Rethinking the Entropy of Instance in Adversarial Training

Minseon Kim[1], Jihoon Tack[1], Jinwoo Shin[1], and Sung Ju Hwang[1,2]

[1]Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea
[2]AITRICS, Seoul, South Korea

***Abstract*—Adversarial training, which minimizes the loss of adversarially-perturbed training examples, has been extensively studied as a solution to improve the robustness of deep neural networks. However, most adversarial training methods treat all training examples equally, while each example may have a different impact on the model's robustness during the course of adversarial training. A couple of recent works have exploited such unequal importance of adversarial samples to the model's robustness by proposing to assign more weights to the misclassified samples or to the samples that violate the margin more severely, which have been shown to obtain high robustness against untargeted PGD attacks. However, we empirically find that they make the feature spaces of adversarial samples across different classes overlap and thus yield more high-entropy samples whose labels could be easily flipped. This makes them more vulnerable to adversarial perturbations, and their seemingly good robustness against PGD attacks is actually achieved by a false sense of robustness. To address such limitations, we propose simple yet effective re-weighting scheme that weighs the loss for each adversarial training example proportionally to the entropy of its predicted distribution to focus on examples whose labels are more uncertain.**

I. Introduction

The deep neural networks often output incorrect predictions even with small perturbations to the input examples [1], despite their impressive performances in a variety of real-world applications. This adversarial vulnerability is a crucial problem in deploying them to safety-critical real-world applications, such as autonomous driving or medical diagnosis. To tackle the adversarial vulnerability problem, various approaches have been proposed to ensure the robustness of the trained networks against adversarial attacks [2, 3, 4, 5, 5, 6, 7, 8].

The most promising approach to improve the adversarial robustness of deep networks is *adversarial training*, which trains the model to minimize the loss on the adversarially perturbed examples. Goodfellow et al. [10], early work on this topic, propose to train the model with samples attacked with the Fast Gradient Sign Method (FGSM), which applies a perturbation to a given clean sample in the direction of the gradient. Following this work, various adversarial defense algorithms have been suggested. For example, Adversarial Training (standard AT) [2] uses a min-max formulation where the examples are perturbed with the loss maximization objective with the Projected Gradient Descent (PGD) attack. Further, TRADES [3] demonstrates the trade-off between clean accuracy and robustness, and proposes to minimize the Kullback-Leibler divergence between the prediction on the clean example and its adversarial counterpart, to achieve robustness against adversarial perturbations.

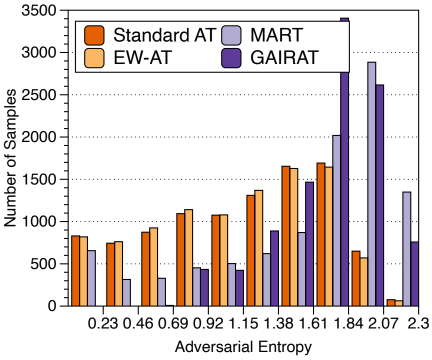


Figure 1: **Distribution of the entropy for the adversarial samples obtained with different re-weighting methods.** Previous re-weighting approach (MART [5], GAIRAT [9]) induces a large number of high entropy examples than standard AT [2] which may cause vulnerability against the targeted attack.

In natural image classification training, some works have shown that only a small portion of examples from the training set contribute to the generalization performance [11], where each sample has a different impact on the model's final performance. Similarly, it is also natural to assume that some training examples are more important than others, in enhancing the adversarial robustness of the adversarially trained model.

Based on this intuition, previous studies suggest identifying such robustness-critical instances, to assign more weights on them during adversarial training. To name a few, Wang et al. [5] argue that misclassified *clean* samples are more important in achieving robustness and impose larger KL-divergence regularization on them (MART). On the other hand, Zhang et al. [9] suggests assigning more weights to examples that were close to the decision boundary before the adversarial attack (GAIRAT). These methods have shown to achieve impressive robustness against untargeted PGD attacks.

However, we discover that the re-weighting scheme in MART and GAIRAT creates a false sense of robustness. In short, while they appear

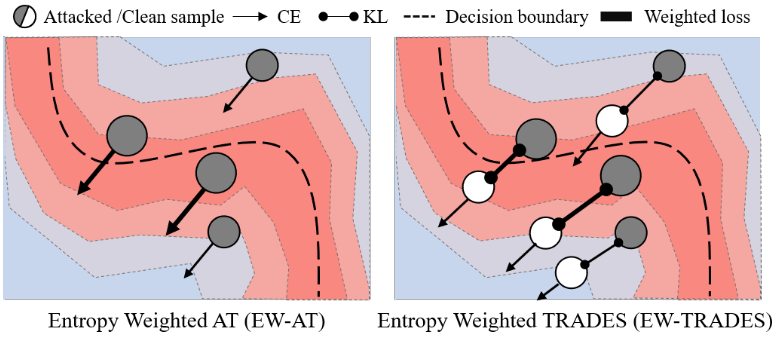

Attacked /Clean sample
CE
KL
Decision boundary
Weighted loss
Entropy Weighted AT (EW-AT)
Entropy Weighted TRADES (EW-TRADES)

Figure 2: **Overview of EWAT.** EWAT weighs more on the uncertain examples which have large entropy (red) while adjusting relatively low weights on the low entropy examples (blue). For standard AT, weighting is applied on the cross-entropy loss. For TRADES, weighting is applied on the Kullback-Leibler loss.

more robust against untargeted PGD attacks, they become more vulnerable to other types of adversarial attacks, such as logit scaling attack [12] and AutoAttack [13], compared to standard AT. We further show that these re-weighting schemes make the feature spaces of adversarial samples belonging to different classes overlap (Figure 3) and thus increase the entropy of the adversarially-perturbed training examples (Figure 1).

