IMPROVING ADVERSARIAL ROBUSTNESS BY PUTTING MORE REGULARIZATIONS ON LESS ROBUST SAMPLES

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarial training, which is to enhance robustness against adversarial attacks, has received much attention because it is easy to generate human-imperceptible perturbations of data to deceive a given deep neural network. In this paper, we propose a new adversarial training algorithm that is theoretically well motivated and empirically superior to other existing algorithms. A novel feature of the proposed algorithm is to apply more regularization to data vulnerable to adversarial attacks than other existing regularization algorithms do. Theoretically, we show that our algorithm can be understood as an algorithm of minimizing a newly derived upper bound of the robust risk. Numerical experiments illustrate that our proposed algorithm improves the generalization (accuracy on examples) and robustness (accuracy on adversarial attacks) simultaneously to achieve the state-of-the-art performance.

1 INTRODUCTION

It is easy to generate human-imperceptible perturbations that put prediction of a deep neural network (DNN) out. Such perturbed samples are called *adversarial examples* (Szegedy et al., 2014) and algorithms for generating adversarial examples are called *adversarial attacks*. It is well known that adversarial attacks can greatly reduce the accuracy of DNNs, for example from about 96% accuracy on clean data to almost zero accuracy on adversarial examples (Madry et al., 2018). This vulnerability of DNNs can cause serious security problems when DNNs are applied to security critical applications (Kurakin et al., 2017; Jiang et al., 2019) such as medicine (Ma et al., 2020; Finlayson et al., 2019) and autonomous driving (Kurakin et al., 2017; Deng et al., 2020; Morgulis et al., 2019; Li et al., 2020).

Adversarial training, which is to enhance robustness against adversarial attacks, has received much attention. Various adversarial training algorithms can be categorized into two types. The first one is to learn prediction models by minimizing the robust risk - the risk for adversarial examples. PGD-AT (Madry et al., 2018) is the first of its kinds and various modifications including Zhang et al. (2020); Ding et al. (2020); Zhang et al. (2021) have been proposed since then.

The second type of adversarial training algorithms is to minimize the regularized risk which is the sum of the empirical risk for clean examples and a regularized term related to adversarial robustness. TRADES (Zhang et al., 2019) decomposes the robust risk into the sum of the natural and boundary risks, where the first one is the risk for clean examples and the second one is the remaining part, and replaces them to their upper bounds to have the regularized risk. HAT (Rade & Moosavi-Dezfolli, 2022) modifies the regularization term of TRADES by adding an additional regularization term based on helper samples.

The aim of this paper is to develop a new adversarial training algorithm for DNNs, which is theoretically well motivated and empirically superior to other existing competitors. Our algorithm modifies the regularization term of TRADES (Zhang et al., 2019) to put more regularization on less robust samples. This new regularization term is motivated by an upper bound of the boundary risk.

Our proposed regularized term is similar to that used in MART (Wang et al., 2020). The two key differences are that (1) the objective function of MART consists of the sum of the robust risk and regularization term while ours consists of the sum of the natural risk and regularization term and (2) our algorithm regularizes less robust samples more but MART regularizes less accurate samples more. Note that our algorithm is theoretically well motivated from an upper bound of the robust risk

but no such theoretical explanation of MART is available. In numerical studies, we demonstrate that our algorithm outperforms MART as well as TRADES with large margins.

1.1 Our Contributions

We propose a new adversarial training algorithm. Novel features of our algorithm compared to other existing adversarial training algorithms are that it is theoretically well motivated and empirically superior. Our contributions can be summarized as follows:

- We derive an upper bound of the robust risk which is tighter than that of TRADES.
- As a surrogate version of this upper bound, we propose a new regularized risk.
- We develop an adversarial training algorithm that learns a robust prediction model by minimizing the proposed regularized risk.
- By analyzing benchmark data sets, we show that our proposed algorithm is superior to other competitors in view of the generalization (accuracy on clean examples) and robustness (accuracy on adversarial examples) simultaneously to achieve the state-of-the-art performance.
- We illustrate that our algorithm is helpful to improve the fairness of the prediction model in the sense that the error rates of each class become more similar compared to TRADES.

2 Preliminaries

2.1 Robust Population Risk

Let $\mathcal{X} \subset \mathbb{R}^d$ be the input space, $\mathcal{Y} = \{1, \cdots, C\}$ be the set of output labels and $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}^C$ be the score function parameterized by the neural network parameters $\boldsymbol{\theta}$ (the vector of weights and biases) such that $\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}) = \mathrm{softmax}(f_{\boldsymbol{\theta}}(\boldsymbol{x}))$ is the vector of the conditional class probabilities. Let $F_{\boldsymbol{\theta}}(\boldsymbol{x}) = \arg\max_c [f_{\boldsymbol{\theta}}(\boldsymbol{x})]_c$, $\mathcal{B}_p(\boldsymbol{x}, \varepsilon) = \{\boldsymbol{x}' \in \mathcal{X} : \|\boldsymbol{x} - \boldsymbol{x}'\|_p \le \varepsilon\}$ and $\mathbb{1}(\cdot)$ be the indicator function. Let capital letters $\mathbf{X}, \mathbf{Y}$ denote random variables or vectors and small letters $\boldsymbol{x}, y$ denote their realizations.

The robust population risk used in the adversarial training is defined as

$$\mathcal{R}_{\mathrm{rob}}(\theta) := \mathbb{E}_{(\mathbf{X},\mathbf{Y})} \max_{\mathbf{X}' \in \mathcal{B}_p(\mathbf{X},\varepsilon)} \mathbb{1}\left\{F_{\boldsymbol{\theta}}(\mathbf{X}') \neq \mathbf{Y}\right\}, \tag{1}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are a random vector in $\mathcal{X}$ and a random variable in $\mathcal{Y}$, respectively. Most adversarial training algorithms learn $\boldsymbol{\theta}$ by minimizing an empirical version of the above robust population risk. In turn, most empirical versions of (1) require to generate an *adversarial example* which is a surrogate version of

$$\boldsymbol{x}^{\mathrm{adv}} := \arg\max_{\boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x},\varepsilon)} \mathbb{1}\left\{F_\theta(\boldsymbol{x}') \neq y\right\}.$$

Any method of generating an adversarial example is called an *adversarial attack.*

2.2 Algorithms for Generating Adversarial Examples

Existing adversarial attacks can be categorized into either the white-box attack (Goodfellow et al., 2015; Madry et al., 2018; Carlini & Wagner, 2017; Croce & Hein, 2020a) or the black-box attack (Papernot et al., 2016; 2017; Chen et al., 2017; Ilyas et al., 2018; Papernot et al., 2018). For the white-box attack, the model structure and parameters are known to adversaries who use this information for generating adversarial examples, while outputs for given inputs are only available to adversaries for the black-box attack.

The most popular method for the white-box attack is PGD (Projected Gradient Descent) (Madry et al., 2018). Let $\eta(\boldsymbol{x}'|\theta, \boldsymbol{x}, y)$ be a surrogate loss of $\mathbb{1}\left\{F_\theta(\boldsymbol{x}') \neq y\right\}$ for given $\boldsymbol{\theta}, \boldsymbol{x}, y$. PGD finds the adversarial example by applying the gradient ascent algorithm to η to update $\boldsymbol{x}'_\eta$ and projecting it to $\mathcal{B}_p(\boldsymbol{x}, \varepsilon)$. That is, the update rule of PGD is

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{\Pi}_{\mathcal{B}_p(\boldsymbol{x},\varepsilon)}\left(\boldsymbol{x}^{(t)} + \nu\, \mathrm{sgn}\left(\nabla_{\boldsymbol{x}^{(t)}} \eta(\boldsymbol{x}^{(t)}|\boldsymbol{\theta}, \boldsymbol{x}, y)\right)\right), \tag{2}$$

where $\nu > 0$ is the step size, $\mathbf{\Pi}_{\mathcal{B}_p(\boldsymbol{x},\varepsilon)}(\cdot)$ is the projection operator to $\mathcal{B}_p(\boldsymbol{x},\varepsilon)$ and $\boldsymbol{x}^{(0)} = \boldsymbol{x}$. We define $\boldsymbol{x}^{\text{pgd}}$ as $\boldsymbol{x}^{\text{pgd}} := \lim_{t\to\infty} \boldsymbol{x}^{(t)}$. For the surrogate loss η, the cross entropy (Madry et al., 2018) or the KL divergence (Zhang et al., 2019) is used.

