Combining Adversaries with Anti-adversaries in Training

Xiaoling Zhou, Nan Yang, Ou Wu *

Center for Applied Mathematics, Tianjin University, China
{xiaolingzhou, yny, wuou}@tju.edu.cn

Abstract

Adversarial training is an effective learning technique to improve the robustness of deep neural networks. In this study, the influence of adversarial training on deep learning models in terms of fairness, robustness, and generalization is theoretically investigated under more general perturbation scope that different samples can have different perturbation directions (the adversarial and anti-adversarial directions) and varied perturbation bounds. Our theoretical explorations suggest that the combination of adversaries and anti-adversaries (samples with anti-adversarial perturbations) in training can be more effective in achieving better fairness between classes and a better tradeoff between robustness and generalization in some typical learning scenarios (e.g., noisy label learning and imbalance learning) compared with standard adversarial training. On the basis of our theoretical findings, a more general learning objective that combines adversaries and anti-adversaries with varied bounds on each training sample is presented. Meta learning is utilized to optimize the combination weights. Experiments on benchmark datasets under different learning scenarios verify our theoretical findings and the effectiveness of the proposed methodology.

Introduction

Apart from the standard generalization error (also known as natural error), robust generalization error (also known as robust error) has received great attention in recent years. A deep neural network with a low robust error can cope well with adversarial attacks. Adversarial training is an effective technique to reduce the robust error of a model (Wong, Rice, and Kolter 2020; Bai and Luo 2021). Given a model $f(\cdot)$ and a sample $\boldsymbol{x}$ associated with a label y, classical adversarial training methods (Madry et al. 2018; Goodfellow, Shlens, and Szegedy 2014) 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, $\boldsymbol{\delta}$ is the perturbation term, and ϵ is the perturbation bound. Adversaries are then leveraged as the training data to learn a more robust model. A number of variations for adversarial training have been proposed in recent literature. Zhang et al. (2019) decomposed the robust error into the natural and boundary errors. They developed a new method, namely, TRADES, to obtain a better tradeoff between standard generalization and robustness. Wang et al. (2020) proposed a misclassification-aware adversarial training method to focus on the misclassified examples.

In addition to the design of new methods, theoretical studies have been conducted to explore the effectiveness and ineffectiveness of adversarial training (Bai and Luo 2021). Yang et al. (2020) concluded that existing adversarial methods cannot achieve an ideal tradeoff between accuracy and robustness due to the insufficient smoothness (Xie et al. 2020) and generalization properties of classifiers trained by these methods. They pointed out that customized optimization methods or better network architectures should be proposed. Xu et al. (2021) revealed that adversarial training introduces severe unfairness between different categories. Thus, they developed a new method that sets varied perturbation bounds for each class, resulting in better fairness. Different from these studies, we conjectured that one possible reason leading to unsatisfied tradeoff and fairness is that not all training samples should be perturbed adversarially. For instance, adversaries of noisy samples may harm the model performance (Uesato et al. 2019), and these samples should be perturbed in the anti-adversarial direction to reduce their negative influence on model optimization. Zhu et al. (2021) 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] ($\boldsymbol{x}_{\text{at-adv}}$)

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

This study conducts a comprehensive theoretical analysis of adversarial training in the presence of 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, and noisy label learning. Our theoretical findings reveal that the perturbation directions and

*Corresponding author.

[1]The anti-adversary defined by Alfarra et al. (2022) is different from ours. They utilize anti-adversaries to deal with attacks, whereas we aim to improve robustness, accuracy, and fairness.

bounds can remarkably influence the model training. The combination of adversaries and anti-adversaries with varied bounds can improve the fairness among classes and achieve a better tradeoff between accuracy and robustness. Accordingly, a general objective that combines adversaries and anti-adversaries is constructed for adversarial training. A meta learning-based method is then proposed to optimize this objective, in which the perturbation direction and bound of each training sample is adjusted in accordance with its learning characteristics during training. Our experimental results show that the combining strategy outperforms state-of-the-art adversarial training methods. 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. A comprehensive theoretical analysis is conducted for the role of the combination strategy with varied perturbation bounds[2] under three typical learning scenarios.
- A new objective is established for adversarial training by combining adversaries and anti-adversaries. Meta learning is utilized to solve the optimization, and the perturbation direction and bound for each training sample are determined in accordance with its learning characteristics, such as training loss and margin.

Related Work

Tradeoff and Fairness in Adversarial Training

Recent studies on adversarial training focus on the tradeoff between accuracy and robustness. Efforts (Raghunathan et al. 2019; Zhang et al. 2019, 2020; Yang et al. 2021) 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 (Song et al. 2020). Rice et al. (2020) systematically investigated the role of various 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. (2021). Unfairness is also a problem caused by adversarial training. Xu et al. (2021) trained a robust classifier to minimize error and stressed it to satisfy two fairness constraints. Several studies (Ding et al. 2020; Cheng et al. 2020; Balaji, Goldstein, and Hoffman 2019) adaptively tune the perturbation bounds for each sample with the inspiration that samples near the decision boundary should have small bounds.