Such high-entropy samples are more vulnerable to targeted adversarial attacks, since their predicted labels are uncertain, and could be flipped with less effort. Based on this observation, we propose a simple yet effective re-weighted adversarial training method that improves the model's robustness in re-weighting against both untargeted attack and targeted attack, which assigns a weight to each adversarially-perturbed sample based on the entropy of its predicted distribution. Specifically, our method assigns larger weights to training examples with high entropies (Figure 2).

The experimental validation of our re-weighted adversarial training scheme, named Entropy-Weighted Adversarial Training (EWAT), verifies that it improves the robustness of the existing adversarially-trained models on multiple benchmark datasets (MNIST, CIFAR10, and CIFAR100). Our instance-weighting scheme is simple to implement, compute, and use, while improving the robustness without any additional computational cost. In summary, the contributions of this paper are as follows:

- We show the previous re-weighting schemes are suboptimal and induce vulnerability against the AutoAttack and logit scaling attack compared to standard AT.
- We discover evidence that previous re-weighting schemes make samples gather around the decision boundary with high entropy that leads to a vulnerability against strong attacks.
- Based on these observations, we further propose a surprisingly simple, yet effective **entropy weighting** scheme that can enhance the adversarially trained model's robustness, which weighs the loss of adversarial samples with respect to their entropy.

II. Related Work

a) Adversarial robustness: Szegedy et al. [1] firstly showed that deep neural networks for image classification are vulnerable to imperceptible *small perturbations* applied to input images. To achieve robustness against such adversarial attacks, Goodfellow et al. [10] proposed the Fast Gradient Sign Method (FGSM), which perturbs a target sample to its gradient direction to increase its loss. Then, they proposed an adversarial training objective that aims to minimize the loss of the perturbed samples as well as clean samples, which have shown to be effective against such adversarial attacks. Follow-up works [14, 4, 15] proposed a variety of gradient attacks that are stronger than FGSM that can be used for adversarial training, and Madry et al. [2] proposed a minimax formulation to minimize the loss of adversarial examples, which are perturbed to maximize its loss with the projected gradient method. After a surge of interest in the adversarial robustness of neural networks, various defense mechanisms [16, 17, 18] have been proposed to defend against such adversarial attacks. However, Athalye et al. [19] showed that many of them except *standard AT*, rely on gradient masking, which results in obfuscated gradient in the representation space, and are highly vulnerable to stronger attacks that circumvent it. TRADES [3] propose to minimize the Kullback-Leibler divergence (KL) between a clean example and its adversarial counterpart, to enforce consistency between their predictions, and further show that there is a theoretical trade-off between the clean accuracy of a model and its robustness. Recently, using additional unlabeled data [20, 7] or using an additional attack mechanism [21] have been proposed. To utilize the additional data, Carmon et al. [20] propose to use Tiny ImageNet [22] as pseudo-label to learn more rich representation of CIFAR10 [23] dataset that could lead to robust model (RST). Gowal et al. [7] proposes to use generative models to artificially increase the size of the original training set and improve adversarial robustness with those additionally generated images. Wu et al. [21] propose a double-perturbation mechanism that conducts additional adversarial weight perturbation (AWP) with conventional adversarial training. Recently, to overcome the adversarial overfitting problem, several works [8, 24, 25] have been proposed. Among them, Rebuffi et al. [8] that uses data augmentation techniques combined with model weight averaging outperforms the most.

b) Instance-wise weighting for adversarial training: While successful in general, none of the aforementioned works consider the varying impact of samples on adversarial robustness. A recent work, Misclassification Aware adveRsarial Training (MART) [5], focuses on this problem and proposes to put more weights on the misclassified clean samples for the KL-divergence regularization, achieving state-of-the-art robustness against untargeted PGD attacks. Furthermore, another recent work, Geometry-aware Instance-Reweighted Adversarial Training (GAIRAT) [9] proposed a method with a similar motivation, which weighs the adversarial loss of each sample based on the clean sample's distance to the decision boundary. GAIRAT also achieves impressive performance against untargeted PGD attacks. However, these methods make increase the entropies of the adversarially perturbed samples and thus make the model to be more vulnerable against targeted attacks, such as AutoAttack [13] and the logit scaling attack [12]. We observe that both re-weighting methods for the adversarial training largely increase the entropies of the perturbed examples, which makes the samples more vulnerable as their predictions are easier to alter.

