Improving Adversarial Robustness by Putting More Regularizations on Less Robust Samples

Dongyoon Yang [1] Insung Kong [1] Yongdai Kim [1]

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 the regularized empirical risk motivated from 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 significant margins.

[1] Department of Statistics, Seoul National University, Seoul, Republic of Korea. Correspondence to: Yongdai Kim <ydkim0903@gmail.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

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 for multi-classification problems
- 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 $F_{\boldsymbol{\theta}}(\boldsymbol{x}) = \underset{c}{\mathrm{argmax}}[f_{\boldsymbol{\theta}}(\boldsymbol{x})]_c$, $\mathcal{B}_p(\boldsymbol{x}, \varepsilon) = \{\boldsymbol{x}' \in \mathcal{X} : \|\boldsymbol{x} - \boldsymbol{x}'\|_p \le \varepsilon\}$ 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

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

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) with infinite norm (Madry et al., 2018). Let $\ell(\boldsymbol{x}'|\theta, \boldsymbol{x}, y)$ 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 ℓ to update $\boldsymbol{x}'$ and projecting it to $\mathcal{B}_\infty(\boldsymbol{x}, \varepsilon)$. That is, the update rule of PGD is

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

where $\eta > 0$ is the step size, $\mathbf{\Pi}_{\mathcal{B}_\infty(\boldsymbol{x},\varepsilon)}(\cdot)$ is the projection operator to $\mathcal{B}_\infty(\boldsymbol{x}, \varepsilon)$ and $\boldsymbol{x}^{(0)} = \boldsymbol{x}$. We define $\boldsymbol{x}^{\mathrm{pgd}}$ as $\boldsymbol{x}^{\mathrm{pgd}} := \lim_{m\to\infty} \boldsymbol{x}^{(m)}$ and denote the proxy by $\widehat{\boldsymbol{x}}^{\mathrm{pgd}} = \boldsymbol{x}^{(M)}$ with finite step M. For the surrogate loss ℓ, 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 $\widehat{\boldsymbol{x}}_i^{\mathrm{pgd}}$ 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

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

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 $\kappa_\theta(\boldsymbol{x}_i, y_i) = \min\Big(\min(\{t : F_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(t)}) \neq y_i\}), T\Big)$ 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

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

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:

$$\begin{aligned}
\sum_{i=1}^{n} \Bigg\{ &\ell_{\text{ce}}\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i\right), y_i\right) + \lambda \cdot \text{KL}\left(\mathbf{p}_\theta\left(\cdot|\boldsymbol{x}_i\right) \| \mathbf{p}_\theta(\cdot|\widehat{\boldsymbol{x}}_i^{\text{pgd}})\right) \\
&+ \gamma \ell_{\text{ce}}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{\text{helper}}), F_{\boldsymbol{\theta}_{\text{pre}}}(\widehat{\boldsymbol{x}}_i^{\text{pgd}})\right) \Bigg\}, \qquad (3)
\end{aligned}$$

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

$$\begin{aligned}
\sum_{i=1}^{n} \Bigg\{ &\ell_{\text{margin}}(f_{\boldsymbol{\theta}}(\widehat{\boldsymbol{x}}_i^{\text{pgd}}), y_i) \\
&+ \lambda \cdot \text{KL}(\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}_i) \| \mathbf{p}_{\boldsymbol{\theta}}(\cdot|\widehat{\boldsymbol{x}}_i^{\text{pgd}}))(1 - p_{\boldsymbol{\theta}}(y_i|\boldsymbol{x}_i)) \Bigg\},
\end{aligned} \qquad (4)$$

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

$$z(\boldsymbol{x}) \in \operatorname*{argmax}_{\boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x}, \varepsilon)} \mathbb{1}\left(F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(\boldsymbol{x}')\right).$$

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

$$\begin{aligned}
&\mathcal{R}_{rob}(\boldsymbol{\theta}) \leq \mathbb{E}_{(\mathbf{X},Y)} \mathbb{1}(Y \neq F_{\boldsymbol{\theta}}(\mathbf{X})) \\
&\quad + \mathbb{E}_{(\mathbf{X},Y)} \mathbb{1}(F_{\boldsymbol{\theta}}(\mathbf{X}) \neq F_{\boldsymbol{\theta}}(z(\mathbf{X}))) \mathbb{1}\left\{p_{\boldsymbol{\theta}}(Y|z(\mathbf{X})) < 1/2\right\}
\end{aligned} \qquad (5)$$