Meta Learning

Meta learning has aroused great interest in recent years. Existing meta learning methods can be divided into three categories, namely, metric-based (Snell, Swersky, and Zemel 2017; Sung et al. 2018), model-based (Santoro et al. 2016), and optimizing-based (Finn, Abbeel, and Levine 2017; Nichol, Achiam, and Schulman 2018) methods. The algorithm we adopted that is inspired by Model-Agnostic Meta-Learning (Finn, Abbeel, and Levine 2017) belongs to the optimizing-based methods. The data-driven manner of meta optimization is always utilized to learn the sample weights or the hyperparameters (Ren et al. 2018; Shu et al. 2019).

[2]Existing theoretical studies presume that the perturbation bounds are identical for all training samples.

Theoretical Investigation

This section conducts theoretical analyses to assess the influence of two different perturbation directions and varied bounds on adversarial training in three typical binary classification cases. Proofs are presented in the online material.

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 is a mapping from the input data space $\mathcal{X}$ to the label space $\mathcal{Y}$. It can be parametrized 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 $\mathcal{R}_{\text{rob}}(f) := \Pr(\exists \boldsymbol{\delta} \ ||\boldsymbol{\delta}|| \leq \epsilon, \text{s.t.} f(\boldsymbol{x} + \boldsymbol{\delta}) \neq y)$.

Case I: Classes with Different Difficulties

In this case, the binary setting established by Xu et al. (2021) 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 $K > 1$. The data follow

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

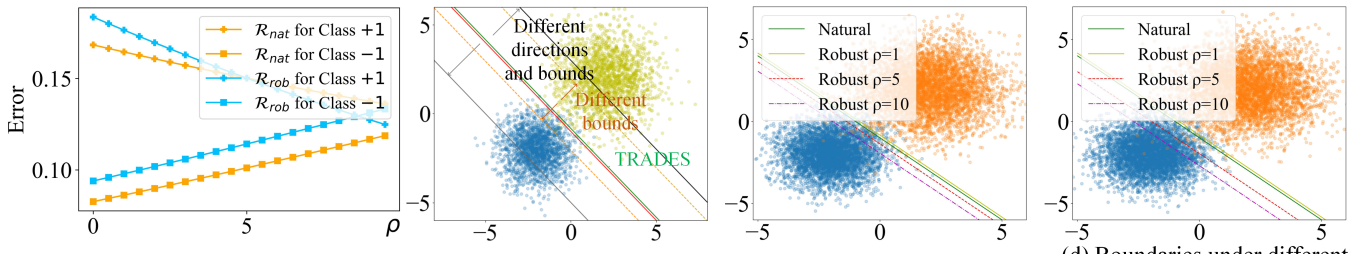
Class "$+1$" is harder because the optimal linear classifier will give a larger error to class "$+1$" than class "-1". Xu et al. (2021) proved that adversarial training with an equal bound will exacerbate the performance gap (including natural and robust errors) between classes and hurt the harder class. We show that adversarial training with unequal bounds on two classes can tune the performance gap and the tradeoff between the robustness and accuracy of the model. Let $\sigma_{-1} = \sigma$. The following theorem is first proposed.

Theorem 1 *For a data distribution $\mathcal{D}$ in Eq. (3), assume that the perturbation bounds of class "-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 two classes are*

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

where $B = \frac{2}{K^2-1}\frac{\sqrt{d}(\eta - \frac{\epsilon(1+\rho)}{2})}{\sigma}$, *and* $q(K) = \frac{2\log K}{K^2-1}$.

The robust errors are shown in the online material. The natural and robust errors change with different ρ values. A corollary is derived in accordance with Theorem 1.



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

Figure 1: (a) Variation of performance gaps between classes as ρ increases. (b) Scope of the classification boundary of different manners. The values of 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. $\rho_+(\rho_-) < 0$ denotes that class "+1(−1)" is anti-adversarially perturbed. The online material provides the formulas of boundaries. (c) Logistic regression classifier boundaries (natural and robust) on simulated data in Eq. (3). (d) Logistic regression classifier boundaries (natural and robust with different directions).

Corollary 1 *The data and perturbations in Theorem 1 are followed. When $K < \exp(d(\eta-\epsilon)^2/2\sigma^2)$, the adversarially trained model will increase and decrease the natural and robust errors of class "−1" and class "+1", with the increase in ρ, respectively.*

Accordingly, the performance gaps of $\mathcal{R}_{\text{nat}}$ and $\mathcal{R}_{\text{rob}}$ decrease with the increase in ρ, and better fairness can be achieved, as shown in Fig. 1 (a). In Fig. 1 (c), the boundary shifts toward the easy class "−1". From Fig. 1 (b), adversarial training with varied bounds contributes to larger scope of the boundary compared with TRADES (Zhang et al. 2019). Thus, a better tradeoff can be attained. Therefore, fairness and tradeoff can be tuned with different ρ values. Next, anti-adversaries are considered. Assume that samples in class "−1" perform anti-adversarial perturbation. Similar to Theorem 1, a theorem calculating the natural and robust errors is proposed as shown in the online material. A corollary is then derived, indicating that the combination of adversaries and anti-adversaries can tune the performance gap and tradeoff.