c) Adversarial attacks: Most of the adversarial defense mechanisms have been broken with stronger attacks that were not aware of at the time they were first introduced. Athalye et al. [19] is an important work that has helped many researchers to explore means to achieve fundamental robustness rather than take advantage of a false sense of security created with the obfuscated gradients. To verify the robustness, several adversarial attacks [15, 14, 2] based on gradients have been proposed. Recently, Croce and Hein [13] proposed an ensemble attack that consists of four different attacks (AutoAttack), namely untargeted APGD-CE, targeted APGD-DLR, FAB [26] and square attack [27]. APGD-CE and APGD-DLR are step-size-free variants of the PGD attack. AutoAttack revealed that most defense methods are actually more vulnerable than TRADES if the attacker carries out a targeted attack, which is a more viable scenario in real-world cases.

A. Preliminaries

In this section, we first recap the adversarial training (standard AT) [2], TRADES [3] and previous instance weighting methods for adversarial training (MART [5], GAIRAT [9]).

Let us denote the dataset $\mathcal{D} = \{X, Y\}$, where $x \in X$ is a training example and $y \in Y$ is its corresponding label, and a supervised learning model $f_\theta : X \to Y$ where θ is the set of the parameters of the model. Given such a dataset and a model, *adversarial attacks* aim toward finding the worst-case examples by searching for the perturbation for each example that maximizes the loss within a certain radius from it (e.g., norm balls). We can define such adversarial ℓ_∞ attacks as follows:

$$\delta^{t+1} = \Pi_{B(0,\epsilon)}\Big(\delta^t + \alpha \mathtt{sign}\Big(\nabla_{\delta^t}\mathcal{L}_{\mathtt{CE}}\big(f(\theta, x+\delta^t), y\big)\Big)\Big), \tag{1}$$

where $B(0,\epsilon)$ is the ℓ_∞ norm-ball with radius ϵ, Π is the projection function to the norm-ball, α is the step size of the attacks and $\mathtt{sign}(\cdot)$ is the sign of the vector. Further, the perturbation δ is the accumulated $\alpha\mathtt{sign}(\cdot)$ over multiple attack iterations t, and $\mathcal{L}_{\mathtt{CE}}$ is the cross-entropy loss. In the case of Projected Gradient Descent (PGD) [2], the attack starts from a random point within the epsilon ball and performs t gradient steps, to obtain a perturbed sample x^{adv}.

The most straightforward way to defend against such adversarial attacks is to minimize the loss of adversarial examples, which is often called *adversarial training*. The standard AT framework proposed by Madry et al. [2] solves the following min-max problem where δ is the perturbation of the adversarial example of the given input x, and y is its target class label. Then the loss is:

$$\mathcal{L}_{\mathtt{AT}} = \max_{\delta \in B(x,\epsilon)} \mathcal{L}_{\mathtt{CE}}\big(f(\theta, x+\delta), y\big). \tag{2}$$

Another popular algorithm for adversarial training, TRADES [3], suggests minimizing the Kullback-Leibler (KL) divergence between a clean example and its adversarial perturbation, to enforce consistency between their predictions while using cross-entropy loss on clean samples as follow:

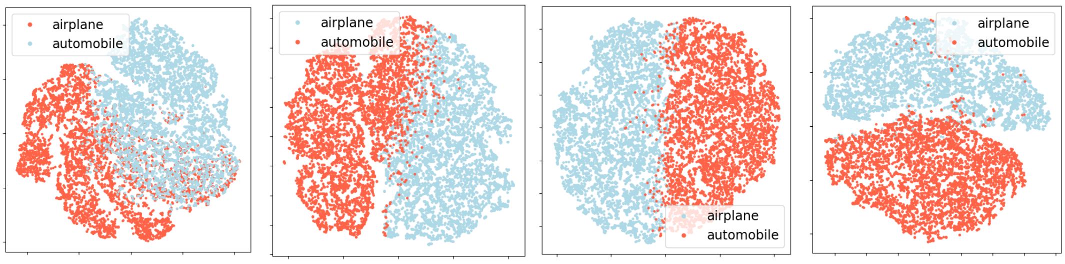
$$\begin{aligned}\mathcal{L}_{\mathtt{TRADES}} &= \mathcal{L}_{\mathtt{CE}}\big(f(\theta, x), y\big) \\ &\quad + \beta \max_{\delta \in B(x,\epsilon)} \mathcal{L}_{\mathtt{KL}}\big(f(\theta, x) || f(\theta, x+\delta)\big),\end{aligned} \tag{3}$$

where $\mathcal{L}_{\mathtt{KL}}$ is KL divergence loss and β is a parameter to control the trade-off between clean and adversarial performance.

a) Instance weighting schemes for adversarial training: Recently, MART [5] proposed a new weighted adversarial training framework with the boosted cross entropy loss and the weighted KL divergence loss. The boosted cross-entropy loss maximizes the $1-$ the second highest class probability, to increase the margin of the classifier. The weighted KL divergence loss assigns higher weights to the KL-divergence term, for the samples that are misclassified before applying the adversarial perturbations. The loss of MART is defined as follows:

$$\begin{aligned}\mathcal{L}_{\mathtt{MART}} &= \mathcal{L}_{\mathtt{AT}} - \log\big(1 - \max_{k \neq y} p_k(f(\theta, x+\delta))\big) \\ &+ \lambda \mathcal{L}_{\mathtt{KL}}\big(f(\theta, x) || f(\theta, x+\delta)\big)\big(1 - p_y(f(\theta, x))\big),\end{aligned} \tag{4}$$

where p_k is the probability of k^{th} class.