The upper bound (5) 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

$$\mathcal{R}_{\text{rob}}(\boldsymbol{\theta}) \leq \mathbb{E}_{(\mathbf{X},Y)} \phi(Y f_{\boldsymbol{\theta}}(\mathbf{X})) + \mathbb{E}_{\mathbf{X}} \phi(f_{\boldsymbol{\theta}}(\mathbf{X}) f_{\boldsymbol{\theta}}(z(\mathbf{X}))), \qquad (6)$$

where

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

and $\phi(\cdot)$ is an upper bound of $\mathbb{1}(\cdot < 0)$. Our upper bound (5) is a modification of the upper bound (6) for multiclass problems where $\phi(\cdot)$ and $f_{\boldsymbol{\theta}}$ in (6) are replaced by $\mathbb{1}(\cdot < 0)$ and $F_{\boldsymbol{\theta}}$, respectively. A key difference, however, between (5) and (6) is the term $\mathbb{1}\left\{p_{\boldsymbol{\theta}}(Y|z(\mathbf{X})) < 1/2\right\}$ at the last part of (5) that is not in (6).

It is interesting to see that the upper bound in Theorem 3.1 becomes equal to the robust risk for binary classification problems. That is, the upper bound (5) is an another formulation of the robust risk. However, this rephrased formula of the robust risk is useful since it provides a new learning algorithm when the indicator functions are replaced by their surrogates as we do.

3.2. Algorithm

Algorithm 1 Anti-Robust Weighted (ARoW) Regularization

Input : network $f_{\boldsymbol{\theta}}$, training dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, learning rate η, perturbation budget ε, number of PGD steps M, hyperparameters (λ, α) of (7), 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
        for $k = 1, \cdots, K$ do
            Generate $\widehat{\boldsymbol{x}}_{b,k}^{\text{pgd}}$ using PGD$^{(M)}$ in (2)
        end for

            $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \frac{1}{K} \nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{ARoW}}(\boldsymbol{\theta}\,; \{(\boldsymbol{x}_k, y_k)\}_{k=1}^K, \lambda, \alpha)$