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 $\rho \times \epsilon$ ($0 \le \epsilon, \rho\epsilon < \eta$). When $K < \exp(d(\eta-\epsilon)^2/2\sigma^2)$, the adversarially trained model will increase and decrease the natural and robust errors of class "−1" and class "+1", with the increase in ρ, respectively.*

In accordance with Corollaries 1 and 2, the adversarial training and the combination strategy can nearly attain the same performance. However, the combination strategy can contribute to the largest scope of the boundary, as shown in Fig. 1 (b). Thus, the combination strategy is more effective in achieving a better tradeoff and fairness theoretically. As shown in Figs. 1 (c) and (d), the combination strategy has a more pronounced effect under the same bound (i.e., the same ρ), indicating that it needs smaller bounds when the same performance is achieved. Thus, the combination strategy is more efficient than only the adversarial perturbation, indicating that anti-adversaries are valuable.

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 $V > 1$.

Class "−1" is the majority category, and an optimal linear classifier will give a smaller natural error for class "−1" than class "+1", as proved in the online material. Similarly, we proved that standard adversarial training will exacerbate the performance gap between classes and hurt the smaller class. We then show that adversarial training with unequal bounds on the two classes will tune the performance gap between classes and the tradeoff between robustness and accuracy. The following theorem is first proposed.

Theorem 2 *For a data distribution $\mathcal{D}_V$ described above with the imbalance factor V, assume that the perturbation bounds of classes "−1" and "+1" are ϵ and $\rho \times \epsilon$ ($0 \le \epsilon, \rho\epsilon < \eta$), respectively. The natural errors of the optimal robust linear classifier f_{rob} for the two classes are*

$$\begin{aligned}\mathcal{R}_{nat}(f_{rob}, -1) &= \Pr\left\{\mathcal{N}(0,1) \le -A - \frac{\log V}{2A} - \frac{\sqrt{d}}{\sigma}\epsilon\right\},\\ \mathcal{R}_{nat}(f_{rob}, +1) &= \Pr\left\{\mathcal{N}(0,1) \le -A + \frac{\log V}{2A} - \frac{\sqrt{d}\rho}{\sigma}\epsilon\right\},\end{aligned} \tag{5}$$

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

A corollary is derived on the basis of Theorem 2.

Corollary 3 *The data and perturbations in Theorem 2 are followed. When $V < \exp(d(\eta-\epsilon)^2/2\sigma^2)$, the adversarially trained model will increase and decrease the natural and robust errors of class "−1" and class "+1", with the increase in ρ, respectively.*

From Corollary 3, the performance gaps between classes can be decreased with different ρ values. The boundary can be moved within the scope with different ρ values that covers the boundary of standard adversarial training. Therefore, a better tradeoff can be attained by adversarial training

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

with varied bounds. Next, the anti-adversaries are considered. We assume that samples in class "-1" perform anti-adversarial perturbation. Similar to Theorem 2, a theorem that portrays the training occasion where the adversaries and anti-adversaries are combined is proposed, as shown in the online material. A corollary is then derived.

Corollary 4 *For a data distribution $\mathcal{D}_V$ in Theorem 2, the perturbations in Corollary 2 are followed. When $V < \exp(d(\eta-\epsilon)^2/2\sigma^2)$, the adversarially trained model will increase and decrease the natural and robust errors of class "-1" and class "$+1$", with the increase in ρ, respectively.*

In accordance with Corollary 4, the performance gaps between classes can be tuned by the combination strategy. In addition, it can contribute to the larger scope of the classification boundary compared with only adversaries, and a better tradeoff can be attained. When the same performance is achieved, combining adversaries and anti-adversaries has a smaller bound. Therefore, the combination strategy is more efficient than only the adversarial perturbation. More details are presented in the online material.

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. Two main conclusions are obtained. 1) The adversaries of noisy samples will harm the tradeoff and fairness of the robust model. 2) If noisy samples are anti-adversarially perturbed with a bound $\rho \times \epsilon$ and clean samples are adversarially perturbed with a bound ϵ, then the natural and robust errors of class "-1" and class "$+1$" will be decreased and increased with the increase in ρ, respectively. Thus, the combination strategy with varied bounds is effective in achieving a lower performance gap between classes and a better tradeoff between the accuracy and robustness on noisy data. The relevant theorems are shown in the online material.

Summarization

Our theoretical analysis comprehensively reveals that the perturbation directions and bounds remarkably influence the generalization, robustness, and fairness of the robust model under three typical learning scenarios. Adversarial training with different perturbation directions and bounds can better tune the performance gap between classes and the tradeoff between robustness and accuracy. Existing studies ignored anti-adversaries that are valuable. Thus, a new optimized objective considering anti-adversaries is proposed.

Methodology

Illuminated by the theoretical analysis, a new objective function is first established. Accordingly, a meta learning-based method that combines adversaries and anti-adversaries (CAAT) in training with a varied bound for each sample is proposed to solve the optimization, as shown in Fig. 2.

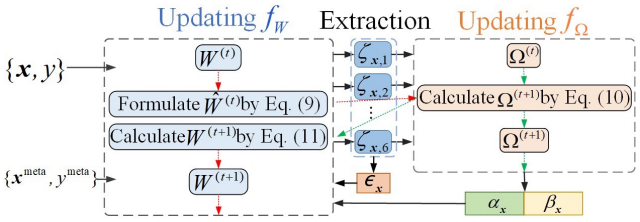


Figure 2: Overall structure of CAAT. The red and green lines represent the learning loops of the classifier network and weighting network, respectively.