(a) t-SNE of GAIRAT (b) t-SNE of MART (c) t-SNE of standard AT (d) t-SNE of EW-AT

Figure 3: **Visualization of the embeddings of adversarial examples from each model.** All models are trained with PreActResNet18. We only visualize train examples from airplane and automobile classes in the CIFAR10.

A recent approach, GAIRAT [9], suggests a loss weighting scheme based on the clean sample's distance to the decision boundary:

$$\mathcal{L}_{\texttt{GAIRAT}} = \frac{\gamma}{\sum_{i=0}^{B}\gamma}\mathcal{L}_{\texttt{AT}}, \quad \gamma = \frac{\Big(1+\texttt{tanh}\big(\psi + 5(1-2\kappa(x,y)/K)\big)\Big)}{2}, \tag{5}$$

where B is batch size, and $\kappa(x, y)$ is geometric distance of a data point (x, y). $\kappa(x, y)$ is calculated as a total number of attack steps minus the least number of necessary attacked steps to change the label y of x during the PGD attack Eq. (1). ψ is a constant hyperparameter and K is the total attack steps. Therefore, if the sample is already far from the decision boundary, those samples are not used during the training. This causes the highly under-confident model and induces vulnerability against AutoAttack [13] and logit scaling attack [12].

III. Unequal Importance of Each Sample in Adversarial Training

In this section, we elaborate on what should be considered in instance-wise weighted adversarial training, in order to consider the unequal importance of each sample to the adversarial robustness of the model. Moreover, we also show the adversarial vulnerability of the previous instance weighting schemes for adversarial training. To be precise, this vulnerability does not come from any types of obfuscated gradients introduced in the Athalye et al. [19], but it also creates a false sense of robustness.

To design the weighting scheme in adversarial training, we first have to define two conditions.

1) Which criteria should we use to evaluate the importance of samples during adversarial training?

2) How can we assign attention/weight to differently contributed samples?

The previous works answered both questions with their intuitions and verified their hypotheses with the empirical results on the PGD attacks. MART [5] argues that the predictions on the non-perturbed samples are important criteria to measure the sample-wise importance in adversarial training. Thus, MART assigns more adversarial attention to KL loss that has low confidence before perturbation Eq. (4). GAIRAT [9], on the other hand, hypothesizes that the number of steps to perturb the given sample is an important measure of its importance in adversarial training. Thus, GAIRAT assigns more attention to the adversarial samples that violate the margin more, as in Eq. (5).

Table I: **Vulnerability loophole in the previous re-weighted adversarial training methods.** We validate MART and GAIRAT against logit scaling attack (LS) [12] with α = 10 and the AutoAttack (AA) [13] with $\epsilon = 0.031$. All models are trained with PreActResNet18 architecture.

Method	PGD	LS	AA
GAIRAT [9]	55.16	31.78	22.37
MART [5]	**57.08**	48.70	46.79
standard AT [2]	53.96	51.26	48.16
+ Ours (EW-AT)	53.49	51.81	49.20
TRADES [3]	53.95	50.10	49.30
+ Ours (EW-TRADES)	53.83	50.13	**49.90**

However, we empirically find that the performance achieved by weighting only improves untargeted PGD attacks. Yet, both methods achieve lower performance compared to standard AT against logit scaling attack [12] and AutoAttack [13] (see Table I). Thus, while they appear to be more robust than standard AT, they are actually more vulnerable. We further examine why the previous weighting schemes are vulnerable to non-PGD attacks, by visualizing the t-SNE embeddings of the adversarially perturbed training samples in Figure 3. As shown in Figure 3, adversarial examples generated by GAIRAT and MART for two different classes have large overlaps, while the t-SNE of the adversarial samples trained with standard AT shows clear separation.

This is because the previous weighting schemes make the model only focus on certain samples that they deem difficult, making the prediction on others more uncertain. This will make the entropy of the predictive distribution of such samples to be high. This is shown in Figure 4, where MART and GAIRAT have more than double the number of 'high-entropy' samples (> 1.5). It is evident that

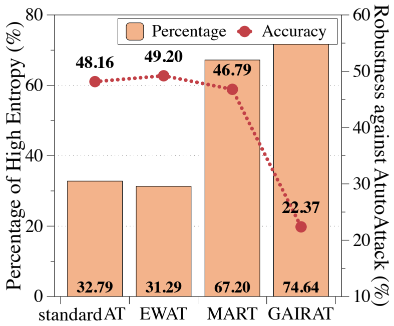


Figure 4: Percentage of high entropy samples (>1.5) in the test set from standard AT, EW-AT, MART, and GAIRAT which shows a correlation to the performance of AutoAttack.

such high-entropy samples will be more prone to make wrong predictions if they are perturbed only a little to a manifold of another class that is predicted high, thus making them more vulnerable to targeted attacks, such as AutoAttack. This suggests that if we can minimize the existence of such high-entropy samples, the model's robustness will be improved.