For the black-box attack, an adversary generates a dataset $\{\boldsymbol{x}_i, \tilde{y}_i\}_{i=1}^n$ where $\tilde{y}_i$ is an output of a given input $\boldsymbol{x}_i$. Then, the adversary trains a substitute prediction model based on this data set, and generates adversarial examples from the substitute prediction model by PGD (Papernot et al., 2017).

2.3 Review of Adversarial Training Algorithms

We review some of the adversarial training algorithms which, we think, are related to our proposed algorithm. Typically, adversarial training algorithms consist of the maximization and minimization steps. In the maximization step, we generate adversarial examples for given $\boldsymbol{\theta}$, and in the minimization step, we fix the adversarial examples and update $\boldsymbol{\theta}$. In the followings, we denote $\widehat{\boldsymbol{x}}_i^{\text{pgd}}$ as the adversarial example corresponding to $(\boldsymbol{x}_i, y_i)$ generated by PGD.

2.3.1 Algorithms minimizing the robust risk directly

PGD-AT Madry et al. (2018) proposes PGD-AT which updates $\boldsymbol{\theta}$ by minimizing

$$\sum_{i=1}^n \ell_{\text{ce}}(f_{\boldsymbol{\theta}}(\widehat{\boldsymbol{x}}_i^{\text{pgd}}), y_i),$$

where ℓ_{ce} is the cross-entropy loss.

GAIR-AT Geometry Aware Instance Reweighted Adversarial Training (GAIR-AT) (Zhang et al., 2021) is a modification of PGD-AT, where the weighted robust risk is minimized and more weights are given to samples closer to the decision boundary. To be more specific, the weighted empirical risk of GAIR-AT is given as

$$\sum_{i=1}^n w_\theta(\boldsymbol{x}_i, y_i)\ell_{\text{ce}}(f_{\boldsymbol{\theta}}(\widehat{\boldsymbol{x}}_i^{\text{pgd}}), y_i),$$

where $\kappa_\theta(\boldsymbol{x}_i, y_i) = \min\left(\min(\{t : F_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(t)}) \neq y_i\}), T\right)$ for a prespecified maximum iteration T and $w_\theta(\boldsymbol{x}_i, y_i) = (1 + \tanh(5(1 - 2\kappa_\theta(\boldsymbol{x}_i, y_i)/T)))/2$.

There are other similar modifications of PGA-AT including Max-Margin Adversarial (MMA) Training (Ding et al., 2020) and Friendly Adversarial Training (FAT) (Zhang et al., 2020).

2.3.2 Algorithms minimizing a regularized empirical risk

Robust risk, natural risk and boundary risk are defined by

$$\begin{aligned}
\mathcal{R}_{\text{rob}}(\boldsymbol{\theta}) &= \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}\left\{\exists \mathbf{X}' \in \mathcal{B}_p(\mathbf{X},\varepsilon) : F_{\boldsymbol{\theta}}(\mathbf{X}') \neq Y\right\}, \\
\mathcal{R}_{\text{nat}}(\boldsymbol{\theta}) &= \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}\left\{F_{\boldsymbol{\theta}}(\mathbf{X}) \neq Y\right\}, \\
\mathcal{R}_{\text{bdy}}(\boldsymbol{\theta}) &= \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}\left\{\exists \mathbf{X}' \in \mathcal{B}_p(\mathbf{X},\varepsilon) : F_{\boldsymbol{\theta}}(\mathbf{X}) \neq F_{\boldsymbol{\theta}}(\mathbf{X}'), F_{\boldsymbol{\theta}}(\mathbf{X}) = Y\right\}.
\end{aligned}$$

Zhang et al. (2019) shows

$$\mathcal{R}_{\text{rob}}(\boldsymbol{\theta}) = \mathcal{R}_{\text{nat}}(\boldsymbol{\theta}) + \mathcal{R}_{\text{bdy}}(\boldsymbol{\theta}).$$

By treating $\mathcal{R}_{\text{bdy}}(\boldsymbol{\theta})$ as the regularization term, various regularized risks for adversarial training have been proposed.

TRADES Zhang et al. (2019) proposes the following regularized empirical risk which is a surrogate version of the upper bound of the robust risk:

$$\sum_{i=1}^n \left\{\ell_{\text{ce}}(f_\theta(\boldsymbol{x}_i), y_i) + \lambda \cdot \text{KL}(\mathbf{p}_\theta(\cdot|\boldsymbol{x}_i)\|\mathbf{p}_\theta(\cdot|\widehat{\boldsymbol{x}}_i^{\text{pgd}}))\right\},$$

HAT Helper based training (Rade & Moosavi-Dezfolli, 2022) is a variation of TRADES where an additional regularization term based on helper examples is added to the regularized risk. The role of helper examples is to restrain the decision boundary from having excessive margins. HAT minimizes the following regularized empirical risk:

$$\sum_{i=1}^{n}\left[\ell_{\text{ce}}\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i\right), y_i\right) + \lambda \cdot \text{KL}\left(\mathbf{p}_\theta\left(\cdot|\boldsymbol{x}_i\right) \| \mathbf{p}_\theta(\cdot|\widehat{\boldsymbol{x}}_i^{\text{pgd}})\right) + \gamma \ell_{\text{ce}}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{\text{helper}}), F_{\boldsymbol{\theta}_{\text{pre}}}(\widehat{\boldsymbol{x}}_i^{\text{pgd}})\right)\right],$$

where $\boldsymbol{\theta}_{\text{pre}}$ is the parameter of a pre-trained model only with clean examples, $\boldsymbol{x}_i^{\text{helper}} = \boldsymbol{x}_i + 2(\widehat{\boldsymbol{x}}_i^{\text{pgd}} - \boldsymbol{x}_i)$.

MART Misclassification Aware adveRsarial Training (MART) (Wang et al., 2020) minimizes

$$\sum_{i=1}^{n}\left\{\ell_{\text{margin}}(f_{\boldsymbol{\theta}}(\widehat{\boldsymbol{x}}_i^{\text{pgd}}), y_i) + \lambda \cdot \text{KL}(\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}_i)\|\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\widehat{\boldsymbol{x}}_i^{\text{pgd}}))(1 - p_{\boldsymbol{\theta}}(y_i|\boldsymbol{x}_i))\right\}, \tag{3}$$

where $\ell_{\text{margin}}(f_{\boldsymbol{\theta}}(\widehat{\boldsymbol{x}}_i^{\text{pgd}}), y_i) = -\log p_{\boldsymbol{\theta}}(y_i|\widehat{\boldsymbol{x}}_i^{\text{pgd}}) - \log(1 - \max_{k \neq y_i} p_{\boldsymbol{\theta}}(k|\widehat{\boldsymbol{x}}_i^{\text{pgd}}))$. This objective function can be regarded as the regularized robust risk and thus MART can be considered as a hybrid algorithm of PGD-AT and TRADES.

3 Anti-Robust Weighted Regularization (ARoW)

In this section, we develop a new adversarial training algorithm called Anti-Robust Weighted Regularization (ARoW), which is an algorithm minimizing a regularized risk. We propose a new regularized term which applies more regularization to data vulnerable to adversarial attacks than other existing algorithms such as TRADES and HAT do. Our new regularized term is motivated by the upper bound of the robust risk derived in the following section.

3.1 Upper Bound of the Robust Risk

In this subsection, we provide an upper bound of the robust risk for multi-classification problem which is stated in the following theorem. The proof is deferred to Appendix A.

