $8/255$. The results are presented in Table 9. As evident, the model's performance is notably inferior when compared to the utilization of meta-learning or reinforcement learning. This indicates that employing meta-learning and reinforcement learning to determine the perturbed directions and bounds of samples based on their diverse training characteristics is a more reasonable approach compared to determining them artificially.

TABLE 9
Average natural, boundary, and robust errors (%) on CIFAR10 with an imbalance factor of 10.

Method	Avg. Nat.	Avg. Bdy.	Avg. Rob.
Meta-CAAT (Grad-based)	**15.8**	14.2	30.0
Meta-CAAT (ReMargin)	16.2	**13.7**	**29.9**
Reinforce-CAAT	16.3	15.9	32.2
Without	20.5	20.1	40.6

7 Conclusions

This study theoretically investigates the role of adversarial training with different directions (adversarial and anti-adversarial) and bounds for the robust model. Four typical learning occasions are considered, including classes with different difficulties, imbalance learning, noisy label learning, and classes with skewed distributions. A series of theoretical findings are obtained, illuminating a new objective that combines adversaries and anti-adversaries (CAAT) in training with varied perturbation bounds. Consequently, two novel adversarial training frameworks (Meta-CAAT and Reinforce-CAAT), which are based on meta-learning and reinforcement learning, respectively, are proposed to solve the objective, in which the perturbation directions and bounds are determined by the training characteristics of samples. The role of the combination strategy with varied bounds is further explained from a regularization aspect. Extensive experiments verify the rationality of our theoretical findings and the effectiveness of the proposed adversarial training frameworks in achieving better accuracy, robustness, and fairness of the robust models compared with other adversarial training methods.

References

[1] S. Lee, H. Kim, and J. Lee, "Graddiv: Adversarial robustness of randomized neural networks via gradient diversity regularization," *TPAMI*, vol. 45, no. 2, pp. 2645–2651, 2023.

[2] T. Bai and J. Luo, "Recent advances in adversarial training for adversarial robustness," in *IJCAI*, 2021, pp. 4312–4321.

[3] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018, pp. 1–28.

[4] X. Mao, Y. Chen, S. Wang, H. Su, Y. He, and H. Xue, "Composite adversarial attacks," in *AAAI*, 2021, pp. 8884–8892.

[5] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *ICML*, 2019, pp. 12 907–12 929.

[6] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, "Improving adversarial robustness requires revisiting misclassified examples," in *ICLR*, 2020, pp. 1–14.

[7] Y.-Y. Yang, C. Rashtchian, H. Zhang, R. Salakhutdinov, and K. Chaudhuri, "A closer look at accuracy vs. robustness," in *NeurIPS*, 2020, pp. 8588–8601.

[8] C. Xie, M. Tan, B. Gong, A. Yuille, and Q. V. Le, "Smooth adversarial training," *arXiv preprint arXiv:2006.14536*, 2020.

[9] L. Rice, E. Wong, and J. Z. Kolter, "Overfitting in adversarially robust deep learning," in *ICML*, 2020, pp. 8093–8104.

[10] Y. Dong, K. Xu, X. Yang, T. Pang, Z. Deng, H. Su, and J. Zhu, "Exploring memorization in adversarial training," *arXiv preprint arXiv:2106.01606*, 2021.

[11] H. Xu, X. Liu, Y. Li, A. K. Jain, and J. Tang, "To be robust or to be fair: Towards fairness in adversarial training," in *ICML*, 2021, pp. 11 492–11 501.

[12] J. Uesato, J.-B. Alayrac, P.-S. Huang, R. Stanforth, A. Fawzi, and P. Kohli, "Are labels required for improving adversarial robustness?" in *NeurIPS*, 2019, pp. 12 214–12 223.

[13] J. Zhu, J. Zhang, B. Han, T. Liu, G. Niu, H. Yang, M. Kankanhalli, and M. Sugiyama, "Understanding the interaction of adversarial training with noisy labels," *arXiv preprint arXiv:2102.03482*, 2021.

[14] M. Alfarra, J. C. Pérez, A. Thabet, A. Bibi, P. H. S. Torr, and B. Ghanem, "Combating adversaries with anti-adversaries," in *AAAI*, 2022, pp. 5992–6000.

[15] Y. Netzer, T. Wang, A. Coates, R. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," *Scandinavian Journal of Statistics*, 2011.

[16] M. Cheng, Q. Lei, P.-Y. Chen, I. Dhillon, and C.-J. Hsieh, "Cat: Customized adversarial training for improved robustness," *arXiv preprint arXiv:2002.06789*, 2020.

[17] Y. Balaji, T. Goldstein, and J. Hoffman, "Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets," *arXiv preprint arXiv:1910.08051*, 2019.

[18] G. W. Ding, Y. Sharma, K. Y. C. Lui, and R. Huang, "MMA training: Direct input space margin maximization through adversarial training," in *ICLR*, 2020, pp. 1–28.