We want to emphasize that despite the philosophy of MART, and GAIRAT (i.e., specific examples contribute more to the adversarial robustness) is a proper suggestion for only untargeted PGD attacks, but those re-weighting can lead to a vulnerability against another type of attack. We presume that misconstrued re-weighting formulation leads to a false sense of robustness. Therefore, we further propose a legitimate strategy to re-weight the adversarial training that is robust against strong attacks.

IV. Entropy in Re-weighted Adversarial Training

We now describe our observation and additionally proposed method to overcome previous re-weighting approach, Entropy-Weighted Adversarial Training (EWAT). EWAT weighs the loss of each adversarial example based on its entropy of the predictive distribution (Algorithm 1).

In the previous section, we showed that GAIRAT and MART make many of its adversarial samples have relatively high entropies compared to adversarial samples from standard AT, as shown in Figure 1, and that this makes them vulnerable against AutoAttack [13] (Figure 4). They have created more vulnerable samples while trying to focus on samples they deem as important, by assigning relatively smaller weights to other samples, making the prediction confidence on them low.

Further, we also observe the percentage of high entropy samples in the correctly classified sets from several conventional models (Figure 5[1][2][3]). Notably,

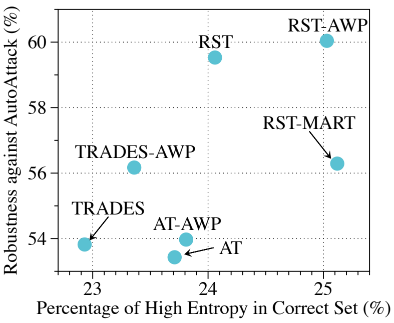


Figure 5: Correlation between high entropy samples and robustness against AutoAttack. The percentage is calculated as a portion of the top 30% high entropy in correctly classified samples.

a model that is more robust against AutoAttack can easily classify the samples that are large entropy. All of this empirical evidence suggests that the ratio of high entropy samples is highly related to the model's robustness. Also, the entropy is a more direct measure of a sample's robustness, unlike its distance to the margin (GARIAT) or whether the sample is predicted incorrectly (MART), since a high entropy sample's predicted label could be altered more easily. Thus, we propose to consider the *entropy* of each adversarially perturbed sample as a criterion to measure its vulnerability and propose a loss weighting scheme based on the entropy.

a) Entropy: The *entropy* $\mathcal{E}$ is a measurement of the state of uncertainty and randomness. The entropy for an adversarial sample $x^{\mathtt{adv}}$ for classification tasks can be defined as follows:

$$\mathcal{E}(\theta, x^{\mathtt{adv}}) := \\ -\sum_{j=1}^{\mathtt{C}} p_j(f(\theta, x^{\mathtt{adv}})/\tau)\log\left(p_j(f(\theta, x^{\mathtt{adv}})/\tau)\right), \tag{6}$$

where p_j stands for the j^{th} class probability of $f(\theta, x^{\mathtt{adv}})$, $\mathtt{C}$ is the number of classes and τ is temperature scaling factor where we set as 1. We can control τ to affect the $\mathcal{E}$ by making the predictive distribution to be sharper or smoother.

b) Entropy-based sample weighting: We now propose an additional entropy-weighted loss term for adversarial training, which weighs each adversarial example by its entropy.

The entropy value of each training example continuously changes during the course of training. This is beneficial since the weighting changes adaptively, such that it focuses on the most uncertain samples at each iteration. However, one caveat here is that entropies of all samples will go low as the model trains on, which will simply have small or no effects on weighting. Since this will be the same as non-weighted training at the end, we normalize the entropy weights with the batch mean of the entropy at each iteration. Formally, for a given batch of adversarial examples $\mathcal{B} := \{(x_i^{\mathtt{adv}}, y_i)\}_{i=1}^m$, we define the entropy weighting ($w_i^{\mathtt{ent}}$) for a given

[1] The pre-trained network of RST [20] is from https://github.com/yaircarmon/semisup-adv

[2] The pre-trained network of AWP [21] is from https://github.com/csdongxian/AWP

[3] The pre-trained network of MART-RST [5] is from https://github.com/YisenWang/MART