Proposed Objective Function

Ideally, the objective function that combines adversaries and anti-adversaries can be formulated as

$$\min_{\boldsymbol{W},\alpha,\beta} \mathbb{E}_{\boldsymbol{x}}\{\alpha_{\boldsymbol{x}}\ell(f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}), y) + \beta_{\boldsymbol{x}}\ell(f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{at-adv}}), y)\}, \quad \text{s.t. } \alpha_{\boldsymbol{x}} + \beta_{\boldsymbol{x}} = 1 \text{ and } \alpha_{\boldsymbol{x}}, \beta_{\boldsymbol{x}} \in \{0, 1\}, \tag{6}$$

where $\boldsymbol{x}_{\text{adv}}$ and $\boldsymbol{x}_{\text{at-adv}}$ are calculated by using Eqs. (1) and (2) with varied bound $\epsilon_{\boldsymbol{x}}$ for each sample $\boldsymbol{x}$, respectively; $\alpha_{\boldsymbol{x}}$ and $\beta_{\boldsymbol{x}}$ are the combination weights; $f_{\boldsymbol{W}}$ is the classifier network with the parameter $\boldsymbol{W}$. When $\alpha_{\boldsymbol{x}} \equiv 1$, Eq. (6) can be reduced to the objective of standard adversarial training.

To solve Eq. (6), we first assume that the values of $\alpha_{\boldsymbol{x}}$ and $\beta_{\boldsymbol{x}}$ depend on the training characteristics of sample $\boldsymbol{x}$. Accordingly, their values are produced by a weighting network $f_{\boldsymbol{\Omega}}$ (parameterized by $\boldsymbol{\Omega}$), where its input is a series of training characteristics $\boldsymbol{\zeta}_{\boldsymbol{x}}$ of $\boldsymbol{x}$ shown in Fig. 2. $\ell(f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}), y)$ can be divided into $\ell(f_{\boldsymbol{W}}(\boldsymbol{x}), y)$ and $\ell(f_{\boldsymbol{W}}(\boldsymbol{x}), f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}))$ to achieve a better tradeoff between the accuracy and robustness (Zhang et al. 2019). To improve the fairness among classes, we further stress f to satisfy two fairness constraints following Ref. Xu et al.(2021). Thus, our adopted objective function is

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{\Omega}} \mathbb{E}_{\boldsymbol{x}}\{&\alpha_{\boldsymbol{x}}[\ell(f_{\boldsymbol{W}}(\boldsymbol{x}), y) + \lambda\ell(f_{\boldsymbol{W}}(\boldsymbol{x}), f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}))] \\ &+ \beta_{\boldsymbol{x}}\ell(f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{at-adv}}), y)\}, \\ \text{s.t. } &\begin{cases} [\alpha_{\boldsymbol{x}}, \beta_{\boldsymbol{x}}] = f_{\boldsymbol{\Omega}}(\boldsymbol{\zeta}_{\boldsymbol{x}}), \forall \boldsymbol{x} \in \mathcal{X}, \\ \mathcal{R}_{\text{nat}}(f_{\boldsymbol{W}}, c) - \mathcal{R}_{\text{nat}}(f_{\boldsymbol{W}}) \le \tau_1, \forall c \in \mathcal{Y}, \\ \mathcal{R}_{\text{bdy}}(f_{\boldsymbol{W}}, c) - \mathcal{R}_{\text{bdy}}(f_{\boldsymbol{W}}) \le \tau_2, \forall c \in \mathcal{Y}, \end{cases}\end{aligned} \tag{7}$$

where $\mathcal{R}_{\text{bdy}}$ is the boundary error of the model, denoted as $\mathcal{R}_{\text{bdy}}(f_{\boldsymbol{W}}) = \Pr(\exists \boldsymbol{x}_{\text{adv}} \in \mathbb{B}(\boldsymbol{x}, \epsilon), f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}) \neq f_{\boldsymbol{W}}(\boldsymbol{x}))$; $\mathcal{R}_{\text{nat}}(f_{\boldsymbol{W}}, c) = \Pr(f_{\boldsymbol{W}}(\boldsymbol{x}) \neq y \mid y = c)$; $\mathcal{R}_{\text{bdy}}(f_{\boldsymbol{W}}, c) = \Pr(\exists \boldsymbol{x}_{\text{adv}} \in \mathbb{B}(\boldsymbol{x}, \epsilon), f_{\boldsymbol{W}}(\boldsymbol{x}_{\text{adv}}) \neq f_{\boldsymbol{W}}(\boldsymbol{x}) \mid y = c)$; $f_{\boldsymbol{\Omega}}$ is a multilayer perception (MLP) network with a hidden layer and a τ-softmax layer: Softmax$((\boldsymbol{h}\boldsymbol{\omega} + \boldsymbol{b})/\tau)$; $\lambda > 0$ is a regularization parameter that adjusts the influence of the natural and boundary errors on the model; τ_1 and τ_2 are small and positive predefined parameters. The approach for solving the two fairness constraints is the same as that in Ref. Xu et al.(2021), where a Lagrangian is formed.

