Adaptive Weight Assignment for Adversarial Training Based on Predicted Class Probabilities Across Different Attacks and Perturbation Sizes

Modeste Atsague[1(✉)], Jin Tian[2], and Olukorede Fakorede[3]

[1] Iowa State University, Ames, USA
modeste@iastate.edu
[2] Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
Jin.Tian@mbzuai.ac.ae
[3] Iowa State University, Ames, USA
fakorede@iastate.edu

Abstract. Adversarial training (AT) improves model robustness by incorporating adversarial examples during training. Traditional methods, however, treat all examples equally, limiting their effectiveness. Recent studies show that adversarial examples vary in importance, and failing to account for this can weaken robustness. New approaches assign different weights to adversarial examples, improving defenses against specific attacks while maintaining natural accuracy. However, existing reweighting strategies often struggle against stronger attacks like CW and AA. Our analysis reveals that misclassified inputs may be assigned to different incorrect classes depending on the attack type and perturbation size, suggesting that more than one metric for weight assignment is required. To tackle this, we propose an Adaptive Weight Assignment (AWA) strategy that uses predicted class probabilities across multiple attack types and perturbation sizes. This method strengthens weaker adversarially trained models and significantly improves robustness against strong attacks like CW and AA, as confirmed by our extensive experiments.

1 Introduction

Our modern society heavily depends on technology, with deep neural networks (DNNs) playing a pivotal role in critical areas like self-driving cars, recommendation systems, and facial recognition. While DNNs have brought significant advancements, they are vulnerable to adversarial examples. To address these challenges, researchers have explored various methods to improve model robustness, with adversarial training (AT) [7] emerging as the most effective and foundational method. [11] formulated the adversarial training process as an optimization problem, aiming to find the model parameters θ that minimize the risk $\min_\theta \frac{1}{n}\sum_{i=1}^{n} l(f_\theta(x_i'), y_i)$. $l(\cdot)$ represents the loss function, $f_\theta(x_i')$ denotes the neural network's prediction with parameters θ for the adversarially perturbed input x_i' generated during the inner maximization, and y_i is the corresponding class label. This approach has paved the way for the development of

X. Wu et al. (Eds.): PAKDD 2025, LNAI 15873, pp. 27–38, 2025.
https://doi.org/10.1007/978-981-96-8183-9_3

numerous defense strategies, including several variants of adversarial training, as proposed in [2,5,11,13,14,16]. The results of adversarial training and its variations, though impactful, remain unsatisfactory, with a persistent gap between natural and adversarial accuracy. Meanwhile, other researchers have focused on assigning unequal weights to the training loss [6,8,10,17]. Among these improvement strategies, our work aligns with reweighting. Thus, we will discuss existing reweighting strategies and then elaborate on our proposed method.

1.1 Existing Works

In recent years, adversarial machine learning has garnered significant attention due to the vulnerability of deep learning models to adversarial attacks. Various defense strategies have been proposed to mitigate these risks, out of which some well-known approaches include adversarial training, regularization-based methods, reweighting, robust optimization, and detection-based defenses. We provide a brief description of regularization and reweighting methods.

Regularization-Based Methods:
Regularization techniques enhance training by promoting smoother decision boundaries, stabilizing model behavior under adversarial conditions, and reducing overfitting to adversarial examples. Researchers have explored these strategies to strengthen defenses in deep learning models. One notable example is TRADES [16], which balances natural accuracy and adversarial robustness using a loss function with two components. MART (Misclassification-Aware Adversarial Training), proposed by [13], focuses on misclassified examples during adversarial training. It uses a regularization term that maximizes the margin between misclassified and correctly classified samples based on the premise that misclassified examples are more susceptible to adversarial attacks.

Reweighting Based Defenses in Adversarial Training:
Geometry-Aware Instance-Reweighted Adversarial Training (GAIRAT) [17], prioritizing examples near decision boundaries as they are more vulnerable to attacks and thus require more attention during training. Similarly, Margin-Aware Instance Reweighting Learning (MAIL) [10] focuses on examples close to the decision margin, using predicted class probabilities to estimate their distance from the boundary. Existing reweighting strategies need improvement to perform effectively against stronger adversarial attacks such as Carlini-Wagne (CW) [3] and AutoAttack (AA) [4]. One of the main reasons for this shortcoming is that current reweighting approaches often rely on a single criterion to determine the weight of each adversarial example, such as the distance to the decision boundary or the classification confidence on the perturbed example. This can lead to misallocation of weights, where adversarial examples that are assigned lower weights may still contain important information that is crucial for improving the model's overall robustness.