Algorithm 1 Entropy weighted adversarial training for standard AT [2]

```
Input: Dataset $\mathcal{D}$, parameters of model $\theta$, model $f$, number of epochs T, batch size m, number of batches M, Cross-entropy loss $\mathcal{L}_{\texttt{CE}}$, number of classes $\texttt{C}$
for epoch = 1, $\cdots$ , T do
    for mini-batch = 1, $\cdots$, M do
        Sample mini-batch from training set ($\mathcal{D}$): $\{(x_i, y_i)\}_{i=1}^{m}$
        Generate adversarial examples $x_i^{\texttt{adv}}$
        Calculate entropy $\mathcal{E}$ for weighting
            $\mathcal{E}(\theta, x_i^{\texttt{adv}}) = -\sum_j^{\texttt{C}} p_j(f(\theta, x_i^{\texttt{adv}})) \log(p_j(f(\theta, x_i^{\texttt{adv}})))$
            $\eta = \frac{1}{m}\sum_{i=1}^{m} \mathcal{E}(\theta, x_i^{\texttt{adv}})$
            $w_{\texttt{ent}}^{i} = \mathcal{E}(\theta, x_i^{\texttt{adv}})/\eta$
        Calculate total loss
            $\mathcal{L}_{\texttt{EW-AT}} = \mathcal{L}_{\texttt{AT}} + w_{\texttt{ent}} \cdot \mathcal{L}_{\texttt{CE}}(f(\theta, x^{\texttt{adv}}), y)$
        Take gradient descent with respect to the model parameters
    end for
end for
```

instance $x_i^{\texttt{adv}}$ as follows:

$$w_i^{\texttt{ent}} := \frac{1}{\eta} \cdot \mathcal{E}(\theta, x_i^{\texttt{adv}}), \tag{7}$$

where $\eta := \sum_{x^{\texttt{adv}} \in \mathcal{B}} \mathcal{E}(\theta, x^{\texttt{adv}})/B$ is the batch mean of the predicted entropy. The final objective consisting of the original adversarial training loss and entropy weighted cross-entropy loss is as follows:

$$\begin{aligned} \mathcal{L}_{\texttt{Ent-AT}} &:= w^{\texttt{ent}} \cdot \mathcal{L}_{\texttt{CE}}\big(f(\theta, x^{\texttt{adv}}), y\big), \qquad (8)\\ \mathcal{L}_{\texttt{EW-AT}} &:= \mathcal{L}_{\texttt{AT}} + \mathcal{L}_{\texttt{Ent-AT}} \\ &= (1 + w^{\texttt{ent}}) \cdot \mathcal{L}_{\texttt{CE}}\big(f(\theta, x^{\texttt{adv}}), y\big). \end{aligned}$$

For the TRADES loss, the overall weighted loss is as follows:

$$\begin{aligned} \mathcal{L}_{\texttt{Ent-TRADES}} &:= w^{\texttt{ent}} \cdot \mathcal{L}_{\texttt{KL}}\big(f(\theta, x) || f(\theta, x^{\texttt{adv}})\big), \\ \mathcal{L}_{\texttt{EW-TRADES}} &:= \mathcal{L}_{\texttt{TRADES}} + \mathcal{L}_{\texttt{Ent-TRADES}} \\ &= \mathcal{L}_{\texttt{CE}}\big(f(\theta, x), y\big) \\ &\quad + (\beta + w^{\texttt{ent}}) \cdot \mathcal{L}_{\texttt{KL}}\big(f(\theta, x) || f(\theta, x^{\texttt{adv}})\big). \end{aligned} \tag{9}$$

V. Experiments

In this section, we first validate our entropy-weighted adversarial training against the PGD attack, logit scaling attack [12], and AutoAttack [13]. Then, we examine the effect of temperature scaling (Section V-C), which is the only parameter our model has. Moreover, we report the results of the generality of our model on multiple benchmark datasets and utilizing unlabeled data (Section V-D0a).

A. Experimental Setup.

Dataset description. For experiments, we use CIFAR10, CIFAR100, and MNIST. CIFAR10 and CIFAR100[4] consist of 50,000 training images and 10,000 test images with 10 and 100 classes, respectively. All CIFAR images are 32×32×3 resolution (width, height, and channel). MNIST dataset contains hand-written digits, ranging from 0 to 9. MNIST contains a training set of 60,000 examples and a test set of 10,000 examples, where each image has 28×28×1 resolution (width, height, and channel). For the additional dataset in Table 5, we utilize the 500K unlabeled data from TinyImages (with pseudo-labels)[5]. The pickle file consists of 500K unlabeled TinyImageNet. TinyImageNet has 100,000 training with 200 image classes.

Training detail.

- **MNIST.** For all methods compared, we train the network with ℓ_∞ attacks with the attack strength of $\epsilon = 0.3$ and the step size of $\alpha = 0.01$, with the number of inner maximization iterations set to $K = 40$. For optimization, we train every model for 100 epochs using the SGD optimizer with the weight decay of $1\text{e}{-4}$ and the momentum of 0.9. As for learning rate scheduling, we use the decay of 0.1 at the 20^{th} and 40^{th} epoch with the initial learning rate of 0.01.
- **CIFAR.** For all methods, we train the network with ℓ_∞ attacks with the attack strength of $\epsilon = 8/255$ and the step size of $\alpha = 2/255$, with the number of inner maximization iteration set to $K = 10$. For the optimization, we train every model for 100 epochs using the SGD optimizer with the weight decay of $5\text{e}{-4}$ and the momentum of 0.9. For learning rate scheduling, we use the decay of 0.1 at the 100^{th} and 105^{th} epoch with the initial learning rate of 0.1.
- **Hyperparameters.** When setting the hyperparameters for baselines, we follow their official settings in the original papers. For TRADES [3], we set β as 6.0, and for EW-TRADES we set β as 5.5. In MART [5], we set λ as 6.0. In GAIRAT [9], we set ψ as -1.0.