Theorem 1. *For a given score function $f_{\boldsymbol{\theta}}$, let $z(\cdot)$ be an any measurable mapping from $\mathcal{X}$ to $\mathcal{X}$ satisfying*

$$z(\boldsymbol{x}) \in \underset{\boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x}, \varepsilon)}{\arg\max}\, \mathbb{1}\left(F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(\boldsymbol{x}')\right).$$

for every $\boldsymbol{x} \in \mathcal{X}$. Then, we have

$$\mathcal{R}_{rob}(\boldsymbol{\theta}) \leq \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}(Y \neq F_{\boldsymbol{\theta}}(\mathbf{X})) + \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}(F_{\boldsymbol{\theta}}(\mathbf{X}) \neq F_{\boldsymbol{\theta}}(z(\mathbf{X})))\mathbb{1}\left\{p_{\boldsymbol{\theta}}(Y|z(\mathbf{X})) < 1/2\right\} \tag{4}$$

The upper bound (4) consists of the two terms : the first term is the natural risk itself and the second term is an upper bound of the boundary risk. This upper bound is motivated by the upper bound derived in TRADES (Zhang et al., 2019). For binary classification problems, Zhang et al. (2019) shows that

$$\mathcal{R}_{\text{rob}}(\boldsymbol{\theta}) \leq \mathbb{E}_{(\mathbf{X},Y)}\phi(Y f_{\boldsymbol{\theta}}(\mathbf{X})) + \mathbb{E}_{\mathbf{X}}\phi(f_{\boldsymbol{\theta}}(\mathbf{X}) f_{\boldsymbol{\theta}}(z(\mathbf{X}))), \tag{5}$$

where

$$z(\boldsymbol{x}) \in \underset{\boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x}, \varepsilon)}{\arg\max}\, \phi\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}) f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$$

and $\phi(\cdot)$ is an upper bound of $\mathbb{1}(\cdot < 0)$. Our upper bound (4) is a modification of the upper bound (5) for multiclass problems where $\phi(\cdot)$ and $f_{\boldsymbol{\theta}}$ in (5) are replaced by $\mathbb{1}(\cdot < 0)$ and $F_{\boldsymbol{\theta}}$, respectively. A key difference, however, between (4) and (5) is the term $\mathbb{1}\left\{p_{\boldsymbol{\theta}}(Y|z(\mathbf{X})) < 1/2\right\}$ at the last part of (4) that is not in (5). That is, our upper bound (4) is tighter than the upper bound (5) even after we replace $\phi(\cdot)$ by $\mathbb{1}(\cdot < 0)$ in (5).

Algorithm 1 ARoW Algorithm

Input : network $f_{\boldsymbol{\theta}}$, training dataset $\mathcal{D} = \left\{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^{d+1} : i = 1, \cdots, n\right\}$, learning rate η, hyperparameters (λ, α) of (6), number of epochs T, number of batch B, batch size K

Output : adversarially robust network $f_{\boldsymbol{\theta}}$

for $t = 1, \cdots, T$ **do**
  for $b = 1, \cdots, B$ **do**
    $\widehat{\boldsymbol{x}}^{\text{pgd}}_{t,b,k} \leftarrow \underset{\boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x}_{t,b,k}, \varepsilon)}{\arg\max} \ \mathrm{KL}(\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}_{t,b,k}) \| \mathbf{p}_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}'))$; $\boldsymbol{x}_{t,b,k} \in \mathbb{R}^d$, $k = 1, \ldots, K$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \frac{1}{K} \nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{ARoW}}(\boldsymbol{\theta}; \{(\boldsymbol{x}_{t,b,k}, y_{t,b,k})\}_{k=1}^{K}, \lambda, \alpha)$, where $\mathcal{R}_{\text{ARoW}}$ is (6).
  end for
end for
Return $f_{\boldsymbol{\theta}}$

3.2 ALGORITHM

By replacing the indicator functions in Theorem 1 by their smooth proxies, we propose a new regularized risk and develop the corresponding adversarial learning algorithm called the Anti-Robust Weighted Regularization (ARoW) algorithm. The four indicator functions in (4) are replaced by

- the adversarial example $z(\boldsymbol{x})$ is replaced by $\widehat{\boldsymbol{x}}^{\text{pgd}}$ obtained by the PGD algorithm with the KL divergence;
- the term $\mathbb{1}(Y \neq F_{\boldsymbol{\theta}}(\mathbf{X}))$ is replaced by the label smooth cross-entropy (Müller et al., 2019) $\ell^{\text{LS}}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = -\boldsymbol{y}_{\alpha}^{\text{LS}}{}^{\top} \log \mathbf{p}_{\theta}(\cdot|\boldsymbol{x})$ for a given $\alpha > 0$, where $\boldsymbol{y}_{\alpha}^{\text{LS}} = (1-\alpha)\mathbf{u}_y + \frac{\alpha}{C}\mathbf{1}_C$, $\mathbf{u}_y \in \mathbb{R}^C$ is the one-hot vector whose the y-th entry is 1 and $\mathbf{1}_C \in \mathbb{R}^C$ is the vector whose entries are all 1;
- the term $\mathbb{1}(F_{\boldsymbol{\theta}}(\mathbf{X}) \neq F_{\boldsymbol{\theta}}(z(\mathbf{X})))$ is replaced by $\lambda \cdot \mathrm{KL}(\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\mathbf{X}) || \mathbf{p}_{\boldsymbol{\theta}}(\cdot|\widehat{\mathbf{X}}^{\text{pgd}}))$ for $\lambda > 0$;
- the term $\mathbb{1}\left\{p_{\boldsymbol{\theta}}(Y|z(\mathbf{X})) < 1/2\right\}$ is replaced by its convex upper bound $2(1 - p_{\boldsymbol{\theta}}(Y|\widehat{\mathbf{X}}^{\text{pgd}}))$;

to have the following regularized risk for ARoW, which is a smooth surrogate of the upper bound (4),

$$\begin{aligned} &\mathcal{R}_{\text{ARoW}}(\boldsymbol{\theta}; \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}, \lambda) \\ &:= \sum_{i=1}^{n} \left\{ \ell^{\text{LS}}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i) + 2\lambda \cdot \mathrm{KL}(\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}_i) || \mathbf{p}_{\boldsymbol{\theta}}(\cdot|\widehat{\boldsymbol{x}}_i^{\text{pgd}})) \cdot (1 - p_{\boldsymbol{\theta}}(y_i|\widehat{\boldsymbol{x}}_i^{\text{pgd}})) \right\}. \end{aligned} \tag{6}$$

Here, we introduce the regularization parameter $\lambda > 0$ to control the robustness of a trained prediction model to adversarial attacks. That is, the regularized risk (6) can be considered as a smooth surrogate of the regularized robust risk of $\mathcal{R}_{\text{nat}}(\boldsymbol{\theta}) + \lambda \mathcal{R}_{\text{bdy}}(\boldsymbol{\theta})$.

We use the label smoothing cross-entropy as a surrogate for $\mathbb{1}(Y \neq F_{\boldsymbol{\theta}}(\mathbf{X}))$ instead of the standard cross-entropy to estimate the conditional class probabilities $\mathbf{p}_{\theta}(\cdot|\boldsymbol{x})$ more accurately (Müller et al., 2019). The accurate estimation of $\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})$ is important since it is used in the regularization term of ARoW. It is well known that DNNs trained by minimizing the cross-entropy are poorly calibrated (Guo et al., 2017), and so we use the label smoothing cross-entropy technique. We set $\alpha = 0.2$ in our numerical studies for simplicity even if it can be tuned optimally.

The ARoW algorithm, which learns $\boldsymbol{\theta}$ by minimizing $\mathcal{R}_{\text{ARoW}}(\boldsymbol{\theta}; \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}, \lambda)$, is summarized in Algorithm 1.

Comparison to TRADES A key difference of the regularized risks of ARoW and TRADES is that TRADES does not have the term $(1 - p_{\boldsymbol{\theta}}(y_i|\widehat{\boldsymbol{x}}_i^{\text{pgd}}))$ at the last part of (6). That is, ARoW puts more regularization to samples which are vulnerable to adversarial attacks (i.e. $p_{\boldsymbol{\theta}}(y_i|\widehat{\boldsymbol{x}}_i^{\text{pgd}})$ is small). Note that this term is motivated by the tighter upper bound of the robust risk (4) and thus is expected to lead better results. Numerical studies confirm that it really works.