Limitations of Existing Works:
Robustness still falls short, especially under strong attacks like CW, AA, and powerful black-box attacks. These challenges emphasize the need for more

effective strategies to enhance model resilience and ensure strong performance against sophisticated adversarial threats.

2 Notations

Consider a standard classification problem defined over a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$ where x_i represents the natural input corresponding to the label $y_i \in Y = \{1, ..., C\}$, with C denoting the total number of classes. Let $f_\theta(.)$ denote a deep neural network parameterized by θ. Let $f_c(x_i, \theta)$ be the *logit* output of the deep neural network with model parameters θ corresponding to class c and $p_c(x_i, \theta) = e^{f_c(x_i,\theta)} / \sum_{c'=1}^{C} e^{f_{c'}(x_i,\theta)}$ represent the probability that the network predicts class c given the input example x_i. Let $f_\theta(x_i)$ represent the class prediction of the network. We denote by $l(.)$ and $\mathbb{E}_{(x,y)}$ the loss and expected loss, respectively. The expected loss of the network over the dataset D is defined by

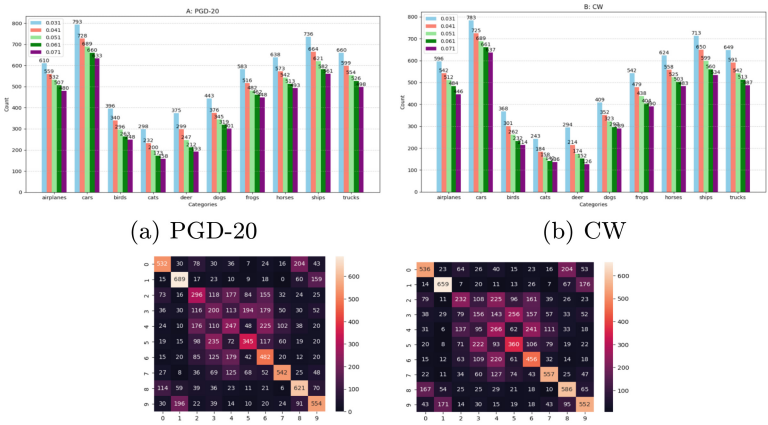
$$\mathbb{E}_{(x,y)} = \frac{1}{n} \sum_{i=1}^{n} l(f_\theta(x_i), y_i). \tag{1}$$

In the context of adversarial learning, we also consider the adversarial samples x_i', which are perturbed versions of the natural inputs x_i, designed to mislead the classifier. Assume an initial point $x^{(0)}$, representing the natural data perturbed by small Gaussian or uniform random noise, i.e., $x^{(0)} = x_i + Gaussian/Uniform$, where $x^{(0)}$ lies in the input feature space with a distance metric $||x - x'||_\infty$. Let $t \in \mathbb{N}$. PGD generates adversarial examples using the following update rule $x^{(t+1)} = \prod_{B_\epsilon[x_i]}(x^{(t)} + \alpha \cdot sign(\nabla_{x^{(t)}} g_i'(f_\theta(x^{(t)}), y_i)))$. α is a step size, $\prod_{B_\epsilon[x_i]}(.)$ is the projection function, $B_\epsilon[x] = \{x' | \, \|x' - x\|_p < \epsilon\}$ is a neighborhood of x, $x^{(t)}$ is the adversarial example at step t and $g_i'(.)$ is the loss used to generate the adversarial used for training.

3 Proposed Method

Despite the progress made with existing reweighting techniques, their performance against stronger attacks like CW and AA still needs to be improved. This highlights the need to reconsider and refine current reweighting strategies. This section begins with an insightful experiment to motivate our approach, followed by the details of the proposed reweighting method.

Motivating Experiment:
To motivate our proposed method, we consider a simple yet insightful experiment. Specifically, we analyze a model trained with TRADES, one of the most robust defensive approaches. This experiment is conducted on the CIFAR-10 dataset using a ResNet-18 architecture. The experiment consists of ten runs, recording the average number of correctly classified samples under two attack scenarios: PGD-20 (Fig. 1(a)) and CW (Fig. 1(b)). These results are presented



(a) PGD-20 (b) CW

(c) PGD-20: perturbation size 0.051 (d) PGD-20: perturbation size 0.061

Fig. 1. Adversarial Vulnerabilities Across Attack Methods and Perturbation Sizes.

across varying perturbation sizes. Additionally, under two specific perturbation sizes, confusion matrices for the PGD-20 attack are shown in Figs. 1(c) and 1(d).