    end for
end for
Return $f_{\boldsymbol{\theta}}$
```

By replacing the indicator functions in Theorem 3.1 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 (5) 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 or cross entropy;
- the term $\mathbb{1}(Y \neq F_{\boldsymbol{\theta}}(\mathbf{X}))$ is replaced by the label smooth cross-entropy (Müller et al., 2019) $\ell^{\text{LS}}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = -{\boldsymbol{y}_\alpha^{\text{LS}}}^\top \log \mathbf{p}_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})$ 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 $\lambda \cdot \text{KL}(\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\mathbf{X})||\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\widehat{\mathbf{X}}^{\text{pgd}}))$ 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 (5),

$$\begin{aligned} &\mathcal{R}_{\text{ARoW}}(\boldsymbol{\theta}; \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n, \lambda) \\ &:= \sum_{i=1}^n \Bigg\{ \ell^{\text{LS}}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i) \\ &+ 2\lambda \cdot \text{KL}(\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}_i)||\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\widehat{\boldsymbol{x}}_i^{\text{pgd}})) \cdot (1 - p_{\boldsymbol{\theta}}(y_i|\widehat{\boldsymbol{x}}_i^{\text{pgd}})) \Bigg\}. \end{aligned} \tag{7}$$

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 (7) 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.

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 (7). 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 (5) and thus is expected to lead better results. Numerical studies confirm that it really works.

Comparison to MART The objective function in MART (4) is similar with the objective function of ARoW. But, there are two main differences. First, the 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, the regularization term in MART is proportional to $(1 - p_{\boldsymbol{\theta}}(y|\boldsymbol{x}))$ while that in ARoW is proportional to $(1 - p_{\boldsymbol{\theta}}(y|\widehat{\boldsymbol{x}}^{\text{pgd}}))$. Even though these two terms look similar, their roles are quite different.

Table 1. **Comparison of ARoW and Other Competitors.** 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)			CIFAR100 (WRN-34-10)		
	Stand	**PGD^{20}**	**AA**	**Stand**	**PGD^{20}**	**AA**
PGD-AT	87.02(0.20)	57.50(0.12)	53.98(0.14)	62.20(0.11)	32.27(0.05)	28.66(0.05)
GAIR-AT	85.44(0.10)	67.27(0.07)	46.41(0.07)	62.25(0.12)	30.55(0.04)	24.19(0.16)
TRADES	85.86(0.09)	56.79(0.08)	54.31(0.08)	62.23(0.07)	33.45(0.22)	29.07(0.25)
HAT	86.98(0.10)	56.81(0.17)	54.63(0.07)	60.42(0.03)	33.75(0.08)	29.42(0.02)
MART	83.17(0.18)	57.84(0.13)	51.84(0.09)	59.76(0.13)	33.37(0.11)	29.68(0.08)
ARoW	**87.65**(0.02)	**58.38**(0.09)	**55.15**(0.14)	**62.38**(0.07)	**34.74**(0.11)	**30.42**(0.10)

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)
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)

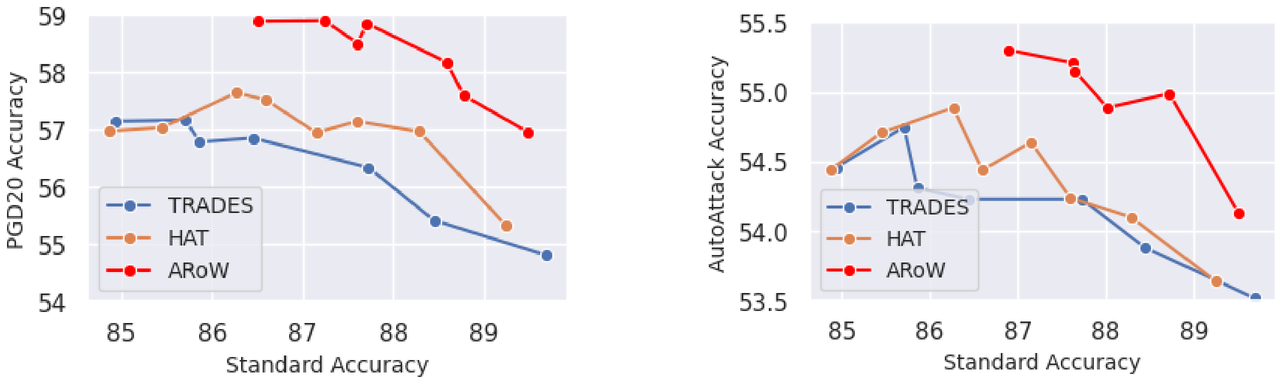

Figure 1. **Comparison of ARoW, TRADES and HAT with varying** λ. The x-axis and y-axis are the standard and robust accuracies, respectively. The robust accuracies in the left panel are against PGD^{20} while the robust accuracies in the right panel are against AutoAttack. We exclude the results of MART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

In Appendix A.2, we derive an upper bound of the robust risk which suggests $p_{\boldsymbol{\theta}}(y|\boldsymbol{x})$ as the regularization term that is completely opposite to that for MART. Numerical studies in Appendix B.1 show that the corresponding algorithm, called Confidence Weighted regularization (CoW), outperforms MART with large margins, which indicates that the regularization term is MART would be suboptimal. Note that ARoW is better than CoW even if the differences are not large.

4. Experiments

In this section, we investigate ARoW algorithm in view of robustness and generalization by analyzing the four benchmark data sets - CIFAR10, CIFAR100 (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 and CIFAR100 while ResNet-18 (He et al., 2016) is used for F-MNIST and SVHN. We apply SWA for mitigating robust overfitting (Chen et al., 2021) on CIFAR10 and CIFAR100. The effect of SWA are described in Section 4.3. Experimental details are presented in Appendix C. The code is available at https://github.com/dyoony/ARoW.

4.1. Comparison of ARoW to Other Competitors