[19] S. Yang, T. Guo, Y. Wang, and C. Xu, "Adversarial robustness through disentangled representations," in *AAAI*, 2021, pp. 3145–3153.

[20] A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, and P. Liang, "Adversarial training can hurt generalization," *arXiv preprint arXiv:1906.06032*, 2019.

[21] J. Zhang, X. Xu, B. Han, G. Niu, L. Cui, M. Sugiyama, and M. Kankanhalli, "Attacks which do not kill training make adversarial learning stronger," in *ICML*, 2020, pp. 11 258–11 287.

[22] C. Song, K. He, J. Lin, L. Wang, and J. E. Hopcroft, "Robust local features for improving the generalization of adversarial training," in *ICLR*, 2020, pp. 1–12.

[23] T. Pang, X. Yang, Y. Dong, H. Su, and J. Zhu, "Bag of tricks for adversarial training," in *ICLR*, 2021, pp. 1–21.

[24] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *TPAMI*, vol. 44, no. 9, pp. 5149–5169, 2021.

[25] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few shot learning," in *NeurIPS*, 2017, pp. 4078–4088.

[26] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *CVPR*, 2018, pp. 1199–1208.

[27] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and D. Wierstra, "Meta learning with memory-augmented neural networks," in *ICML*, 2016, pp. 2740–2751.

[28] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017, pp. 1856–1868.

[29] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

[30] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *ICML*, 2018, pp. 6900–6909.

[31] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, and D. Meng, "Meta-weight-net: Learning an explicit mapping for sample weighting," in *NeurIPS*, 2019, pp. 1917–1928.

[32] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[33] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *JAIR*, vol. 4, pp. 237–285, 1996.

[34] P. Ladosz, L. Weng, M. Kim, and H. Oh, "Exploration in deep reinforcement learning: A survey," *Information Fusion*, 2022.

[35] T. Xu, Z. Li, and Y. Yu, "Error bounds of imitating policies and environments for reinforcement learning," *TPAMI*, vol. 44, no. 10, pp. 6968–6980, 2021.

[36] X. Jia, Y. Zhang, B. Wu, K. Ma, J. Wang, and X. Cao, "Las-at: Adversarial training with learnable attack strategy," in *CVPR*, 2022, pp. 13 398–13 408.

[37] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[38] A. Azzalini, "A class of distributions which includes the normal ones," *Scandinavian Journal of Statistics*, vol. 12, pp. 171–178, 1985.

[39] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. Kankanhalli, "Geometry-aware instance-reweighted adversarial training," in *ICLR*, 2021, pp. 1–29.

[40] C. Santiago, C. Barata, M. Sasdelli, G. Carneiro, and J. C.Nascimentoa, "LOW: Training deep neural networks by learning optimal sample weights," *Pattern Recognition*, vol. 110, pp. 130–141, 2021.

[41] Q. A. Wang, "Probability distribution and entropy as a measure of uncertainty," *Journal of Physics A: Mathematical and Theoretical*, vol. 41, no. 6, p. 65004, 2008.

[42] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *CVPR*, 2019, pp. 9260–9270.

[43] Z. Zhang and T. Pfister, "Learning fast sample re-weighting without reward data," in *ICCV*, 2021, pp. 705–714.

[44] C.-J. Simon-Gabriel, Y. Ollivier, L. Bottou, B. Schölkopf, and D. Lopez-Paz, "First-order adversarial vulnerability of neural networks and input dimension," in *ICML*, 2019, pp. 5809–5817.

[45] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Scandinavian Journal of Statistics*, 2009.

[46] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, , and M. Bernstein, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, pp. 211–252, 2015.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[48] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *BMVC*, 2016.

[49] A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. Wallach, "A reductions approach to fair classification," in *ICML*, 2018, pp. 60–69.

Xiaoling Zhou received the B.Sc. degree in Mathematics from Tiangong University, Tianjin, China, in 2020, and the M.Sc. degree in Mathematics from the Center for Applied Mathematics, Tianjin University, Tianjin, China, in 2023. She is currently pursuing the Ph.D. degree with the National Engineering Research Center for Software Engineering, Peking University, Beijing.

Her research interests include adversarial training and meta-learning.

Ou Wu received the B.Sc. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2003, and the M.Sc. and Ph.D. degrees in computer science from the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2006 and 2012, respectively.

In 2007, he joined NLPR as an Assistant Professor. In 2017, he joined the Center for Applied Mathematics, Tianjin University, Tianjin, China, as a Full Professor. His research interests include data mining and machine learning.

Nan Yang received the B.Sc. degree in Mathematics from China University of Petroleum, Qingdao, China, in 2021. She is currently pursuing a master's degree in Mathematics with the Center for Applied Mathematics, Tianjin University, Tianjin, under the supervision of Professor Ou Wu.

Her research interests include meta-learning and imbalance learning.