Closely examining Figs. 1(a) and 1(b) reveals that as the perturbation size increases, the drop in correctly classified adversarial samples is influenced by both class and attack type. For instance, in the "deer" class: Under PGD-20, the number of correctly classified samples decreased by 76, 52, 35, and 19 as the perturbation size rose from 0.031 to 0.041, 0.041 to 0.051, 0.051 to 0.061, and 0.061 to 0.071, respectively. On the other hand, under CW attack, the decreases were 80, 40, 22, and 26 for the same perturbation size increments, highlighting that adversarial robustness strongly depends on the attack type and the perturbation size. While increasing the perturbation size prompts a decline in correctly classified samples, the rate of this decline varies. A slow decline as the perturbation size increases indicates some degree of resilience at a higher perturbation size. This comparison highlights two key observations: First, the rate of decline in correctly classified samples varies between attack types, indicating the nuanced impact of attack strategies on robustness. Second, a slower decline at larger perturbation sizes suggests that the model retains some resilience under more aggressive adversarial conditions. Bridging this gap in decline across attacks and perturbation sizes remains a critical challenge for enhancing robustness while maintaining consistency across diverse adversarial scenarios, highlighting the need for adaptive defenses that address attack diversity and varying perturbation levels to enhance robustness. Relying on a single factor, such as the distance to the decision boundary or classification confidence, when assigning weights to adversarial examples can overlook essential nuances in the adversarial training process. As a result, suboptimal weight assignment may occur, where valuable adversarial examples are not given the necessary

attention during training, leading to a weakened defense against more complex attacks. Additionally, the confusion matrices (Figs. 1(c) and 1(d)) provide further insights. These matrices show that the counts of correctly and incorrectly classified samples vary inconsistently across classes. For example: In Fig. 1(c), under PGD-20, the model incorrectly classified class 6 as classes 5, 4, and 3 with a count of 117, 225, and 179 times, respectively. Under the same PGD-20 but at different perturbation sizes (Fig. 1(d)), the same misclassifications occurred 106, 241, and 157 times, suggesting that certain classes are more sensitive to the perturbation size than others. Classes such as 6 exhibit varying vulnerability to attacks depending on the specific perturbation level and attack type. To the best of our knowledge, this is the first exploration of the rate of decline in correctly classified samples and misclassification patterns across classes under varying perturbation sizes. These findings emphasize the need to account for class-specific and attack-specific behavior when designing adversarial training methods to improve robustness.

Proposed Method: Adaptive Weight Assignment:
We consider an adversarially trained model f_{weak}, a baseline model intended for improvement. During training, we leverage the prediction confidence of f_{weak} to develop a more stable and robust model. This approach dynamically adapts the weights based on insights derived from f_{weak}, enabling the model to focus on addressing its own vulnerabilities and enhancing robustness. Specifically, using f_{weak}, we evaluate each adversarial example's predicted class probability distribution under attacks with varying perturbation sizes, allowing us to assign higher weights to adversarial examples that are more challenging for the model f_{weak}, such as those that cause misclassification under attacks like CW and PGD-20. At the same time, adversarial examples deemed less challenging, based on their predicted class probabilities, are assigned lower weights but are not ignored entirely, as they may still provide helpful information for the model's learning process. This ensures that the model is able to focus on difficult-to-classify adversarial examples while still learning from easier examples, thus addressing the imbalance that traditional AT methods often encounter. Let $w()$ represent the weighting function used in the adaptive reweighting strategy. The weighting function $w()$ can take advantage of the weak model's confidence levels when incorrectly classifying natural and adversarial samples. The function can emphasize or de-emphasize certain misclassified examples during training by assigning weights based on the weak model's certainty or uncertainty in its predictions.

The results illustrated and reported in Fig. 1 demonstrate that the distribution of misclassified samples is heavily influenced by the type of adversarial attack and the perturbation size introduced. In particular, for certain classes, the number of samples incorrectly classified into another class becomes notably high under specific adversarial conditions, suggesting that certain classes are more prone to adversarial perturbations, leading to a significant shift in predicted labels, which emphasizes the need for adaptive mechanisms to improve model robustness across different attack scenarios.