Extraction of Training Characteristics ($\boldsymbol{\zeta}_{\boldsymbol{x}}$)

Our theoretical investigation reveals that different training samples can have different perturbation directions. The perturbation direction of a training sample depends on a series

Algorithm 1: CAAT

Input: #Iteration T, step sizes η_0, η_1, and η_2, batch size n, meta batch size m, bound ϵ, #iterations K in inner optimization, classifier network $f_{\boldsymbol{W}}$, weighting network $f_{\boldsymbol{\Omega}}$, D^{train}, D^{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^{train} and D^{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 ϵ_i for sample $\boldsymbol{x}_i$;
    for $k = 1$ to K **do**
      $\boldsymbol{x}_i^{\text{adv}} \leftarrow \Pi_{\mathbb{B}(\boldsymbol{x}_i, \epsilon_i)}(\eta_0 \text{sign}(\nabla_{\boldsymbol{x}_i^{\text{adv}}} \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i), f_{\boldsymbol{W}}(\boldsymbol{x}_i^{\text{adv}}))) + \boldsymbol{x}_i^{\text{adv}})$, where Π is the projection operator;
      $\boldsymbol{x}_i^{\text{at-adv}} \leftarrow \Pi_{\mathbb{B}(\boldsymbol{x}_i, \epsilon_i)}(-\eta_0 \text{sign}(\nabla_{\boldsymbol{x}_i^{\text{at-adv}}} \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i^{\text{at-adv}}), y_i)) + \boldsymbol{x}_i^{\text{at-adv}})$;
    end for
  end for
  Formulate $\hat{\boldsymbol{W}}^{(t)}(\boldsymbol{\Omega})$ by Eq. (9);
  Update $\boldsymbol{\Omega}^{(t+1)}$ by Eq. (10) and update $\boldsymbol{W}^{(t+1)}$ by Eq. (11);
end for

of factors, including learning difficulty, class proportion, and noise degree. Therefore, six training characteristics of each training sample $\boldsymbol{x}$, namely, 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 ($\boldsymbol{\zeta}_{\boldsymbol{x},6}$), are extracted, as shown in the extraction module in Fig. 2. The calculation detail of each characteristic is shown in the online material.

Perturbation Bound ($\epsilon_{\boldsymbol{x}}$) Calculation

We employ two types of varied bound in our framework. Following Ref. Xu et al.(2021), the class-wise perturbation bound named ReMargin, which is suitable for imbalanced data, is utilized. A sample-wise bound is proposed to handle noise. It is inspired by the intuition that noisy samples have a large norm of loss gradient in general and these samples should exhibit the greatest degree of anti-adversarial training. Thus, the Grad-Based bound can be calculated as

$$\epsilon_{\boldsymbol{x}} = (\alpha_{\boldsymbol{x}} \overline{\boldsymbol{g}}_{\boldsymbol{x}_{\text{adv}}} + \beta_{\boldsymbol{x}} \overline{\boldsymbol{g}}_{\boldsymbol{x}_{\text{at-adv}}} + \varepsilon) \times \epsilon, \tag{8}$$

where $\overline{\boldsymbol{g}}_{\boldsymbol{x}_{\text{adv}}}$ and $\overline{\boldsymbol{g}}_{\boldsymbol{x}_{\text{at-adv}}}$ 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 perturbation 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 training.

Training with Meta-Learning

On the basis of the extracted characteristics and calculated bounds, an online learning strategy is adopted to alternatively update $\boldsymbol{W}$ and $\boldsymbol{\Omega}$ using a single optimization loop, as shown in Fig. 2. 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} (Zhang and Pfister 2021). The main steps are shown below. Here, we ignore the regularization terms introduced by the fairness constraints, while the online material provides the complete formulas.

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

$$\hat{\boldsymbol{W}}^{(t)}(\boldsymbol{\Omega}) = \boldsymbol{W}^{(t)} - \eta_1 \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{W}} \{\alpha_i [\ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i), y_i) + \lambda \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i), f_{\boldsymbol{W}}(\boldsymbol{x}_i^{\text{adv}}))] + \beta_i \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i^{\text{at-adv}}), y_i)\}|_{\boldsymbol{W}^{(t)}}, \tag{9}$$

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

$$\boldsymbol{\Omega}^{(t+1)} = \boldsymbol{\Omega}^{(t)} - \eta_2 \frac{1}{m} \sum_{i=1}^{m} \nabla_{\boldsymbol{\Omega}} [\ell^{\text{meta}}(f_{\hat{\boldsymbol{W}}^{(t)}(\boldsymbol{\Omega})}(\boldsymbol{x}_i), y_i) + \lambda \ell^{\text{meta}}(f_{\hat{\boldsymbol{W}}^{(t)}(\boldsymbol{\Omega})}(\boldsymbol{x}_i), f_{\hat{\boldsymbol{W}}^{(t)}(\boldsymbol{\Omega})}(\boldsymbol{x}_i^{\text{adv}})) + \ell^{\text{meta}}(f_{\hat{\boldsymbol{W}}^{(t)}(\boldsymbol{\Omega})}(\boldsymbol{x}_i^{\text{at-adv}}), y_i)]|_{\boldsymbol{\Omega}^{(t)}}, \tag{10}$$

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

$$\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \eta_1 \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{W}} \{\alpha_i [\ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i), y_i) + \lambda \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i), f_{\boldsymbol{W}}(\boldsymbol{x}_i^{\text{adv}}))] + \beta_i \ell(f_{\boldsymbol{W}}(\boldsymbol{x}_i^{\text{at-adv}}), y_i)\}|_{\boldsymbol{W}^{(t)}}. \tag{11}$$

The steps of our CAAT method are shown in Algorithm 1.

Experiments

Experiments are conducted to verify our theoretical findings and the effectiveness of the proposed CAAT in improving the accuracy, robustness, and fairness of the robust models.