We compare ARoW to other competitors TRADES (Zhang et al., 2019), HAT (Rade & Moosavi-Dezfolli, 2022), MART (Wang et al., 2020) explained in Section 2.3.2, PGD-AT (Madry et al., 2018) and GAIR-AT (Zhang et al., 2021) which are the algorithms minimizing the robust risk directly.

Table 2. **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**

Table 1 shows that ARoW outperforms the other competitors for various data sets and architectures 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 due to the gradient masking (Papernot et al., 2018; 2017) as described in Appendix D. The selected values of the hyperparameters for the other algorithms are listed in Appendix B.2.

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. Additional results for the trade-off are provided in Appendix G.2.

Experimental comparison to MART We observe that MART has relatively high robust accuracies against PGD-based attacks than other attacks. Table 3 shows the robust accuracies against four attacks included in AutoAttack (Croce & Hein, 2020b). Table 3 shows that MART has good performance for APGD, but not for APGD-DLR, FAB and SQUARE. This result indicates that the gradient masking occurs for MART. That is, PGD does not find good adversarial examples, but the other attacks easily find adversarial examples. See Appendix D for details about gradient masking.

4.2. 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 2 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 even though the margins compared to HAT are not significant. Note that ARoW has advantages other than the high robust accuracies. For example, ARoW is easy to implement compared to HAT since HAT requires a pre-trained model. Moreover, as we will see in Table 7, ARoW improves the fairness compared to TRADES while HAT improves the performance with sacrificing fairness.

4.3. Ablation studies

We study the following three issues - (i) the effect of label smoothing to ARoW, (ii) the effect of stochastic weighted averaging (Izmailov et al., 2018), (iii) the role of the new regularization term in ARoW to improve robustness and (iV) modifications of ARoW by applying tools which improve existing adversarial training algorithms.

4.3.1. Effect of Stochastic Weighting Averaging

The table presented in Appendix G.4 demonstrates a significant improvement in the performance of ARoW when

Table 3. **Comparison of MART and ARoW**. We compare the robustness of MART (Wang et al., 2020) and ARoW against the four attacks used in AutoAttack on CIFAR10. The results are based on WRN-34-10. We set $\lambda = 3$ and ARoW, respectively.

Method	**Standard**	**APGD**	**APGD-DLR**	**FAB**	**SQUARE**
MART	83.17	56.30	51.87	51.28	58.59
ARoW	**87.65**	**56.37**	**55.17**	**56.69**	**63.50**

Table 4. **Role of the new regularization term in ARoW. # Rob**$_{\text{TRADES}}$ and # **Rob**$_{\text{ARoW}}$ represent the number of samples which are robust to TRADES and ARoW, respectively. **Diff.** and **Rate of Impro.** denote (# **Rob**$_{\text{ARoW}}$ - # **Rob**$_{\text{TRADES}}$) and **Diff.** / # **Rob**$_{\text{TRADES}}$). The PGD^{10} is used for evaluating the robustness.

Sample's Robustness	# **Rob**$_{\text{TRADES}}$	# **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

SWA is applied. We believe this improvement is primarily due to the adaptive weighted regularization effect of SWA. Ensembling methods can improve the performance of models by diversifying them (Jantre et al., 2022) and SWA can be considered one of the ensembling methods (Izmailov et al., 2018). In the case of ARoW, the adaptively weighted regularization term $(1 - p_{\boldsymbol{\theta}}(y|\widehat{\boldsymbol{x}}^{\text{pgd}}))$ diversifies the models for averaging weights, which significantly improves the performance of ARoW.

4.3.2. Effect of Label Smoothing

Table 8 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 8 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 G.3 presents the results of an additional experiment to assess the effect of label smoothing to the performance.

4.3.3. 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 as well as TRADES, MART whose results are presented in Tables 4 and 5. Note that ARoW improves the robustness of initially less robust samples compared with TRADES and MART, respectively. We believe that this improvement is due to the regularization term in ARoW that enforces more regularization on less robust samples.

4.3.4. 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 G.6.