Evaluation detail.

- **ℓ_∞ attack.** For all ℓ_∞ attacks used in the test phase, we use the attack strength of $\epsilon = 8/255$ and the step size of $\alpha = 2/255$ with the number of inner maximization iteration set to $K = 10$ for PGD10. For PGD20 We use the $\alpha = \epsilon/10$ with $K = 20$, respectively.
- **Logit scaling attack.** We further test our EWAT against Logit scaling attack [12] that the previous instance-weighted method is vulnera-

[4] The full dataset of CIFAR can be downloaded at http://www.cs.toronto.edu/~kriz/cifar.html.

[5] The unlabeled pickle file can be downloaded at https://github.com/yaircarmon/semisup-adv.

Table II: **Results against ℓ_∞ attack in CIFAR10 with PreActResNet18.** Clean denotes the accuracy on natural images. The robust accuracy against AutoAttack (AA) is calculated against $\epsilon = 0.031$. We report the mean performance and standard deviations of 5 multiple random seed runs.

Method	Clean	AA
standard AT [2]	82.90 (± 0.43)	48.69 (± 0.41)
+ Ours (`EW-AT`)	82.46 (± 0.56)	**49.18 (± 0.12)**
TRADES [3]	81.55 (± 0.21)	49.48 (± 0.15)
+ Ours (`EW-TRADES`)	81.02 (± 0.34)	**49.83 (± 0.25)**

ble. Logit scaling attack is multiplying constant in the logit with α as follow:

$$\delta^{t+1} = \Pi_{B(0,\epsilon)} \Big(\delta^t + p\,\texttt{sign}\big(\nabla_{\delta^t}\mathcal{L}_{\texttt{CE}}(\alpha f(\theta, x+\delta^t), y)\big)\Big), \tag{10}$$

where $B(0, \epsilon)$ is the ℓ_∞ norm-ball with radius ϵ, Π is the projection function to the norm-ball, p is the step size of the attacks and $\texttt{sign}(\cdot)$ is the sign of the vector. We set α as 10 for testing in Table 1. When α is 1, the logit scaling attack is the same as the PGD attack.

- **AutoAttack.** We further test our EWAT against AutoAttack (AA) [13]. AutoAttack is an ensemble attack that consists of four different attacks (APGD-CE, APGD-T, FAB-T [26], and Square [27]). APGD-T and FAB-T are targeted attacks and Square is a black box attack[6].

Resource description. All experiments are conducted with a single GPU (NVIDIA RTX 2080 Ti), except for the TRADES experiments with WideResNet in Table III. For WideResNet TRADES, two GPUs (NVIDIA RTX 2080 Ti) are used. All experiments are processed in Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz.

B. Validation of Entropy Weighted Adversarial Training

a) Against standard attacks: Our entropy weighted adversarial training improves upon the baselines, outperforming standard AT model by 0.51%, and TRADES by 0.45%, with the PreActResNet18 model (Table II) against AutoAttack. Moreover, with WideResNet34-10 (Table III), our weighting achieves the performance gains of 0.44% and 0.76% over standard AT and TRADES against AutoAttack, respectively. Considering that TRADES is considered as powerful by making 1% improvement over the standard AT, this is a meaningful improvement of the robust accuracy. On the contrary, GAIRAT and MART achieve lower robustness over standard AT against AutoAttack, although they attempted to improve upon the standard AT model by the proposed weighting schemes.

In practice, we cannot assume that the attacker will only use a single type of attack, and thus the most important measure of robustness is the robust accuracy against the strongest attack, which is the AutoAttack in this case. Our EWAT shows high robustness against this worst-case attack, although it also obtains comparable performance to baselines', against the PGD attacks.

b) Against logit scaling attack and AutoAttack: Previous weighting methods, MART and GAIRAT, suffer from low robustness against logit scaling attack [12] and AutoAttack. However, our model demonstrates improved robustness against both types of attacks (Table I) and does not suffer from the vulnerability loophole, unlike the existing loss weighting schemes.

c) Verification of obfuscated gradient: We test our model against ℓ_∞ attack with larger PGD steps $K = 10, 100$ to check whether the obfuscated gradient occurs during the PGD attacks (Table III). We vary the step size $\alpha = \epsilon/(K/2)$. Our model retains robustness even with the different number of steps. While EWAT has no mechanism for gradient obfuscation, this empirical result further shows evidence that it does not benefit from the obfuscated gradient on PGD.