Formally, considering a classification problem with C distinct classes, we define the predicted class probability distribution for natural examples as $P_C = [p_{c_1}, p_{c_2}, \ldots, p_{c_{|C|}}]$, where each p_{c_i} represents the probability that a natural example is classified into class i, $1 < i < |C|$ and $\sum_{i=1}^{|C|} p_{c_i} = 1$. Similarly, $P'_C = [p'_{c_1}, p'_{c_2}, \ldots, p'_{c_{|C|}}]$ represent the predicted class probability distribution for the corresponding adversarial example, perturbed under the perturbation size α, where each p'_{c_i} represents the model's confidence in assigning the adversarial example to class i and $\sum_{i=1}^{|C|} p'_{c_i} = 1$. For each input x_i and the corresponding adversarial x'_i, the model prediction probability is given by $p_i = \max(P_C)$ and, $p'_i = \max(P'_C)$ respectively. Now consider a batch B of N input-label pairs, denoted by $\{(x_i, y_i)\}_{i=1}^{N}$, where x_i is an input example and y_i is its corresponding true class label. For each sample, x_i let x'_i denote the adversarial example generated to attack the weak model $f_{\text{weak}}(\cdot)$. If $f_{\text{weak}}(x'_i) \neq y_i$ with probability p_i, we define the probability sets $P_{\text{CW}}[\alpha]$ and $P_{\text{PGD-20}}[\alpha]$ to represent the probabilities of misclassified samples under CW and PGD-20 attacks with perturbation size α, respectively. Formaly, $P_{\text{CW}}[\alpha] = \{p'_i | f_{weak}(x'_i) \neq y_i, \text{under CW attack with perturbation size } \alpha\}$ and $P_{\text{PGD-20}}[\alpha] = \{p'_i | f_{weak}(x'_i) \neq y_i, \text{under PGD-20 attack with perturbation size } \alpha\}$.

Additionally, let $P_{Nat} = \{p_i | f_{weak}(x_i) \neq y_i\}$, which represents the set of probabilities for inputs x_i misclassified by the weak model on natural examples. Finally, let

$$P_{Adv} = mean(\sum_{j=1}^{n} P_{PGD-20}[\alpha_j]) + mean(\sum_{j=1}^{n} P_{CW}[\alpha_j]) \tag{2}$$

and

$$P_{Nat} = mean(P_{Nat}), \quad w = \exp \frac{(P_{Adv} + P_{Nat})}{c} \tag{3}$$

In Eqs. 2 and 3, $mean(\cdot)$ denotes the average of the input values. The perturbation sizes are represented by α_i. In this work, we consider $n = 2$, corresponding to two perturbation sizes: α_1 and α_2. In Eq. 3, the constant c is introduced to adjust the weight.

In summary, the sets $P_{CW}[\alpha]$, $P_{PGD-20}[\alpha]$ and P_{Nat} capture the weak model's confidence in misclassifying adversarial and natural examples under different conditions. These probabilities are used in conjunction with the weighting function $w()$ to adjust the model's learning based on the difficulty and nature of the examples, thereby enhancing adversarial robustness. Our adaptive weighting scheme is designed to dynamically modify the contribution of each training example, encompassing both natural and adversarial instances according to their respective misclassification probabilities. Specifically, we assign larger weights to those examples that are misclassified with high confidence by the weak model when exposed to both unperturbed data and adversarial attacks, such as the CW and PGD-20 methods. In our approach, we also account for varying perturbation sizes during these attacks to enhance the robustness of the training process. The core principle underlying our adaptive weighting strategy is to steer the model's

learning process toward examples where it displays significant uncertainty or misclassification. Since the weight increases exponentially with the misclassification probability, the loss for harder-to-classify adversarial examples is amplified, causing the model to focus more on minimizing the risk for these examples during training, thereby reducing the overall adversarial misclassification rate. In this case, we consider two perturbation sizes, $\alpha_1 = 0.051$ and $\alpha_2 = 0.061$.

Integrating Into Existing Works:
We propose integrating our novel reweighting strategy with two promising adversarial training methods to enhance their robustness and adaptability. We aim to achieve superior robustness and performance across different scenarios by incorporating our reweighting strategy into these adversarial training methods. In the TRADES framework, the loss function is composed of two key components: the cross-entropy loss, which measures the performance on clean examples, and the Kullback-Leibler (KL) divergence, which quantifies the discrepancy between the natural and adversarial predictions. The TRADES loss is defined as

$$CE(p(x_i, \theta), y_i) + \frac{1}{\lambda} \cdot KL(p(x_i, \theta)||p(x_i', \theta)). \tag{4}$$

where $p(x_i', \theta)$ is the predicted probability for the perturbed input x_i' and y_i is the true label. To improve TRADES, we introduce a reweighted KL term

$$CE(p(x_i, \theta), y_i) + w \cdot \frac{1}{\lambda} KL(p(x_i, \theta)||p(x_i', \theta)). \tag{5}$$