Experimental Settings

Benchmark adversarial learning datasets: CIFAR10 (Krizhevsky 2009) and SVHN (Netzer et al. 2011) are adopted in our experiments, including the noisy and imbalanced versions of the CIFAR data (Shu et al.

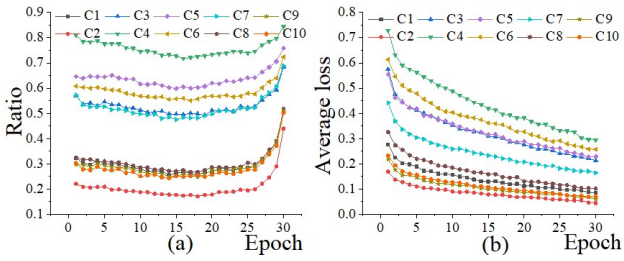


Figure 3: (a): Ratio of adversaries in each class during training on standard CIFAR10. (b): Average loss of each class during training on standard CIFAR10.

	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	49.3	77.6
Baseline ReWeight	19.2	28.3	39.2	53.7	58.2	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
CAAT (Grad-Based)	14.6	23.6	**14.4**	**23.3**	**28.6**	48.1
CAAT (ReMargin)	**13.9**	24.3	15.4	24.9	29.3	**44.4**

Table 1: Average and worstclass natural, boundary, and robust errors (%) for various algorithms on CIFAR10.

2019). For the two datasets, PreAct-ResNet18 (He et al. 2016) and Wide-ResNet28-10 (WRN28-10) (Zagoruyko and Komodakis 2016) are adopted as the backbone network. This section only represents the results of PreAct-ResNet18. Others are presented in the online material. The compared methods include three popular adversarial training algorithms, namely, PGD (Madry et al. 2018), TRADES (Zhang et al. 2019), and FRL (Xu et al. 2021). A debiasing method (Alekh et al. 2018) is also compared which is to upweight the loss of the class 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 used in Ref. Xu et al. (2021) are followed. The number of iterations in an adversarial attack is set to 10. Following Xu et al. (2021), 300 samples in each class with clean labels are selected as the meta dataset, which helps us tune the hyperparameters and train the weighting network. Adversarial training is trained on PGD attack setting $\epsilon = 8/255$ with cross-entropy loss. For our method and FRL (ReMargin), the predefined perturbation bound is also set to $8/255$. All the models are trained by using SGD with momentum 0.9 and weight decay 5×10^{-4}. The value of λ is selected in $\{2/3, 1, 1.5, 6\}$. During the evaluation phase, we report each model's average and worstclass natural, boundary, and robust error rates.

Experiments on Standard Dataset

Tables 1 shows the performance of our proposed CAAT and the compared methods on standard CIFAR10. Those on SVHN are shown in the online material. Considering that our training/testing configuration is the same as that in Ref. Xu et al.(2021), the results of the above competing methods reported in the FRL (Xu et al. 2021) paper are directly presented.

From the results, our methods with two types of bound reduce the average natural and robust errors under different degrees, indicating that CAAT obtains better accuracy and robustness of the model. Compared with other methods, CAAT decreases the average and worst robust error rates by 21% and 25% on CIFAR10. Baseline ReWeight can only decrease the worst intraclass natural error but cannot equalize boundary or robust errors. FRL (Xu et al. 2021) has only a limited ability to reduce the worst boundary and robust errors, resulting in limited fairness between classes. Our method more effectively decreases the worst intraclass errors. Thus, CAAT achieves better fairness among classes compared with other methods. Although FRL (ReWeight) obtains the lowest worst natural error, it has large average and worst robust errors, which is inferior to CAAT. Hard classes (classes with a large average loss) have a higher ratio of adversaries than easy ones, as shown in Fig. 3, which helps improve the performance of hard classes and effectively enhances the fairness among classes. The same conclusions can also be obtained on the SVHN dataset.

	Avg. Nat.	Avg. Bdy.	Avg. Rob.
PGD Adv. Training	15.6	37.1	52.8
TRADES ($1/\lambda = 1$)	15.6	31.0	46.5
TRADES ($1/\lambda = 6$)	16.4	21.0	37.4
FRL (ReWeight)	15.3	36.0	51.4
FRL (ReMargin)	15.2	36.0	51.1
FRL (ReWeight+ReMargin)	15.7	34.3	50.0
CAAT (Grad-Based)	**14.6**	**13.9**	**28.5**
CAAT (ReMargin)	14.7	14.7	29.4

Table 2: Average natural, boundary, and robust errors (%) for various algorithms on CIFAR10 with 20% pair-flip noise.

Experiments of Noisy Classification

Two settings of corrupted labels, including uniform and pair-flip noises, are adopted (Shu et al. 2019). The values of the noise ratio are set to 20% and 40%. CIFAR10 dataset,

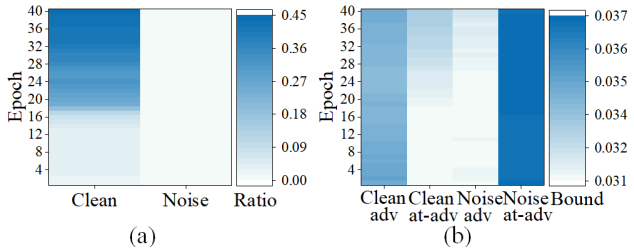


Figure 4: (a): Ratio of adversaries for noisy and clean samples on CIFAR10 with 20% uniform noise during training. (b): Average adversarial and anti-adversarial perturbation bounds for clean and noisy samples during training.