Comparison to MART Although the objective function in MART (3) has no theoretical basis, it is similar with the objective function of ARoW. But, there are two main differences. First, the

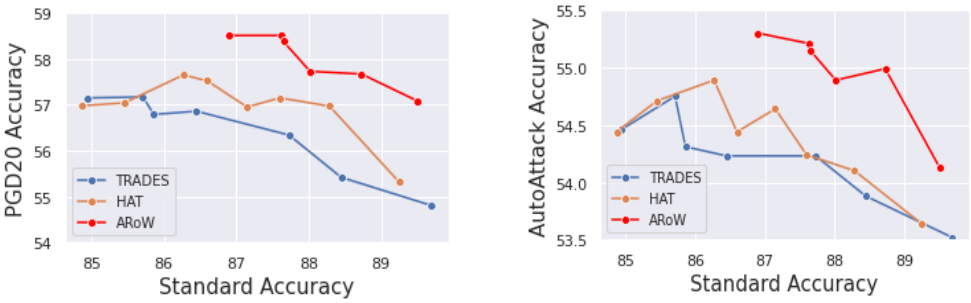

Figure 1: **Comparison of ARoW, TRADES and HAT with varying** λ. The x-axis and y-axis are the standard and robust accuracies, respectively. The robust accuracies in the left panel are against PGD^{20} while the robust accuracies in the right panel are against AutoAttack. We exclude results of MART from the figures because its performance is too low.

supervised loss term of ARoW is the label smoothing loss with clean examples, whereas MART uses the margin cross entropy loss with adversarial examples. Second, regularization term in MART is proportional to $(1 - p_{\boldsymbol{\theta}}(y|\boldsymbol{x}))$ while that in ARoW is proportional to $(1 - p_{\boldsymbol{\theta}}(y|\widehat{\boldsymbol{x}}^{\text{pgd}}))$. In the numerical studies, we observe that ARoW outperforms MART with large margins. These would be partly because ARoW is theoretically well motivated.

4 Experiments

In this section, we investigate the ARoW algorithm in view of robustness and generalization by analyzing the three benchmark data sets - CIFAR10 (Krizhevsky, 2009) , F-MINST (Xiao et al., 2017) and SVHN dataset (Netzer et al., 2011). In particular, we show that ARoW is superior to existing algorithms including TRADES (Zhang et al., 2019), HAT (Rade & Moosavi-Dezfolli, 2022) and MART (Wang et al., 2020) as well as PGD-AT (Madry et al., 2018) and GAIR-AT (Zhang et al., 2021) to achieve state-of-art performances. WideResNet-34-10 (WRN-34-10) (Zagoruyko & Komodakis, 2016) and ResNet-18 (He et al., 2016) are used for CIFAR10 while ResNet-18 (He et al., 2016) is used for F-MNIST and SVHN. Experimental details are presented in Appendix B.

4.1 Comparison of ARoW to TRADES, HAT and MART

We compare ARoW to the the regularization algorithms TRADES (Zhang et al., 2019), HAT (Rade & Moosavi-Dezfolli, 2022) and MART (Wang et al., 2020) explained in Section 2.3.2. Table 1 shows that ARoW outperforms the other regularization algorithms for various data sets and architectures in terms of both the standard and robust accuracies.

To investigate whether ARoW dominates its competitors uniformly with respect to the regularization parameter λ, we compare the trade-off between the standard and robust accuracies of ARoW and other regularization algorithms when λ varies. Figure 1 draws the plots of the standard accuracies in the x-axis and the robust accuracies in the y-axis obtained by the corresponding algorithms with various values of λ. For this experiment, we use CIFAR10 and WideResNet-34-10 (WRN-34-10) architecture.

The trade-off between the standard and robust accuracies is well observed (i.e. a larger regularization parameter λ yields lower standard accuracy but higher robust accuracy). Moreover, we can clearly see that ARoW uniformly dominates TRADES and HAT (and MART) regardless of the choice of the regularization parameter and the methods for adversarial attack.

4.2 Comparison of ARoW to PGD-AT and GAIR-AT

We compare ARoW with PGD-AT (Madry et al., 2018) and GAIR-AT (Zhang et al., 2021) which are the algorithms minimizing the robust risk directly. Table 2 shows that ARoW outperforms PGD-AT and GAIR-AT in terms of the standard accuracy and the robust accuracy against to AutoAttack (Croce & Hein, 2020b). GAIR-AT is, however, better for PGD^{20} attack than ARoW. This would be mainly because of the gradient masking (Papernot et al., 2018; 2017) - PGD does not find an adversarial example well. See Appendix C for details about gradient masking

Table 1: **Comparison of ARoW, TRADES and HAT.** We conduct the experiment three times with different seeds and present the averages of the accuracies with the standard errors in the brackets.

Method	CIFAR10 (WRN-34-10)			CIFAR10 (ResNet-18)		
	Stand	**PGD^{20}**	**AA**	**Stand**	**PGD^{20}**	**AA**
TRADES	85.86(0.09)	56.79(0.08)	54.31(0.08)	82.41(0.07)	52.68(0.22)	49.63(0.25)
HAT	86.98(0.10)	56.81(0.17)	54.63(0.07)	83.05(0.03)	52.91(0.08)	49.60(0.02)
MART	84.69(0.18)	55.67(0.13)	50.95(0.09)	74.87(0.95)	53.68(0.30)	49.61(0.24)
ARoW	**87.65**(0.02)	**58.38**(0.09)	**55.15**(0.14)	**85.30**(0.13)	**53.80**(0.16)	**49.66**(0.18)
Method	SVHN (ResNet-18)			FMNIST (ResNet-18)		
	Stand	**PGD^{20}**	**AA**	**Stand**	**PGD^{20}**	**AA**
TRADES	91.62(0.49)	58.75(0.19)	51.06(0.93)	91.92(0.04)	88.33(0.03)	88.19(0.04)
HAT	91.72(0.12)	58.66(0.06)	51.67(0.12)	92.10(0.11)	88.09(0.16)	87.93(0.13)
MART	91.64(0.41)	60.57(0.27)	49.95(0.42)	92.14(0.05)	88.10(0.10)	87.88(0.14)
ARoW	**92.79**(0.24)	**61.14**(0.74)	**51.93**(0.33)	**92.26**(0.05)	**88.73**(0.03)	**88.54**(0.04)

Table 2: **Comparison of ARoW to PGD-AT and GAIR-AT.** We conduct the experiment three times with different seeds and present the averages of the accuracies with the standard errors in the brackets.

Method	CIFAR10 (WRN-34-10)			CIFAR10 (ResNet-18)		
	Stand	**PGD^{20}**	**AA**	**Stand**	**PGD^{20}**	**AA**
PGD-AT	87.02(0.20)	57.50(0.12)	53.98(0.14)	82.42(0.05)	53.48(0.11)	49.30(0.07)
GAIR-AT	85.44(0.10)	67.27(0.07)	46.41(0.07)	81.09(0.12)	64.89(0.04)	41.35(0.16)
ARoW	**87.65**(0.02)	**58.38**(0.09)	**55.15**(0.14)	**85.30**(0.13)	**53.80**(0.16)	**49.66**(0.18)
Method	SVHN (ResNet-18)			FMNIST (ResNet-18)		
	Stand	**PGD^{20}**	**AA**	**Stand**	**PGD^{20}**	**AA**
PGD-AT	92.75(0.04)	59.05(0.46)	47.66(0.52)	92.25(0.06)	87.43(0.03)	87.19(0.03)
GAIR-AT	91.95(0.40)	70.29(0.18)	38.26(0.48)	90.96(0.10)	87.25(0.01)	85.00(0.12)
ARoW	**92.79**(0.24)	**61.14**(0.74)	**51.93**(0.33)	**92.26**(0.05)	**88.73**(0.03)	**88.54**(0.04)

4.3 Analysis with Extra Data

For improving performance on CIFAR10, Carmon et al. (2019) and Rebuffi et al. (2021) use extra unlabeled data sets with TRADES. Carmon et al. (2019) uses an additional subset of 500K extracted from 80 Million Tiny Images (80M-TI) and Rebuffi et al. (2021) uses a data set of 1M synthetic samples generated by a denoising diffusion probabilistic model (DDPM) (Ho et al., 2020) along with the SiLU activation function and Exponential Moving Average (EMA). Further, Rade & Moosavi-Dezfolli (2022) shows that HAT achieves the SOTA performance for these extra data.

Table 3 compares ARoW with the exiting algorithms for extra data, which shows that ARoW achieves the state-of-the-art performance when extra data are available.

Table 3: **Comparison of ARoW to other adversarial algorithms with extra data on CIFAR10.**

Model	Extra data	Method	Stand	PGD^{20}	AutoAttack
WRN-28-10	80M-TI(500K)	Carmon et al. (2019)	89.69	62.95	59.58
		Rebuffi et al. (2021)	90.47	63.06	60.57
		HAT	91.50	63.42	**60.96**
		ARoW	**91.57**	**64.64**	60.91
ResNet-18	80M-TI(500K)	Carmon et al. (2019)	87.07	56.86	53.16
		Rebuffi et al. (2021)	87.67	59.20	56.24
		HAT	88.98	59.29	56.40
		ARoW	**89.04**	**60.38**	**56.54**
	DDPM(1M)	Carmon et al. (2019)	82.61	56.16	52.82
		Rebuffi et al. (2021)	83.46	56.89	54.22
		HAT	86.09	58.61	55.44
		ARoW	**86.72**	**59.50**	**55.57**

4.4 ABLATION STUDIES

We study the following three issues - (i) the effect of label smoothing to ARoW, (ii) the role of the new regularization term in ARoW to improve robustness and (iii) modifications of ARoW by applying tools which improve existing adversarial training algorithms.