The weight w dynamically adjusts the Kullback-Leibler (KL) divergence based on the model's confidence. Lastly, MART optimizes

$$BCE(p(x_i', \theta), y_i) + \lambda \cdot KL(p(x_i, \theta)||p(x_i', \theta)) \cdot (1 - p_{y_i}(x_i, \theta)). \tag{6}$$

To improve MART, we optimized the weighted loss defined by

$$BCE(p(x_i', \theta), y_i) + w \cdot \lambda \cdot KL(p(x_i, \theta)||p(x_i', \theta)) \cdot (1 - p_{y_i}(x_i, \theta)). \tag{7}$$

By adjusting the weights w based on the probability of misclassification for each adversarial example, we enhance the model's ability to focus on more challenging cases during training. This adaptive reweighting scheme fine-tunes the model's robustness by prioritizing adversarial examples that are harder to classify, thus forcing the model to allocate more learning capacity to regions of the input space where adversarial vulnerability is higher. As a result, this approach leads to improved overall robustness, as demonstrated by the significant performance gains recorded in the experimental results, particularly against stronger adversarial attacks. We denote the enhanced training objectives as

TRADES+AWA and **MART+AWA** representing the improved versions of TRADES and MART, respectively.

Algorithm 1: As an example, we show the training procedure of TRADES+AWA in the following

Input: Training data $D = \{x_i, y_i\}_{i=1}^{n}$, c, step size μ_1 and μ_2 for the inner and the outer optimization respectively, the batch size m, the number of outer iteration T, the number of inner iteration K, and ϵ the perturbation size. Consider the perturbation sizes α_1, α_2. A model $f_{\theta_{weak}}$(Model trained on TRADES, no weights applied)

Initialization:

Instantiate and initialize a model f_θ with the weights of $f_{\theta_{weak}}$

for $t = 1, 2, ..., T$ **do**

// At random, uniformly sample a mini-batch of training data $B_{(t)} = \{x_1, ..., x_m\}$.

// Using $f_{\theta_{weak}}$, α_1 and α_2, generate adversarial samples for each $x_i \in B_t$,

// Compute P_{Adv}, P_{Nat} according to Eqs. 2 and 3.

for *each* $x_i \in B_{(t)}$ **do**

$x_i^{'} = x_i + 0.001 \times \varsigma; \varsigma \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

for $k = 1, 2, ..., K$ **do**

$x_i^{'} = \prod_{B_\epsilon[x_i]}(x_i^{'} + \mu_1 sgn(\nabla_{x_i^{'}}[CE(p(x_i', \theta), y_i)])$

end

end

$w = \exp \frac{(P_{Adv} + P_{Nat})}{}$

$\theta = \theta - \frac{\mu_2}{m} \sum_{i=1}^{c} \nabla_\theta [CE(p(x_i, \theta), y_i) + \frac{1}{\lambda} \cdot w \cdot KL(p(x_i, \theta) || p(x_i', \theta))]$

end

Output: f_θ (This model is significantly more robust, and we refer to it as f_{Strong})

4 Experiments

We conducted a series of experiments and compared our method with the state-of-the-art defenses on benchmark datasets CIFAR-10, CIFAR-100, and TinyImageNet. We tested on two model architectures: ResNet-18 and a larger capacity network, WideResNet-34-10.

Baselines: We compare with top-performing variants of adversarial training defenses TRADES and MART. Additionally, we compare our work against other reweighting methods, Geometry-Aware Instance-Reweighted Adversarial Training (GAIRAT) [17] and MAIL [10]. In addition, we consider recent promising margin-based adversarial training approaches MMA [5] and WAT [15].