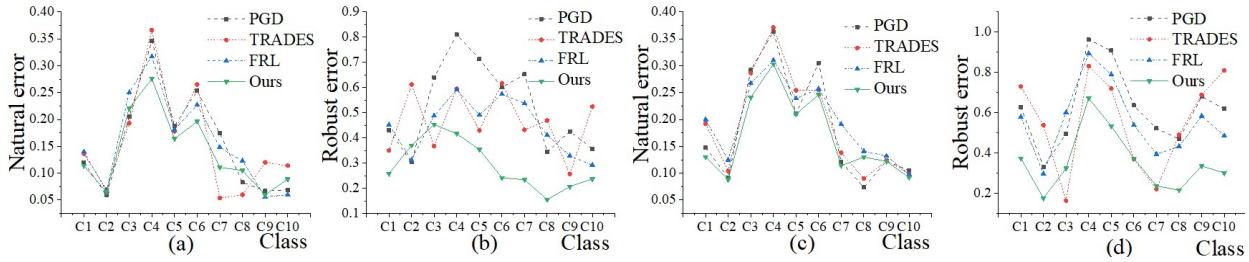


Figure 5: (a) and (b): Natural and robust errors for each class of different methods on CIFAR10 with imbalance factor 10. (c) and (d): Natural and robust errors for each class of different methods on CIFAR10 with imbalance factor 100.

	Avg. Nat.	Avg. Bdy.	Avg. Rob.
PGD Adv. Training	20.1	42.8	62.9
TRADES ($1/\lambda = 1$)	16.8	32.3	49.1
TRADES ($1/\lambda = 6$)	23.6	23.8	47.4
FRL (ReWeight)	16.9	38.1	55.0
FRL (ReMargin)	17.5	35.6	53.1
FRL (ReWeight+ReMargin)	17.2	35.1	52.3
CAAT (Grad-Based)	**15.8**	14.2	30.0
CAAT (ReMargin)	16.2	**13.7**	**29.9**

Table 3: Average and worstclass natural, boundary, and robust errors (%) on CIFAR10 with imbalance factor 10.

which is popularly used for the evaluation of noisy labels, is adopted. Here, we only show the average errors of CIFAR10 with 20% pair-flip noise. Others are presented in the online material. From the results in Table 2 and the online material, CAAT achieves the lowest average and worst natural and robust errors, indicating that it obtains the best generalization, robustness, and fairness compared with other methods.

As shown in Fig. 4 (a), most of the noisy samples are anti-adversarially perturbed during training, which is in accordance with our theoretical findings. From Fig. 4 (b), the average anti-adversarial perturbation bound for noisy samples is the largest, implying that noisy samples exhibit the largest degree of anti-adversarial training. Thus, the negative influence of noisy samples can be decreased. The ratio of adversaries for clean samples increases with the progress of training, demonstrating that clean samples are playing a more important role than noisy ones during training.

Experiments of Imbalanced Classification

The long-tailed version of CIFAR10 compiled by Cui et al. (2019) is utilized. The values of the imbalance factor are set to 10 and 100. Here, we only show the average results when the imbalance factor equals 10. Others are presented in the online material. Compared with other methods, CAAT achieves the minimum average and worst natural and robust errors, as shown in Table 3. As shown in Fig. 5, CAAT decreases the natural and robust errors for most classes and achieves the lowest performance gap among different classes. We also verify that the first head class has the lowest ratio of adversaries and tail classes have a high ratio of adversaries, which is consistent with our theoretical findings.

	Avg. Nat. (%)	Avg. Bdy. (%)	Avg. Rob. (%)
Setting I	16.0	41.6	57.6
Setting II	16.1	35.8	51.9
Setting III	14.9	**13.8**	**28.7**
Setting IV	**13.9**	15.4	29.3

Table 4: Ablation studies of CAAT on standard CIFAR10.

The details are presented in the online material.

Ablation Studies

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). PreAct-ResNet18 is used. The results are shown in Table 4. Settings III and IV obtain better performance compared with Settings I and II. Thus, the combination strategy is more effective. Compared with Setting III, Setting IV further decreases the average natural error, indicating that the varied bound is more valid in some cases. The worst errors are shown in the online material.

Conclusions

This study theoretically investigates the role of adversarial training with different directions (adversarial and anti-adversarial) and bounds for the robust model. Three typical occasions are considered, including classes with different difficulties, imbalance learning, and noisy label learning. A series of theoretical findings are obtained, illuminating a new objective function that combines adversaries and anti-adversaries in training. Consequently, an adversarial training framework (CAAT) is proposed to solve the objective, in which meta learning is utilized to optimize the combined weights of the adversary and anti-adversary for each sample in accordance with its learning characteristics. Extensive experiments verify the rationality of our theoretical findings and the effectiveness of CAAT in achieving better accuracy, robustness, and fairness of the robust models compared with other adversarial training methods.

Acknowledgments

This study is partially supported by NSFC 62076178, TJF 22ZYYYJC00020, and 19ZXAZNGX00050.

References

Alekh, A.; Alekh, A.; Alekh, A.; Alekh, A.; and Alekh, A. 2018. A Reductions Approach to Fair Classification. In *ICML*, 102–119.