4.4.1 EFFECT OF LABEL SMOOTHING

Table 4 indicates that label smoothing is helpful not only for ARoW but also for TRADES. This would be partly because the regularization terms in ARoW and TRADES depend on the conditional class probabilities and it is well known that label smoothing is helpful for the calibration of the conditional class probabilities (Pereyra et al., 2017).

Moreover, the results in Table 4 imply that label smoothing is not a main reason for ARoW to outperform TRADES. Even without label smoothing, ARoW is still superior to TRADES (even with the label smoothing). Appendix E.1 presents the results of an additional experiment to assess the effect of label smoothing to the performance.

Table 4: **Comparison of TRADES and ARoW with/without label smoothing.** With WRN-28-10 architecture and CIFAR10 dataset, we use $\lambda = 6$ for TRADES while use $\lambda = 3$ for ARoW.

Method	Standard	PGD^{20}	AutoAttack
TRADES w/o-LS	85.86(0.09)	56.79(0.08)	54.31(0.08)
TRADES w/-LS	86.33(0.08)	57.45(0.02)	54.66(0.08)
ARoW w/o-LS	86.83(0.16)	58.34(0.09)	55.01(0.10)
ARoW w/-LS	**87.65**(0.02)	**58.38**(0.09)	**55.15**(0.14)

4.4.2 ROLE OF THE NEW REGULARIZATION TERM IN ARoW

The regularization term of ARoW puts more regularization to less robust samples, and thus we expect that ARoW improves the robustness of less robust samples much. To confirm this conjecture, we do a small experiment.

First, we divide the test data into four groups - least robust, less robust, robust and highly robust according to the values of $p_{\boldsymbol{\theta}_{\text{PGD}}}(y_i|\widehat{\boldsymbol{x}}_i^{\text{pgd}})$ (< 0.3, $0.3 \sim 0.5$, $0.5 \sim 0.7$ and > 0.7), where $\boldsymbol{\theta}_{\text{PGD}}$ is the parameter learned by PGD-AT (Madry et al., 2018)[1].

Then, for each group, we check how many samples become robust for ARoW and TRADES, respectively, whose results are presented in Table 5. Note that ARoW improves the robustness of least robust samples most compared with TRADES. We believe that this improvement is due to the regularization term in ARoW that enforces more regularization on less robust samples.

Table 5: **Role of the new regularization term in ARoW.** # $\textbf{Rob}_{\text{TRADES}}$ and # $\textbf{Rob}_{\text{ARoW}}$ represent the number of samples which are only robust to TRADES but not to ARoW, or vice versa. **Diff.** and **Rate of Impro.** denote (# $\textbf{Rob}_{\text{ARoW}}$ - # $\textbf{Rob}_{\text{TRADES}}$) and **Diff.** / # $\textbf{Rob}_{\text{TRADES}}$)

Sample's Robustness	# $\textbf{Rob}_{\text{TRADES}}$	# $\textbf{Rob}_{\text{ARoW}}$	Diff.	Rate of Impro. (%)
Least Robust	317	357	40	12.62
Less Robust	945	1008	63	6.67
Robust	969	1027	58	5.99
Highly Robust	3524	3529	5	0.142

4.4.3 MODIFICATIONS OF ARoW

There are many useful tools which improve existing adversarial training algorithms. Examples are Adversarial Weight Perturbation (AWP) (Wu et al., 2020) and Friendly Adversarial Training (FAT) (Zhang et al., 2020). AWP is a tool to find a flat minimum of the objective function and FAT uses early-stopped PGD when generating adversarial examples in the training phase. Details about AWP and FAT are given in Appendix E.3.

[1]We use PGD-AT instead of a standard non-robust training algorithm since all samples become least robust for a non-robust prediction model.

We investigate how ARoW performs when it is modified by such a tool. We consider the two modifications of ARoW - ARoW-AWP and ARoW-FAT, where ARoW-AWP searches a flat minimum of the ARoW objective function and ARoW-FAT uses early-stopped PGD in the training phase of ARoW.

Table 6 compares ARoW-AWP and ARoW-FAT to TRDAES-AWP and TRADES-FAT. Both of AWP and FAT are helpful for ARoW and TRADES but ARoW still outperforms TRADES with large margins even after modified by AWP or FAT.

Table 6: **Modifications of TRADES and ARoW.** We use CIFAR10 dataset and ResNet-18 architecture. More details of hyerparameters are provided in Appendix E.3.

Method	AWP			FAT		
	Standard	**PGD^{20}**	**AutoAttack**	**Standard**	**PGD^{20}**	**AutoAttack**
TRADES	82.10(0.09)	53.56(0.18)	49.56(0.23)	82.96(0.08)	52.76(0.22)	49.83(0.28)
ARoW	**84.98**(0.11)	**55.55**(0.15)	**50.64**(0.18)	**86.21**(0.06)	**53.37**(0.20)	**50.07**(0.17)

4.5 Improved Fairness

Xu et al. (2021) reports that TRADES (Zhang et al., 2019) increases the variation of the per-class accuracies (accuracy in each class) which is not desirable in view of fairness. In turn, Xu et al. (2021) proposes the Fair-Robust-Learning (FRL) algorithm to alleviate this problem. Even if fairness becomes improved, the standard and robust accuracies of FRL are worse than TRADES.

In contrast, Table 7 shows that ARoW improves the fairness as well as the standard and robust accuracies compared to TRADES. This desirable property of ARoW can be partly understood as follows. The main idea of ARoW is to impose more robust regularization to less robust samples. In turn, samples in less accurate classes tend to be more vulnerable to adversarial attacks. Thus, ARoW improves the robustness of samples in less accurate classes which results in improved robustness as well as improved generalization for such less accurate classes. The class-wise accuracies are presented in Appendix F.

Table 7: **Class-wise accuracy disparity for CIFAR10**. We report the accuracy (ACC), the worst-class accuracy (WC-Acc) and the standard deviation of class-wise accuracies (SD) for each method.

Method	**Standard**			**PGD^{10}**		
	Acc	**WC-Acc**	**SD**	**Acc**	**WC-Acc**	**SD**
TRADES	85.69	67.10	9.27	57.38	27.10	16.97
ARoW	**87.58**	**74.51**	**7.11**	**59.32**	**31.05**	**15.67**

5 Conclusion and Future Works

In this paper, we derived an upper bound of the robust risk and developed a new algorithm for adversarial training called ARoW which minimizes a surrogate version of the derived upper bound. A novel feature of ARoW is to impose more regularization on less robust samples than TRADES. The results of numerical experiments shows that ARoW improves the standard and robust accuracies simultaneously to achieve state-of-the-art performances. In addition, ARoW enhances the fairness of the prediction model without hampering the accuracies.

When we developed a computable surrogate of the upper bound of the robust risk in Theorem 1, we replaced $\mathbb{1}(F_{\boldsymbol{\theta}}(\mathbf{X}) \neq F_{\boldsymbol{\theta}}(z(\mathbf{X})))$ by $\mathrm{KL}(\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\mathbf{X})||\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\widehat{\mathbf{X}}^{\mathrm{pgd}}))$. The KL divergence, however, is not an upper bound of the 0-1 loss and thus our surrogate is not an upper bound of the robust risk. We employed the KL divergence surrogate to make the objective function of ARoW be similar to that of TRADES. It would be worth pursuing to devise an alternative surrogate for the 0-1 loss to reduce the gap between the theory and algorithm.

We have seen in Section 4.5 that ARoW improves fairness as well as accuracies. The advantage of ARoW in view of fairness is an unexpected by-product, and it would be interesting to develop a more principled way of enhancing the fairness further without hampering the accuracy.

REFERENCES

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random searchg. *In ECCV*, 2020.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. *IEEE Symposium on Security and Privacy*, 2017.

Yair Carmon, Aditi Raghunathan, Schmidt Ludwig, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *In NeurIPS*, 2019.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. *In ACM*, 2017. doi: 10.1145/3128572.3140448. URL `http://dx.doi.org/10.1145/3128572.3140448`.