Training Settings: The hyperparameters were selected using the Ray Tune hyperparameter search tool as proposed in [9], and the best parameters identified are as follows: ResNet-18 on TinyImageNet (c=3 and both $\frac{1}{\lambda}$ and λ are 8.0

for TRADES+AWA and MART+AWA respectively). CIFAR-100 (c=6 and 3 for MART+AWA and TRADES+AWA, respectively. On the other hand, $\lambda = 9.0$ for MART+AWA and $\frac{1}{\lambda}$= 8.0 for TRADES+AWA). CIFAR-10 on WRN-34-10 and ResNet-18 (c=6 for both architectures. $\lambda = 9.0$ for MART+AWA and $\frac{1}{\lambda} = 6.0$ for TRADES+AWA). For TRADES, $\frac{1}{\lambda}$ is set to 6.0, and λ is 5.0 in MART as specified in their original papers. We use the same parameters defined in their original papers for other baselines. All the models are trained using SGD for 130 epochs with momentum 0.9 and the batch size m=100. The initial learning rate is 0.01, then decayed by a factor of ten at the 75th and further decayed at the 90th epoch. We consider the weight decay of 3.5e-3. Adversarial data used in training are generated using PGD with a random start, maximum perturbation ϵ set to 8/255. The step size $\mu_1 = \mu_2 = 2/255$, and the number of steps, is K=10. We consider two perturbation sizes, $\alpha_1 = 0.051$ and $\alpha_2 = 0.061$.

Evaluation Details: We evaluated our method under white-box attack including the L_∞ PGD-20/100 [11], CW (PGD optimized with CW loss, confidence level K=50), and AA [4]. The perturbation size is set to ϵ=8/255 under the white-box attack, and the step size is 1/255. Additionally, we evaluated strong Black-box attacks SQUARE [1] and SPSA [12], which is a stronger query-based black box attack, with the perturbation size of 0.001 (for gradient estimation), sample size of 100, 20 iterations, and learning rate 0.01.

Table 1. Clean and robust accuracy on **ResNet-18** and under **CIFAR-10**. We perform six runs and report the average performance with 95% confidence intervals. The 'Clean' column represents accuracy on natural examples.

Method	Clean	PGD-20	PGD-100	CW	AA	SQUARE	SPSA
TRADES	**82.46** $\pm$0.0012	54.78 $\pm$ 0.0010	53.45 $\pm$ 0.0032	51.65 $\pm$ 0.0021	49.08 $\pm$ 0.0031	55.64 $\pm$ 0.0011	56.50 $\pm$ 0.0020
TRADES + AWA	82.30 $\pm$ 0.014	**56.35** $\pm$ 0.0021	**55.02** $\pm$ 0.021	**53.70** $\pm$ 0.012	**51.50** $\pm$ 0.011	**56.93** $\pm$ 0.024	**60.45** $\pm$ 0.021
MART	81.30 $\pm$ 0.003	54.73 $\pm$ 0.006	53.28 $\pm$ 0.005	51.86 $\pm$ 0.0031	49.01 $\pm$ 0.0020	55.66 $\pm$ 0.0031	56.15 $\pm$ 0.0040
MART + AWA	**82.18** $\pm$ 0.022	**56.92** $\pm$ 0.006	**55.29** $\pm$ 0.015	**52.88** $\pm$ 0.014	**49.17** $\pm$ 0.016	**56.30** $\pm$ 0.021	**59.66** $\pm$ 0.021

Table 2. Clean and robust accuracies on **WRN-34-10** and under **CIFAR-10**. We perform six runs and report the average performance with 95% confidence intervals. The 'Clean' column represents accuracy on natural examples.

Method	Clean	PGD-20	PGD-100	CW	AA	SQUARE	SPSA
TRADES	**84.58** $\pm$ 0.0021	57.71 $\pm$ 0.0012	56.69 $\pm$ 0.002	55.01 $\pm$ 0.0013	52.57 $\pm$ 0.002	59.45 $\pm$ 0.0024	61.09 $\pm$ 0.0023
TRADES + Ours	84.38 $\pm$ 0.0012	**58.53** $\pm$ 0.0221	**57.45** $\pm$ 0.0011	**56.49** $\pm$ 0.0112	**54.19** $\pm$ 0.0013	**59.79** $\pm$ 0.0015	**63.10** $\pm$ 0.0031
MART	84.25 $\pm$ 0.001	58.29 $\pm$ 0.0032	55.56 $\pm$ 0.0011	54.82 $\pm$ 0.002	51.40 $\pm$ 0.00	58.21 $\pm$ 0.0013	59.87 $\pm$ 0.00
MART + Ours	**84.88** $\pm$ 0.004	**59.30** $\pm$ 0.005	**57.29** $\pm$ 0.001	**56.04** $\pm$ 0.0021	**52.34** $\pm$ 0.003	**59.09** $\pm$ 0.0022	**62.84** $\pm$ 0.002

Experimental Results: A detailed examination of the results presented in Tables 2 and 3 demonstrates the significant performance improvements achieved

Table 3. Clean and robust accuracies on **ResNet-18** and under **CIFAR-100**. We perform six runs and report the average performance with 95% confidence intervals. The 'Clean' column represents accuracy on natural examples.