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.

4.4. Improved Fairness

Xu et al. (2021) reports that TRADES (Zhang et al., 2019) increases the variation of the per-class accuracies (accuracy

[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.

Table 5. **Comparing of ARoW to MART on sample's robustness. # Rob$_{\text{MART}}$** and **# Rob$_{\text{ARoW}}$** represent the number of samples which are robust to MART and ARoW, respectively. **Diff.** and **Rate of Impro.** denote (**# Rob$_{\text{ARoW}}$** - **# Rob$_{\text{MART}}$**) and **Diff. / # Rob$_{\text{MART}}$**). The autoattack is used for evaluating the robustness because of gradient masking.

Sample's Robustness	**# Rob$_{\text{MART}}$**	**# Rob$_{\text{ARoW}}$**	**Diff.**	**Rate of Impro. (%)**
Least Robust	150	148	-2	-1.3
Less Robust	729	865	**136**	18.65
Robust	962	984	22	2.29
Highly Robust	3515	3530	15	0.04

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

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)

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
HAT	86.74	65.40	11.12	57.92	24.20	18.26
ARoW	**87.58**	**74.51**	**7.11**	**59.32**	**31.05**	**15.67**

Table 8. **Comparison of TRADES and ARoW with/without label smoothing.** With WRN-34-10 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)

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 H.

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 $\mathrm{KL}(\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\mathbf{X})||\mathbf{p}_{\boldsymbol{\theta}}(\cdot|\widehat{\mathbf{X}}^{\text{pgd}}))$. 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 4.4 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.

Acknowledgement This work was supported by National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1A2C3A0100355014), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO.2022-0-00184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics].

References

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: a query-efficient black-box adversarial attack via random searchg. *In ECCV*, 2020.

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

Carmon, Y., Raghunathan, A., Ludwig, S., C Duchi, J., and Liang, P. S. Unlabeled data improves adversarial robustness. *In Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. 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`.

Chen, T., Zhang, Z., Liu, S., Chang, S., and Wang, Z. Robust overfitting may be mitigated by properly learned smoothening. *In International Conference on Learning Representations (ICLR)*, 2021.

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

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

Deng, Y., Zheng, X., Zhang, T., Chen, C., Lou, G., and Kim, M. An analysis of adversarial attacks and defenses on autonomous driving models. *IEEE International Conference on Pervasive Computing and Communications(PerCom)*, 2020.

Ding, G. W., Sharma, Y., Lui, K. Y. C., and Huang, R. Mma training: Direct input space margin maximization through adversarial training. *In International Conference on Learning Representataions(ICLR)*, 2020.

Finlayson, S. G., Chung, H. W., Kohane, I. S., and Beam, A. L. Adversarial attacks against medical deep learning systems. *In Science*, 2019.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *In International Conference on Learning Representations (ICLR)*, 2015.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. *In International Conference on Machine Learning (ICML)*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *In CVPR*, 2016.

Hitaj, D., Pagnotta, G., Masi, I., and Mancini, L. V. Evaluating the robustness of geometry-aware instance-reweighted adversarial training. *archive*, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *In Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information. *In International Conference on Machine Learning (ICML)*, 2018.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. *Proceedings of the international conference on Uncertainty in Artificial Intelligence*, 2018.

Jantre, S., Madireddy, S., Bhattacharya, S., Maiti, T., and Balaprakash, P. Sequential bayesian neural subnetwork ensembles. *arXiv*, 2022.

Jiang, L., Ma, X., Chen, S., Bailey, J., and Jiang, Y.-G. Black-box adversarial attacks on video recognition models. *In ACM*, 2019.

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

Kurakin, A., J Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. *In International Conference on Learning Representations (ICLR)*, 2017.