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. *In ICLR*, 2021.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. *In The European Conference on Computer Vision(ECCV)*, 2020a.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, 2020b.

Yao Deng, Xi Zheng, Tianyi Zhang, Chen Chen, Guannan Lou, and Miryung Kim. An analysis of adversarial attacks and defenses on autonomous driving models. *IEEE International Conference on Pervasive Computing and Communications(PerCom)*, 2020.

Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. *In International Conference on Learning Representataions(ICLR)*, 2020.

Samuel G. Finlayson, Hyung Won Chung, Isaac S. Kohane, and Andrew L. Beam. Adversarial attacks against medical deep learning systems. *In Science*, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *In ICLR*, 2015.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *In ICML*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *In CVPR*, 2016.

Dorjan Hitaj, Giulio Pagnotta, Iacopo Masi, and Luigi V. Mancini. Evaluating the robustness of geometry-aware instance-reweighted adversarial training. *archive*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *In NeurIPS*, 2020.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *In ICML*, 2018.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *Proceedings of the international conference on Uncertainty in Artificial Intelligence*, 2018.

Linxi Jiang, Xingjun Ma, Shaoxiang Chen, James Bailey, and Yu-Gang Jiang. Black-box adversarial attacks on video recognition models. *In ACM*, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009.

Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *In ICLR*, 2017.

Yujie Li, Xing Xu, Jinhui Xiao, Siyuan Li, and Heng Tao Shen. Adaptive square attack: Fooling autonomous cars with adversarial traffic signs. *IEEE Internet of Things Journal*, 2020.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *In ICLR*, 2017.

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

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *In ICLR*, 2018.

Nir Morgulis, Alexander Kreines, Shachar Mendelowitz, and Yuval Weisglass. Fooling a real car with adversarial traffic signs. *ArXiv*, 2019.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? *In NeurIPS*, 2019.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv*, 2016.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. *In ACM*, 2017.

Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2018.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. *In ICLR*, 2017.

Rahul Rade and Seyed-Mohsen Moosavi-Dezfolli. Recuding excessive margin to achieve a better accuracy vs. robustness trade-off. *In ICLR*, 2022.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A Mann. Data augmentation can improve robustness. *In NeurIPS*, 2021.

Leslie Rice, Ericg Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning. *In ICML*, 2020.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *In ICLR*, 2014.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. *In International Conference on Learning Representataions(ICLR))*, 2020.

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

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *archive*, 2017.

Han Xu, Xiaorui Liu, Yaxin Li, Anil K. Jain, and Jiliang Tang. To be robust or to be fair: Towards fairness in adversarial training. *In ICML*, 2021.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *In CVPR*, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *Proceedings of the British Machine Vision Conference 2016*, 2016.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *In International Conference on Machine Learning (ICML)*, 2019.

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

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

Appendices

A PROOF OF THEOREM 1

In this section, we prove Theorem 1. The following lemma provides the key inequality for the proof.

Lemma 2. *For a given score function $f_{\boldsymbol{\theta}}$, let $z(\cdot)$ be an any measurable mapping from $\mathcal{X}$ to $\mathcal{X}$ satisfying*

$$z(\boldsymbol{x}) \in \underset{\boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x}, \varepsilon)}{\arg\max} \mathbb{1}\left(F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$$

for every $\boldsymbol{x} \in \mathcal{X}$. Then, we have

$$\begin{aligned} &\mathbb{1}\left\{\exists \boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x}, \varepsilon) : F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(\boldsymbol{x}'), F_{\boldsymbol{\theta}}(\boldsymbol{x}) = Y\right\} \\ &\qquad \leq \mathbb{1}\left\{F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(z(\boldsymbol{x})), Y \neq F_{\boldsymbol{\theta}}(z(\boldsymbol{x}))\right\} \end{aligned} \tag{A.1}$$

Proof. The inequality holds obviously if $\mathbb{1}\left\{F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(z(\boldsymbol{x})), Y \neq F_{\boldsymbol{\theta}}(z(\boldsymbol{x}))\right\} = 1$. Hence, it suffices to show that $\mathbb{1}\left\{\exists \boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x}, \varepsilon) : F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(\boldsymbol{x}'), F_{\boldsymbol{\theta}}(\boldsymbol{x}) = Y\right\} = 0$ when either $F_{\boldsymbol{\theta}}(\boldsymbol{x}) = F_{\boldsymbol{\theta}}(z(\boldsymbol{x}))$ or $Y = F_{\boldsymbol{\theta}}(z(\boldsymbol{x}))$.

Suppose $F_{\boldsymbol{\theta}}(\boldsymbol{x}) = F_{\boldsymbol{\theta}}(z(\boldsymbol{x}))$. It trivially holds that $\mathbb{1}\left(F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(z(\boldsymbol{x}))\right) \leq \mathbb{1}\left(F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$ for every $\boldsymbol{x}' \in \mathcal{X}$ since $\mathbb{1}\left(F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(z(\boldsymbol{x}))\right) = 0$ and the equality holds if and only if $F_{\boldsymbol{\theta}}(z(\boldsymbol{x})) = F_{\boldsymbol{\theta}}(\boldsymbol{x}')$. By the definition of $z(\boldsymbol{x})$, the left side of (A.1) is 0 since $\mathbb{1}\left\{\exists \boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x}, \varepsilon) : F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(\boldsymbol{x}')\right\} = 0$, and hence the inequality holds.

Suppose $Y = F_{\boldsymbol{\theta}}(z(\boldsymbol{x}))$. If $F_{\boldsymbol{\theta}}(\boldsymbol{x}) = Y$ and there exists $\boldsymbol{x}'$ in $\mathcal{B}_p(\boldsymbol{x}, \varepsilon)$ such that $F_{\boldsymbol{\theta}}(\boldsymbol{x}') \neq F_{\boldsymbol{\theta}}(\boldsymbol{x})$, then we have $F_{\boldsymbol{\theta}}(\boldsymbol{x}') \neq Y = F_{\boldsymbol{\theta}}(\boldsymbol{x}) = F_{\boldsymbol{\theta}}(z(\boldsymbol{x}))$. In turn, it implies $\mathbb{1}\left(F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(z(\boldsymbol{x}))\right) < \mathbb{1}\left(F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$, which is a contradiction to the definition of $z(\boldsymbol{x})$. Hence, the left side of (A.1) should be 0, and we complete the proof of the inequality. □

Theorem 1. *For a given score function $f_{\boldsymbol{\theta}}$, let $z(\cdot)$ be an any measurable mapping from $\mathcal{X}$ to $\mathcal{X}$ satisfying*

$$z(\boldsymbol{x}) \in \underset{\boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x}, \varepsilon)}{\arg\max} \mathbb{1}\left(F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(\boldsymbol{x}')\right).$$

for every $\boldsymbol{x} \in \mathcal{X}$. Then, we have

$$\mathcal{R}_{rob}(\boldsymbol{\theta}) \leq \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}(Y \neq F_{\boldsymbol{\theta}}(\mathbf{X})) + \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}(F_{\boldsymbol{\theta}}(\mathbf{X}) \neq F_{\boldsymbol{\theta}}(z(\mathbf{X})))\mathbb{1}\left\{p_{\boldsymbol{\theta}}(Y|z(\mathbf{X})) < 1/2\right\} \tag{4}$$

Proof. Note that $\mathcal{R}_{\text{rob}}(\boldsymbol{\theta}) = \mathcal{R}_{\text{nat}}(\boldsymbol{\theta}) + \mathcal{R}_{\text{bdy}}(\boldsymbol{\theta})$ where $\mathcal{R}_{\text{nat}}(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}\left\{F_{\boldsymbol{\theta}}(\mathbf{X}) \neq Y\right\}$ and $\mathcal{R}_{\text{bdy}}(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}\left\{\exists \mathbf{X}' \in \mathcal{B}_p(\mathbf{X}, \varepsilon) : F_{\boldsymbol{\theta}}(\mathbf{X}) \neq F_{\boldsymbol{\theta}}(\mathbf{X}'), F_{\boldsymbol{\theta}}(\mathbf{X}) = Y\right\}$.