Alfarra, M.; Pérez, J. C.; Thabet, A.; Bibi, A.; Torr, P. H. S.; and Ghanem, B. 2022. Combating Adversaries with Anti-Adversaries. In *AAAI*, 1–13.

Bai, T.; and Luo, J. 2021. Recent Advances in Adversarial Training for Adversarial Robustness. In *IJCAI*, 4312–4321.

Balaji, Y.; Goldstein, T.; and Hoffman, J. 2019. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. arXiv:1910.08051.

Cheng, M.; Lei, Q.; Chen, P.-Y.; Dhillon, I.; and Hsieh, C.-J. 2020. CAT: Customized Adversarial Training for Improved Robustness. arXiv:2002.06789.

Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-Balanced Loss Based on Effective Number of Samples. In *CVPR*, 9260–9270.

Ding, G. W.; Sharma, Y.; Lui, K. Y. C.; and Huang, R. 2020. MMA Training: Direct Input Space Margin Maximization through Adversarial Training. In *ICLR*, 1–34.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*, 1856–1868.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and Harnessing Adversarial Examples. arXiv:1412.6572.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. *Scandinavian Journal of Statistics*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*, 1–18.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, R.; Wu, B.; and Ng, A. Y. 2011. Reading Digits in Natural Images with Unsupervised Feature Learning. *Scandinavian Journal of Statistics*.

Nichol, A.; Achiam, J.; and Schulman, J. 2018. On First-Order Meta-Learning Algorithms. arXiv:1803.02999.

Pang, T.; Yang, X.; Dong, Y.; Su, H.; and Zhu, J. 2021. Bag of Tricks for Adversarial Training. In *ICLR*, 1–21.

Raghunathan, A.; Xie, S. M.; Yang, F.; Duchi, J. C.; and Liang, P. 2019. Adversarial Training Can Hurt Generalization. arXiv:1906.06032.

Ren, M.; Zeng, W.; Yang, B.; and Urtasun, R. 2018. Learning to Reweight Examples for Robust Deep Learning. In *ICML*, 6900–6909.

Rice, L.; Wong, E.; and Kolter, J. Z. 2020. Overfitting in adversarially robust deep learning. In *ICML*, 8093–8104.

Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; and Wierstra, D. 2016. Meta Learning With Memory-Augmented Neural Networks. In *ICML*, 2740–2751.

Shu, J.; Xie, Q.; Yi, L.; Zhao, Q.; Zhou, S.; Xu, Z.; and Meng, D. 2019. Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting. In *NeurIPS*, 1917–1928.

Snell, J.; Swersky, K.; and Zemel, R. S. 2017. Prototypical Networks For Few Shot Learning. In *NeurIPS*, 4078–4088.

Song, C.; He, K.; Lin, J.; Wang, L.; and Hopcroft, J. E. 2020. Robust Local Features for Improving the Generalization of Adversarial Training. In *ICLR*, 1–12.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning To Compare: Relation Network For Few-Shot Learning. In *CVPR*, 1199–1208.

Uesato, J.; Alayrac, J.-B.; Huang, P.-S.; Stanforth, R.; Fawzi, A.; and Kohli, P. 2019. Are Labels Required for Improving Adversarial Robustness? In *NeurIPS*, 12214–12223.

Wang, Y.; Zou, D.; Yi, J.; Bailey, J.; Ma, X.; and Gu, Q. 2020. Improving Adversarial Robustness Requires Revisiting Misclassified Examples. In *ICLR*, 1–14.

Wong, E.; Rice, L.; and Kolter, J. Z. 2020. Fast is better than free: Revisiting adversarial training. In *ICLR*, 1–17.

Xie, C.; Tan, M.; Gong, B.; Yuille, A.; and Le, Q. V. 2020. Smooth Adversarial Training. arXiv:2002.11242.

Xu, H.; Liu, X.; Li, Y.; Jain, A. K.; and Tang, J. 2021. To be Robust or to be Fair: Towards Fairness in Adversarial Training. In *ICML*, 11492–11501.

Yang, S.; Guo, T.; Wang, Y.; and Xu, C. 2021. Adversarial Robustness through Disentangled Representations. In *AAAI*, 3145–3153.

Yang, Y.-Y.; Rashtchian, C.; Zhang, H.; Salakhutdinov, R.; and Chaudhuri, K. 2020. A Closer Look at Accuracy vs. Robustness. In *NeurIPS*, 8588–8601.

Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. In *BMVC*, 1–12.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E. P.; Ghaoui, L. E.; and Jordan, M. I. 2019. Theoretically Principled Trade-off between Robustness and Accuracy. In *ICML*, 12907–12929.

Zhang, J.; Xu, X.; Han, B.; Niu, G.; Cui, L.; Sugiyama, M.; and Kankanhalli, M. 2020. Attacks Which Do Not Kill Training Make Adversarial Learning Stronger. In *ICML*, 1–15.

Zhang, Z.; and Pfister, T. 2021. Learning Fast Sample Reweighting Without Reward Data. In *ICCV*, 705–714.

Zhu, J.; Zhang, J.; Han, B.; Liu, T.; Niu, G.; Yang, H.; Kankanhalli, M.; and Sugiyama, M. 2021. Understanding the Interaction of Adversarial Training with Noisy Labels. arXiv:2102.03482.