Li, Y., Xu, X., Xiao, J., Li, S., and Shen, H. T. Adaptive square attack: Fooling autonomous cars with adversarial traffic signs. *IEEE Internet of Things Journal*, 2020.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *In International Conference on Learning Representations (ICLR)*, 2017.

Ma, X., Niu, Y., Gu, L., Wang, Y., Zhao, Y., Bailey, J., and Lu, F. Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, 2020.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *In International Conference on Learning Representations (ICLR)*, 2018.

Morgulis, N., Kreines, A., Mendelowitz, S., and Weisglass, Y. Fooling a real car with adversarial traffic signs. *ArXiv*, 2019.

Müller, R., Kornblith, S., and Hinton, G. E. When does label smoothing help? *In Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

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

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. *In ACM*, 2017.

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

Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., and Hinton, G. E. Regularizing neural networks by penalizing confident output distributions. *In International Conference on Learning Representations (ICLR)*, 2017.

Rade, R. and Moosavi-Dezfolli, S.-M. Recuding excessive margin to achieve a better accuracy vs. robustness trade-off. *In International Conference on Learning Representations (ICLR)*, 2022.

Rebuffi, S.-A., Gowal, S., Calian, D. A., Stimberg, F., Wiles, O., and Mann, T. A. Data augmentation can improve robustness. *In Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Rice, L., Wong, E., and Kolter, J. Z. Overfitting in adversarially robust deep learning. *In International Conference on Machine Learning (ICML)*, 2020.

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

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. *In International Conference on Learning Representations (ICLR)*, 2020.

Wu, D., Xia, S.-T., and Wang, Y. Adversarial weight perturbation helps robust generalization. *In Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *archive*, 2017.

Xu, H., Liu, X., Li, Y., Jain, A. K., and Tang, J. To be robust or to be fair: Towards fairness in adversarial training. *In International Conference on Machine Learning (ICML)*, 2021.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. *In CVPR*, 2019.

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

Zhang, H., Yu, Y., Jiao, J., P Xing, E., El Ghaoui, L., and I Jordan, M. Theoretically principled trade-off between robustness and accuracy. *In International Conference on Machine Learning (ICML)*, 2019.

Zhang, J., Xu, X., Han, B., Niu, G., Cui, L., Sugiyama, M., and Kankanhalli, M. S. Attacks which do not kill training make adversarial learning stronger. *In International Conference on Machine Learning (ICML)*, 2020.

Zhang, J., Zhu, J., Niu, G., Han, B., Sugiyama, M., and Kankanhalli, M. S. Geometry-aware instance-reweighted adversarial training. *In International Conference on Learning Representations (ICLR)*, 2021.

A. Proof of Theorem 3.1

In this section, we prove Theorem 3.1. The following lemma provides the key inequality for the proof.

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

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

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

$$\begin{aligned}\mathbb{1}\left\{\exists \boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x},\varepsilon) : F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(\boldsymbol{x}'), F_{\boldsymbol{\theta}}(\boldsymbol{x}) = Y\right\} \\ \leq \mathbb{1}\left\{F_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq F_{\boldsymbol{\theta}}(z(\boldsymbol{x})), Y \neq F_{\boldsymbol{\theta}}(z(\boldsymbol{x}))\right\}\end{aligned} \tag{A.8}$$

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.8) 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.8) should be 0, and we complete the proof of the inequality. □

Theorem 3.1. *For a given score function $f_{\boldsymbol{\theta}}$, let $z(\cdot)$ be an any measurable mapping from $\mathcal{X}$ to $\mathcal{X}$ satisfying*

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

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

$$\begin{aligned}\mathcal{R}_{rob}(\boldsymbol{\theta}) &\leq \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}(Y \neq F_{\boldsymbol{\theta}}(\mathbf{X})) \\ &+ \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}(F_{\boldsymbol{\theta}}(\mathbf{X}) \neq F_{\boldsymbol{\theta}}(z(\mathbf{X})))\mathbb{1}\left\{p_{\boldsymbol{\theta}}(Y|z(\mathbf{X})) < 1/2\right\}\end{aligned} \tag{5}$$