Since

$$\begin{aligned} \mathcal{R}_{\text{bdy}}(\boldsymbol{\theta}) &= \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}\left\{\exists \mathbf{X}' \in \mathcal{B}_p(\mathbf{X}, \varepsilon) : F_{\boldsymbol{\theta}}(\mathbf{X}) \neq F_{\boldsymbol{\theta}}(\mathbf{X}'), F_{\boldsymbol{\theta}}(\mathbf{X}) = Y\right\} \\ &\leq \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}\left\{F_{\boldsymbol{\theta}}(\mathbf{X}) \neq F_{\boldsymbol{\theta}}(z(\mathbf{X})), Y \neq F_{\boldsymbol{\theta}}(z(\mathbf{X}))\right\} \ (\because \text{ Lemma 2}) \\ &= \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}\left\{F_{\boldsymbol{\theta}}(\mathbf{X}) \neq F_{\boldsymbol{\theta}}(z(\mathbf{X}))\right\}\mathbb{1}\left\{Y \neq F_{\boldsymbol{\theta}}(z(\mathbf{X}))\right\} \\ &\leq \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}\left\{F_{\boldsymbol{\theta}}(\mathbf{X}) \neq F_{\boldsymbol{\theta}}(z(\mathbf{X}))\right\}\mathbb{1}\left\{p_{\boldsymbol{\theta}}(Y|z(\mathbf{X})) < 1/2\right\}, \end{aligned}$$

the inequality (4) holds. □

B DETAILED SETTINGS FOR THE EXPERIMENTS WITH BENCHMARK DATASETS

B.1 EXPERIMENTAL SETUP

For CIFAR10, SVHN and FMNIST datasets, input images are normalized into [0, 1]. Random crop and random horizontal flip with probability 0.5 are used for CIFAR10 while only random horizontal flip with probability 0.5 is applied for SVHN. For FMNIST, augmentation is not used.

For generating adversarial examples in the training phase, PGD^{10} with random initial, $p = \infty$, $\varepsilon = 8/255$ and $\nu = 2/255$ is used, where PGD^T is the output of the PGD algorithm (2) with T iterations. For training prediction models, the SGD with momentum 0.9, weight decay 5×10^{-4}, the initial learning rate of 0.1 and batch size of 128 is used and the learning rate is reduced by a factor of 10 at 60 and 90 epochs. Stochastic weighting average (SWA) (Izmailov et al., 2018) is employed after 50-epochs for preventing from robust overfitting (Rice et al., 2020) as Chen et al. (2021) does.

For evaluating the robustness in the test phase, PGD^{20} and AutoAttack are used for adversarial attacks, where AutoAttack consists of three white box attacks - APGD and APGD-DLR in Croce & Hein (2020b) and FAB in Croce & Hein (2020a) and one black box attack - Square Attack (Andriushchenko et al., 2020). To the best of our knowledge, AutoAttack is the strongest attack. The final model is set to be the best model against PGD^{10} on the test data among those obtained until 120 epochs.

B.2 HYPERPARAMETER SETTING

Table 8: **Selected hyperparameters.** Hyperparameters used in the numerical studies in Section 4.1 and Section 4.2.

Dataset	Model	**Method**	λ	γ	Weight Decay	α	SWA
CIFAR10	WRN-34-10	TRADES	6	-	$5e^{-4}$	-	o
		HAT	4	0.25	$5e^{-4}$	-	o
		MART	6	-	$2e^{-4}$	-	x
		PGD-AT	-	-	$5e^{-4}$	-	o
		GAIR-AT	-	-	$5e^{-4}$	-	o
		ARoW	3	-	$5e^{-4}$	0.2	o
	ResNet-18	TRADES	6	-	$5e^{-4}$	-	o
		HAT	4	0.5	$5e^{-4}$	-	o
		MART	6	-	$5e^{-4}$	-	x
		PGD-AT	-	-	$5e^{-4}$	-	o
		GAIR-AT	-	-	$5e^{-4}$	-	o
		ARoW	2.5	-	$5e^{-4}$	0.2	o
SVHN	ResNet-18	TRADES	6	-	$5e^{-4}$	-	x
		HAT	4	0.5	$5e^{-4}$	-	x
		MART	6	-	$5e^{-4}$	-	x
		PGD-AT	-	-	$5e^{-4}$	-	x
		GAIR-AT	-	-	$5e^{-4}$	-	x
		ARoW	3	-	$5e^{-4}$	0.2	x
FMNIST	ResNet-18	TRADES	6	-	$5e^{-4}$	-	x
		HAT	5	0.15	$5e^{-4}$	-	x
		MART	6	-	$5e^{-4}$	-	x
		PGD-AT	-	-	$5e^{-4}$	-	x
		GAIR-AT	-	-	$5e^{-4}$	-	x
		ARoW	6	-	$5e^{-4}$	0.2	x

Table 8 presents the hyperparameters used on our experiments. Most of the hyperparameters are set to be the ones used in the previous studies. The weight decay parameter is set to be $5e^{-4}$ in most experiments, which is the well-known optimal value. Only for MART (Wang et al., 2020) with WRN34-10, we use weight decay $2e^{-4}$ as Wang et al. (2020) did since MART works poorly with $5e^{-4}$. We use stochastic weight averaging (SWA) for CIFAR10 except MART. Note that SWA is not

used in the experiments of Wang et al. (2020), and we confirm that SWA is not helpful for MART. Effects of SWA for all methods are provided in Appendix E.2.

C Checking the Gradient Masking

Table 9: **Comparison of GAIR-AT and ARoW**. We compare the robustness of GAIR-AT (Zhang et al., 2021) and ARoW against the four attacks used in AutoAttack on CIFAR10. The results are based on WRN-34-10. We set $\lambda = 3$ for ARoW.

Method	**Standard**	**PGD**	**APGD**	**APGD-DLR**	**FAB**	**SQUARE**
GAIR-AT	85.44(0.170)	**67.27**(0.07)	**63.14**(0.16)	46.48(0.07)	49.35(0.05)	55.19(0.16)
ARoW	**87.65**(0.02)	58.38(0.09)	56.07(0.14)	**55.17**(0.11)	**56.69**(0.17)	**63.50**(0.08)

Gradient masking (Papernot et al., 2018; 2017) is the case that the gradient of the loss for a given non-robust datum is almost zero (i.e. $\nabla_{\boldsymbol{x}} \ell_{\text{ce}}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y) \approx \mathbf{0}$). In this case, PGD cannot generate an adversarial example. We can check the ocuurence of gradient masking when a prediction model is robust to the PGD attack but not robust to attacks such as FAB (Croce & Hein, 2020a), APGD-DLR (Croce & Hein, 2020b) and SQUARE (Andriushchenko et al., 2020).

In Table 9, the robustness of GAIR-AT becomes worse much for the three attacks in AutoAttack except APGD (Croce & Hein, 2020b) while the robustness of ARoW remains stable regardless of the adversarial attacks. Since APGD uses the gradient of the loss, this observation implies that the gradient masking occurs in GAIR-AT while it does not in ARoW.

Better performance of GAIR-AT for PGD^{20} attack in Table 2 is not because GAIR-AT is robust to adversarial attacks but because adversarial examples obtained by PGD are close to clean samples. This claim is supported by the fact that GAIR-AT performs poorly for AutoAttack while it is still robust to other PGD-based adversarial attacks. Moreover, gradient masking for GAIR-AT is already reported by Hitaj et al. (2021).

D Detailed Setting for the Experiments with Extra Data

Table 10: **Selected hyperparameters.** Hyperparameters used in the numerical studies in Section 4.3. We do not employ cutmix augmentation (Yun et al., 2019) as does in Rade & Moosavi-Dezfolli (2022).

Model	**Method**	λ	γ	Weight Decay	α	EMA	SiLU
WRN-28-10	Carmon et al. (2019)	6	-	$5e^{-4}$	-	x	x
	Rebuffi et al. (2021)	6	-	$5e^{-4}$	-	o	o
	HAT	4	0.25	$5e^{-4}$	-	o	o
	ARoW	3.5	-	$5e^{-4}$	0.2	o	o
ResNet-18	Carmon et al. (2019)	6	-	$5e^{-4}$	-	x	x
	Rebuffi et al. (2021)	6	-	$5e^{-4}$	-	o	o
	HAT	4	0.25	$5e^{-4}$	-	o	o
	ARoW	3.5	-	$5e^{-4}$	0.2	o	o

In Section 4.3, we presented the results of ARoW on CIFAR10 with extra unlabeled data used in Carmon et al. (2019) and Rebuffi et al. (2021). In this section, we provide experimental details.