Method	Clean	PGD-20	PGD-100	CW	AA	SQUARE	SPSA
TRADES	57.16 ± 0.0010	30.32 ± 0.021	29.48 ± 0.021	25.16 ± 0.031	25.18 ± 0.031	30.46 ± 0.022	32.06 ± 0.014
TRADES + Ours	**60.18** ± 0.032	**31.77** ± 0.011	**30.95** ± 0.031	**28.15** ± 0.025	**26.03** ± 0.024	**31.08** ± 0.054	**33.91** ± 0.033
MART	54.02 ± 0.0013	31.13 ± 0.014	30.14 ± 0.011	26.98 ± 0.010	24.83 ± 0.012	31.17 ± 0.016	32.45 ± 0.014
MART + Ours	**56.83** ± 0.014	**32.79** ± 0.021	**32.12** ± 0.025	**29.37** ± 0.027	**26.34** ± 0.014	**31.45** ± 0.012	**33.40** ± 0.018

by our proposed method over TRADES and MART, particularly under stronger attacks like AutoAttack (AA). For instance, under ResNet-18, TRADES is improved by 1.57% under PGD-20 and PGD-100, 2.05% under CW, 2.42% under AA, and lastly 1.29% and 3.95% under SQUARE and SPSA attacks respectively. On the other hand, MART+AWA recorded an improvement of 0.88% under Natural Accuracy, 2.19% and 2.01% under PGD-20 and 100, respectively, 1.02% under CW, 0.64% under SQUARE and lastly 3.51% under SPAS. We recorded a minor improvement under AA. On the WRN-34-10 architecture, MART+AWA achieved gains of 0.63% in natural accuracy, 1.01% and 1.73% under PGD-20 and PGD-100, 1.22% under CW, 0.94% under AA, 0.88% under SQUARE, and 2.97% under SPSA. TRADES+AWA demonstrated a strong performance against AA with a significant improvement of 1.64% while maintaining comparable natural accuracy. Additional improvements included 0.82% and 0.76% under PGD-20 and PGD-100, 1.48% under CW, and 2.01% under SPSA.

Table 4. Clean and robust accuracies on **TinyImageNet**, **ResNet-18** . We perform six runs and report the average performance with 95% confidence intervals. The 'Clean' column represents accuracy on natural examples.

Method	Clean	PGD-20	CW	AA
TRADES	49.56 ± 0.001	22.90 ± 0.0021	19.70 ± 0.0011	16.78 ± 0.001
TRADES + Ours	**51.83** ± 0.003	**25.43** ± 0.003	**20.86** ± 0.001	**18.87** ± 0.002
MART	45.94 ± 0.003	26.02 ± 0.002	21.87 ± 0.001	19.20 ± 0.002
MART + Ours	**46.49** ± 0.004	**26.80** ± 0.001	**22.44** ± 0.002	**20.5** ± 0.006

The results in Table 3 highlight the significant improvements achieved by applying our method to TRADES and MART. For TRADES, we observed enhancements of 3.02% in clean accuracy, 1.45% and 1.47% under PGD-20 and PGD-100, 2.99% under CW, 0.85% under AA, 0.62% under SQUARE, and 1.85% under SPSA attacks. Similarly, for MART, we recorded an improvement of 2.81% in clean accuracy, 1.66% and 1.98% under PGD-20 and PGD-100, 2.39% under CW, 1.51% under AA, 0.28% under SQUARE, and 0.95% under SPSA attacks.

Table 5. Clean and robust accuracies of different margin-based methods on **CIFAR-10** using the **WRN-34-10** model. Results are based on six runs, with the average performance reported along with 95% confidence intervals. The 'Clean' column indicates the accuracy of unperturbed examples.