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

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

the inequality (5) holds. □

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

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

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

$$\begin{aligned}\mathcal{R}_{rob}(\boldsymbol{\theta}) &\leq \mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}(Y \neq F_{\boldsymbol{\theta}}(\mathbf{X})) \\ &\quad + 2\mathbb{E}_{(\mathbf{X},Y)}\mathbb{1}(F_{\boldsymbol{\theta}}(\mathbf{X}) \neq F_{\boldsymbol{\theta}}(z(\mathbf{X}))) \cdot p_{\boldsymbol{\theta}}(Y|\mathbf{X})\end{aligned} \tag{A.9}$$

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

Since

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

the inequality (A.9) holds. □

B. Confidence Weighted Regularization (CoW)

Motivated from A.2, we propose the Confidence Weighted Regularization (CoW) which minimizes the following empirical risk:

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

B.1. Experimental Comparison of CoW to MART

We compare the CoW and MART for four attack methods included in AutoAttack (Croce & Hein, 2020b). CoW outperforms MART both on standard accuracies and robust accuracies except for PGD^{20}.

Table 9. **Comparison of MART and CoW**. We compare the robustness of MART (Wang et al., 2020) and ARoW against the four attacks used in AutoAttack on CIFAR10. The results are based on WRN-34-10. We set $\lambda = 4$ for CoW.

Method	**Standard**	**APGD**	**APGD-DLR**	**FAB**	**SQUARE**
MART	83.17	**56.30**	51.87	51.28	58.59
CoW	**88.53**	56.15	**54.79**	**56.67**	61.88

We divide the test data into four groups - least correct, less correct, correct and highly correct according to the values of $p_{\boldsymbol{\theta}_{\text{PGD}}}(y|\boldsymbol{x})$ (< 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). Note that CoW improves the robustness of correct and highly correct samples compared with MART. We believe that this improvement is due to the regularization term in CoW that enforces more regularization on correct samples.

Table 10. **Comparing of CoW to MART on sample's robustness. # Rob$_{\text{MART}}$** and **# Rob$_{\text{CoW}}$** represent the number of samples which robust to MART and CoW, respectively. **Diff.** and **Rate of Impro.** denote (# **Rob$_{\text{CoW}}$** - # **Rob$_{\text{MART}}$** and **Diff. / # Rob$_{\text{MART}}$**). The autoattack is used for evaluating the robustness because of gradient masking.

Sample's Correctness	**# Rob$_{\text{MART}}$**	**# Rob$_{\text{CoW}}$**	**Diff.**	**Rate of Impro. (%)**
Least Correct	0	3	3	-
Less Correct	78	59	-19	-24.05
Correct	322	346	24	7.45
Highly Correct	4958	5072	**114**	2.30

C. Detailed settings for the experiments with benchmark datasets

C.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.

C.2. Hyperparameter setting

Table 11. **Selected hyperparameters.** Hyperparameters used in the numerical studies in Section 4.1.

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	5	-	$2e^{-4}$	-	o
		PGD-AT	-	-	$5e^{-4}$	-	o
		GAIR-AT	-	-	$5e^{-4}$	-	o
		ARoW	3	-	$5e^{-4}$	0.2	o
CIFAR100	WRN-34-10	TRADES	6	-	$5e^{-4}$	-	o
		HAT	4	0.5	$5e^{-4}$	-	o
		MART	4	-	$5e^{-4}$	-	o
		PGD-AT	-	-	$5e^{-4}$	-	o
		GAIR-AT	-	-	$5e^{-4}$	-	o
		ARoW	4	-	$5e^{-4}$	0.2	o
SVHN	ResNet-18	TRADES	6	-	$5e^{-4}$	-	x
		HAT	4	0.5	$5e^{-4}$	-	x
		MART	4	-	$5e^{-4}$	-	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	4	-	$5e^{-4}$	-	x
		PGD-AT	-	-	$5e^{-4}$	-	x
		GAIR-AT	-	-	$5e^{-4}$	-	x
		ARoW	6	-	$5e^{-4}$	0.25	x

Table 11 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. We use stochastic weight averaging (SWA) for CIFAR10 and CIFAR100. Only for MART (Wang et al., 2020) with WRN-34-10, we use weight decay $2e^{-4}$ as (Wang et al., 2020) did since MART works poorly with $5e^{-4}$ with SWA.

D. Checking the Gradient Masking

Table 12. **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(0.17)	**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 12, 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 1 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).

E. Detailed setting for the experiments with extra data

Table 13. **Selected hyperparameters.** Hyperparameters used in the numerical studies in Section 4.2. 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 4.2, 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.