Rebuffi et al. (2021) use the SiLU activation function and exponential model averaging (EMA) based on TRADES. For HAT (Rade & Moosavi-Dezfolli, 2022) and ARoW, we use the SiLU activation function and exponential model averaging (EMA) with weight decay factor 0.995 as is done in Rebuffi et al. (2021). The cosine annealing learning rate scheduler (Loshchilov & Hutter, 2017) is used with the batch size 512. The final model is set to be the best model against PGD^{10} on the test data among those obtained until 500 epochs.

E ABLATION STUDY

E.1 THE EFFECT OF LABEL SMOOTHING

Table 11 presents the standard and robust accuracies of ARoW on CIFAR10 for various values of the smoothing parameter α in the label smoothing where the regularization parameter λ is fixed at 3 and ResNet18 is used.

Table 11: **Standard and robust accuracies of ARoW on CIFAR10 for varying α.**

α	**Standard**	**PGD^{20}**	**AutoAttack**
0.05	83.54	53.10	49.88
0.10	84.10	53.29	49.75
0.15	84.36	53.56	49.67
0.20	84.52	53.68	49.96
0.25	84.48	53.53	49.93
0.30	84.55	53.53	49.89
0.35	84.66	54.19	50.03
0.40	84.65	54.23	50.11

E.2 EFFECT OF STOCHASTIC WEIGHT AVERAGING (SWA)

We compare the standard and robust accuracies of the adversarial training algorithms with and without SWA whose results are summarized in Table 12. SWA improves the accuracies for all the algorithms except MART. Without SWA, ARoW is competitive to HAT, which is known to be the SOTA method. However, ARoW dominates HAT when SWA is applied.

Table 12: **Effects of SWA on CIFAR10 with WideResNet 34-10.** We conduct the experiment three times with different seeds and present the averages of the accuracies with the standard errors in the brackets. 'w/o' stands for 'without'.

	Method	**Standard**	**PGD^{20}**	**AutoAttack**
SWA	TRADES	85.86(0.09)	56.79(0.08)	54.31(0.08)
	HAT	86.98(0.10)	56.81(0.17)	54.63(0.07)
	MART	78.41(0.07)	56.04(0.09)	48.94(0.09)
	PGD-AT	87.02(0.20)	57.50(0.12)	53.98(0.14)
	ARoW	**87.59**(0.02)	**58.61**(0.09)	**55.21**(0.14)
w/o-SWA	TRADES	85.48(0.12)	56.06(0.08)	53.16(0.17)
	HAT	87.53(0.02)	56.41(0.09)	**53.38**(0.10)
	MART	84.69(0.18)	55.67(0.13)	50.95(0.09)
	PGD-AT	86.88(0.09)	54.15(0.16)	51.35(0.14)
	ARoW	**87.60**(0.02)	**56.47**(0.10)	52.95(0.06)

E.3 AWP AND FAT

E.3.1 ADVERSARIAL WEIGHT PERTURBATION (AWP)

For a given objective function of the adversarial training, AWP (Wu et al., 2020) tries to find a flat minimum in the parameter space. Wu et al. (2020) proposes TRADES-AWP, which minimizes

$$\min_{\boldsymbol{\theta}} \max_{\|\boldsymbol{\delta}_l\| \le \gamma \|\boldsymbol{\theta}_l\|} \frac{1}{n} \sum_{i=1}^{n} \left\{ \ell_{\text{ce}}(f_{\boldsymbol{\theta}+\boldsymbol{\delta}}(\boldsymbol{x}_i), y_i) + \lambda \cdot \text{KL}(\mathbf{p}_{\boldsymbol{\theta}+\boldsymbol{\delta}}(\cdot|\boldsymbol{x}_i) \| \mathbf{p}_{\boldsymbol{\theta}+\boldsymbol{\delta}}(\cdot|\widehat{\boldsymbol{x}}_i^{\text{pgd}})) \right\},$$

where $\boldsymbol{\theta}_l$ is the weight vector of l-th layer and γ is the weight perturbation size. Inspired by TRADES-AWP, we propose ARoW-AWP which minimizes

$$\min_{\boldsymbol{\theta}} \max_{\|\boldsymbol{\delta}_l\| \le \gamma \|\boldsymbol{\theta}_l\|} \frac{1}{n} \sum_{i=1}^{n} \Big\{ \ell_{\text{ce}}(f_{\boldsymbol{\theta}+\boldsymbol{\delta}}(\boldsymbol{x}_i), y_i) + 2\lambda \cdot \text{KL}(\mathbf{p}_{\boldsymbol{\theta}+\boldsymbol{\delta}}(\cdot|\boldsymbol{x}_i) \| \mathbf{p}_{\boldsymbol{\theta}+\boldsymbol{\delta}}(\cdot|\widehat{\boldsymbol{x}}_i^{\text{pgd}})) \cdot (1 - p_{\boldsymbol{\theta}}(y_i|\widehat{\boldsymbol{x}}_i^{\text{pgd}})) \Big\}.$$

In our experiment, we set γ to be 0.005 which is the value used in Wu et al. (2020) and do not use SWA as did in original paper.

E.3.2 Friendly Adversarial Training (FAT)

Zhang et al. (2020) suggests early-stopped PGD which uses a data-adaptive iterations of PGD when an adversarial example is generated. TRADES-FAT, which uses the early-stopped PGD in TRADES, minimizes

$$\sum_{i=1}^{n} \ell_{\text{ce}}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i) + \lambda \cdot \text{KL}(\mathbf{p}_{\theta}(\cdot|\boldsymbol{x}_i) \| \mathbf{p}_{\theta}(\cdot|\widehat{\boldsymbol{x}}_i^{(t_i)}))$$

where $t_i = \min\left\{\min\{t : F_{\boldsymbol{\theta}}(\widehat{\boldsymbol{x}}_i^{(t)}) \neq y_i\} + K, T\right\}$. Here, T is the maximum iterations of PGD.

We propose an adversarial training algorithm ARoW-FAT by combining ARoW and early-stopped PGD. ARoW-FAT minimizes the following regularized empirical risk:

$$\sum_{i=1}^{n} \left\{ \ell_{\alpha}^{\text{LS}}(f_{\theta}(\boldsymbol{x}_i), y_i) + 2\lambda \cdot \text{KL}(\mathbf{p}_{\theta}(\cdot|\boldsymbol{x}_i) \| \mathbf{p}_{\theta}(\cdot|\widehat{\boldsymbol{x}}_i^{(t_i)})) \cdot (1 - p_{\boldsymbol{\theta}}(y_i|\widehat{\boldsymbol{x}}_i^{(t_i)})) \right\}.$$

In the experiments, we set K to be 2, which is the value used in Zhang et al. (2020).

F Improved Fairness

Table 13: **Comparison of per-class robustness and generalization of TRADES and ARoW. Rob**$_{\text{TRADES}}$ and **Rob**$_{\text{ARoW}}$ are the robust accuracies against PGD^{20} of TRADES and ARoW, respectively. **Stand**$_{\text{TRADES}}$ and **Stand**$_{\text{ARoW}}$ are the standard accuracies.

Class	Rob$_{\text{TRADES}}$	Rob$_{\text{ARoW}}$	Stand$_{\text{TRADES}}$	Stand$_{\text{ARoW}}$
0(Airplane)	64.8	66.7	88.3	91.6
1(Automobile)	77.5	77.5	93.7	95.3
2(Bird)	38.5	43.1	72.5	80.6
3(Cat)	26.1	30.2	65.9	75.1
4(Deer)	35.6	40.3	83.4	87.5
5(Dog)	48.6	47.2	76.0	79.3
6(Frog)	67.8	63.6	94.2	95.2
7(Horse)	69.7	69.3	91.0	92.7
8(Ship)	62.3	70.1	90.9	94.9
9(Truck)	75.3	76.3	93.5	93.5

In Table 13, we present the per-class robust and standard accuracies of the prediction models trained by TRADES and ARoW. We can see that ARoW is highly effective for classes difficult to be classified such as Bird, Cat, Deer and Dog. For such classes, ARoW improves much not only the standard accuracies but also the robust accuracies. For example, in the class 'Cat', which is the most difficult class (the lowest standard accuarcy for TRADES and ARoW), the robustness and generalization are improved by 4.1 percentage point $(26.1\% \to 30.2\%)$ and 9.2 percentage point $(65.9\% \to 75.1\%)$ by ARoW compared with TRADES, respectively. This desirable results would be mainly due to the new regularization term in ARoW. Usually, difficult classes are less robust to adversarial attacks. By putting more regularization on less robust classes, ARoW improves the accuracies of less robust classes more.