Method	Clean	PGD-20	PGD-100	CW	AA	SQUARE	SPSA
MMA	86.21 ± 0.003	57.17 ± 0.0021	56.03 ± 0.001	**57.52** ± 0.011	44.57 ± 0.0011	57.12 ± 0.021	59.87 ± 0.011
WAT	85.16 ± 0.003	56.69 ± 0.002	55.13 ± 0.002	54.06 ± 0.014	49.87 ± 0.021	58.89 ± 0.005	60.78 ± 0.002
MAIL	**86.82** ± 0.003	60.38 ± 0.012	**58.68** ± 0.0012	51.48 ± 0.001	47.15 ± 0.001	58.02 ± 0.002	59.23 ± 0.032
GAIRAT	85.39 ± 0.005	**60.59** ± 0.016	58.13 ± 0.0032	45.08 ± 0.014	42.30 ± 0.007	50.98 ± 0.001	52.32 ± 0.004
TRADES + AWA	84.38 ± 0.0012	58.53 ± 0.0021	57.45 ± 0.0011	56.49 ± 0.0024	**54.19** ± 0.0013	**59.79** ± 0.0015	59.87 ± 0.0001
MART + AWA	84.88 ± 0.004	59.30 ± 0.005	57.29 ± 0.001	56.04 ± 0.0021	52.34 ± 0.003	59.09 ± 0.0022	**62.84** ± 0.002

For a more challenging task of classifying TinyImageNet, as presented in Table 4, our method significantly improves both TRADES and MART under all the attacks while maintaining an excellent natural accuracy.

The results in Table 5 demonstrate that the enhanced versions of TRADES (TRADES + AWA) and MART (MART + AWA) significantly outperformed WAT, MAIL, and GAIRAT on stronger attacks like CW and AA. For example, TRADES + AWA surpassed WAT, MAIL, and GAIRAT by 4.32%, 7.04%, and 11.89%, respectively. While MMA, WAT, MAIL, and GAIRAT achieved relatively better performance in terms of natural accuracy, their robustness against stronger attacks was notably poor, highlighting a critical trade-off addressed by our method.

5 Conclusion

We propose an adaptive framework for adversarial training that uses a weight assignment strategy, considering various attack types and perturbation levels. This approach emphasizes the diverse behavior of adversarial examples, allowing the model to focus on more challenging examples during training. As a result, our method significantly boosts robustness against strong adversarial attacks while maintaining a reasonable performance on clean data. This work highlights a step forward in developing efficient and resilient defenses against evolving adversarial threats.

References

1. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search. In: European Conference on Computer Vision, pp. 484–501. Springer (2020)
2. Atsague, M., Nirala, A., Fakorede, O., Tian, J.: A penalized modified huber regularization to improve adversarial robustness. In: 2023 IEEE International Conference on Image Processing (ICIP), pp. 2675–2679. IEEE (2023)
3. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE symposium on security and privacy (sp), pp. 39–57. IEEE (2017)

4. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: International Conference on Machine Learning, pp. 2206–2216. PMLR (2020)
5. Ding, G.W., Sharma, Y., Lui, K.Y.C., Huang, R.: MMA training: Direct input space margin maximization through adversarial training. In: International Conference on Learning Representations (2020)
6. Fakorede, O., Nirala, A.K., Atsague, M., Tian, J.: Vulnerability-aware instance reweighting for adversarial training. Transactions on Machine Learning Research (2023)
7. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. CoRR **abs/1412.6572** (2015)
8. Hou, P., Han, J., Li, X.: Improving adversarial robustness with self-paced hard-class pair reweighting. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 14883–14891 (2023)
9. Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J.E., Stoica, I.: Tune: A research platform for distributed model selection and training. arXiv preprint arXiv:1807.05118 (2018)
10. Liu, F., et al.: Probabilistic margins for instance reweighting in adversarial training. Adv. Neural. Inf. Process. Syst. **34**, 23258–23269 (2021)
11. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018)
12. Uesato, J., O'donoghue, B., Kohli, P., Oord, A.: Adversarial risk and the dangers of evaluating against weak attacks. In: International Conference on Machine Learning, pp. 5025–5034. PMLR (2018)
13. Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., Gu, Q.: Improving adversarial robustness requires revisiting misclassified examples. In: International Conference on Learning Representations (2020)
14. Xie, C., Tan, M., Gong, B., Yuille, A., Le, Q.V.: Smooth adversarial training. arXiv preprint arXiv:2006.14536 (2020)
15. Zeng, H., Zhu, C., Goldstein, T., Huang, F.: Are adversarial examples created equal? a learnable weighted minimax risk for robustness under non-uniform attacks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 10815–10823 (2021)
16. Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., Jordan, M.: Theoretically principled trade-off between robustness and accuracy. In: International Conference on Machine Learning, pp. 7472–7482. PMLR (2019)
17. Zhang, J., Zhu, J., Niu, G., Han, B., Sugiyama, M., Kankanhalli, M.: Geometry-aware instance-reweighted adversarial training. arXiv preprint arXiv:2010.01736 (2020)