F. Additional Results with Extra Data

F.1. Additional Results with Extra Data

In the main manuscript, we use architecture of ResNet18, while Rade & Moosavi-Dezfolli (2022) use PreAct-ResNet18. For better comparison, we conduct an additional experiment with extra data where the same architecture - PreaAct-ResNet18 is used. In addition, we set batch size to 1024 which is used in Rade & Moosavi-Dezfolli (2022). Table 14 shows that ARoW outperforms HAT both on standard accuracy($+0.29\%$) and robust accuracy($+0.11\%$) against autoattack.

Table 14. **Performance with extra data (Carmon et al.) on CIFAR10.** We brought the values in the paper as reported in Rade & Moosavi-Dezfolli (2022).

Method	**Standard**	**AutoAttack**
HAT	89.02	57.67
ARoW	89.31	57.78

G. Ablation study

G.1. The performance on CIFAR10 - ResNet18

Table 15. **Performance on CIFAR10 with ResNet18.** 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
PGD-AT	82.42(0.05)	53.48(0.11)	49.30(0.07)
GAIR-AT	81.09(0.12)	64.89(0.04)	41.35(0.16)
TRADES	82.41(0.07)	52.68(0.22)	49.63(0.25)
HAT	83.05(0.03)	52.91(0.08)	49.60(0.02)
MART	74.87(0.95)	53.68(0.30)	49.61(0.24)
ARoW	**82.53**(0.13)	**55.08**(0.16)	**51.33**(0.18)

G.2. The trade-off due to the choice of λ

Table 16 presents the trade-off between the generalization and robustness accuracies of ARoW on CIFAR10 due to the choice of λ, where ResNet18 is used. The trade-off is obviously observed.

Table 16. **Standard and robust accuracies of ARoW on CIFAR10 for varying λ.**

λ	Standard	PGD^{20}	AutoAttack
TRADES($\lambda = 6$)	82.41	52.68	49.63
ARoW($\lambda = 2.5$)	85.30	53.80	49.66
ARoW($\lambda = 3.0$)	84.65	54.23	50.11
ARoW($\lambda = 3.5$)	83.86	54.13	50.15
ARoW($\lambda = 4.0$)	83.73	54.20	50.55
ARoW($\lambda = 4.5$)	82.97	54.69	50.83
ARoW($\lambda = 5.0$)	82.53	55.08	51.33

G.3. The effect of label smoothing

Table 17 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 17. **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

G.4. 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 18. 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 18. **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)

G.5. Various perturbation budget ε

Table 19 and 20 show the performance of various perturbation budget ε for train and test phases, respectively. The regularization parameters of this studies are 3.5 and 6 for ARoW and TRADES, respectively. We observe that ARoW outperforms TRADES in all cases.

Table 19. **Performance of various train perturbation budget ε on CIFAR10 with ResNet-18.** We train models using ARoW and TRADES with varying ε and evaluate the robustness with same $\varepsilon = 8$.

ε for training	Method	Standard	PGD^{20}	AutoAttack
4	ARoW	89.45	72.98	71.99
	TRADES	88.30	72.22	71.29
6	ARoW	86.40	62.84	60.33
	TRADES	85.13	62.05	59.91
8	ARoW	83.34	53.93	50.37
	TRADES	82.26	52.18	49.13
10	ARoW	81.36	45.09	40.41
	TRADES	80.09	42.75	38.47
12	ARoW	80.03	37.87	32.14
	TRADES	76.49	36.68	31.60

Table 20. **Performance of various test perturbation budget ε on CIFAR10 with ResNet-18.** We train models using ARoW and TRADES with $\varepsilon = 8$ and evaluate the performance with varying ε.

ε for test	Method	Standard	PGD^{20}	AutoAttack
4	ARoW	83.34	70.61	69.02
	TRADES	82.26	68.50	67.17
6	ARoW	83.34	62.50	59.87
	TRADES	82.26	61.11	58.66
8	ARoW	83.34	53.93	50.37
	TRADES	82.26	52.18	49.13
10	ARoW	83.34	45.13	41.01
	TRADES	82.26	43.99	40.25
12	ARoW	83.34	37.10	32.67
	TRADES	82.26	36.08	32.13

G.6. AWP and FAT

G.6.1. 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

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

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.

G.6.2. 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).

H. Improved fairness

Table 7 shows that ARoW improves the fairness in terms of class-wise accuracies. the worst-class accuracy (WC-Acc) and standard deviation of class-wise accracies (SD) are defined by WC-Acc $= \min_{c} \text{Acc}(c)$ and SD $= \sqrt{\frac{1}{C} \sum_{c=1}^{C} (\text{Acc}(c) - \bar{\text{Acc}})^2}$ where $\text{Acc}(c)$ is the accuracy of class c and $\bar{\text{Acc}}$ is the mean of class-wise accuracies.

Table 21. **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 21, 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 $(26.1\% \rightarrow 30.2\%)$ and 9.2 percentage point $(65.9\% \rightarrow 75.1\%)$ 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.