C. Effect of temperature scaling

We further examine the effects of simple temperature scaling, which is the only hyperparameter EWAT has (and is set to 1 by default), as it affects the entropy by making the predictive distribution sharper or smoother. We report the effect of different temperature values on our method's robust accuracy (Table IV). We observe that increasing the temperature value, which increases the overall entropy of all samples, improves the performance of EWAT. However, increasing the temperature to an overly high value will result in almost equal weights and make the weighting scheme meaningless. Therefore, in CIFAR10, $\tau = 5$ is the optimal constant for the highest robustness.

D. Generality of EWAT

a) Results on multiple benchmarks datasets: We validate our methods on multiple benchmark datasets. In Table V, EWAT consistently improve upon standard AT and TRADES against AutoAttack on MNIST [28], and CIFAR100 [23]. Compared to the margin-based methods, our model does not require any warm-up epochs for weighting instances, even on larger datasets such as CIFAR100. This is because it relies on entropy, which can be computed easily and is well defined regardless of the training stage, unlike other values, such as distance to the (estimated) margins. Moreover, our methods work better on a larger dataset (CIFAR100) with more number of classes (100), on which the model's predictions could be more uncertain, due to the increased confusion across the classes, than on a smaller dataset (MNIST) with few classes (10).

[6] AutoAttack https://github.com/fra31/auto-attack

Table III: **Results against ℓ_∞ attack in CIFAR10 with WideResNet34-10.** Clean denotes the accuracy on natural images. Best and Last stand for the best robust accuracy, and the accuracy at the last epoch, against PGD with $\epsilon = 8/255$, respectively. For the AutoAttack (AA), we use the threat model with $\epsilon = 0.031$.

	Last		Best		
Method	Clean	PGD10	Clean	PGD100	AA
GAIRAT [9]	85.24	52.97	86.16	57.37	42.28
MART [5]	83.72	55.73	82.85	59.30	51.39
standard AT [2]	87.38	54.21	85.84	**56.17**	52.07
+ Ours (`EW-AT`)	86.97	**54.69**	85.39	55.54	**52.51**
TRADES [3]	85.62	57.32	85.62	57.54	53.82
+ Ours (`EW-TRADES`)	83.11	**57.84**	82.54	**58.27**	**54.58**

Table IV: **Temperature scaling.** τ is parameter for temperature scaling. The reported results are robust accuracies against the ℓ_∞-AutoAttack (AA) with $\epsilon = 0.031$ on CIFAR10.

Method	τ	Clean	AA
standard AT	-	81.62	48.16
	$\tau = 0.5$	82.92	48.85 (+0.69)
	$\tau = 1.0$	83.01	49.20 (+1.04)
Ours (`EW-AT`)	$\tau = 5.0$	81.69	**49.31 (+1.14)**
	$\tau = 10.0$	81.88	49.26 (+1.00)
	$\tau = 20.0$	82.93	49.09 (+0.93)

E. Results of other types of attacks

We test our model against CW-ℓ_∞ [15] and DeepFool [14] attacks using the Adversarial Robustness Toolbox (v1.7) [7] and Foolbox (v3.0) [8]. We set the epsilon to 8/255 for both attacks. These are weak attacks and the robust accuracy against them is less meaningful, as shown with the PGD experiments with the baselines MART and GAIRAT. Our main focus is rather on the defense against a stronger ensemble attack, AutoAttack, since what matters more for adversarial robustness is the worst-case performance as the attacker can use any arbitrary attacks.

F. Utility of additional unlabeled data

Several recent works have shown that utilizing additional unlabeled data can improve the adversarial robustness [20]. To test our method under this scenario, we follow the settings from Carmon et al. [20] by utilizing the 500K unlabeled Tiny ImageNet dataset [22] as the pseudo-label dataset, to adversarially train a WideResNet 28-10 [29]. With the unlabeled data, our approach also obtains improved performance on clean examples and ℓ_∞-attacked images than the previous weighting scheme, and RST against AutoAttack (Table VIII). One thing to note is that using unlabeled data also enhances the robustness of MART [5] and GAIR-RST [9][9] against AutoAttack.

VI. Why previous re-weighting method is vulnerable against AutoAttack while our re-weighting is robust?

We empirically find that our re-weighting can achieve better performance against AutoAttack while the previous re-weighting can not. We suspect there mainly exist two reasons, (a) informative standard (i.e., entropy), and (b) correct formulation (i.e., giving large weight to high entropy samples).

Entropy, margin, confidence, and probability share similar characteristics in deep neural network training. When the sample is close to the decision boundary, its entropy is large, the margin is small, confidence is low, and probability is small, respectively. However, margin, confidence, and probability contain information between only two primary classes, the top 1 and top 2 classes. On the other hand, entropy contains information between all classes which has the benefit to have more information to use as weight than other standards.

Moreover, we presume previous weighting formulations have misconstruction standards for already misclassified samples. Our weighting scheme shares the same philosophy with the samples that are still in the correct class cluster which is weighing more on difficult samples that are close to the decision boundary. However, there is a difference in weighting design for already misclassified attacked samples. Previous re-weighting focused on the samples that are far from the decision boundary which largely violated the boundary. However, our re-weighting focus on samples that are close to the decision boundary (i.e., large entropy samples). We believe focusing on samples that have more probability to learn during the training is a better strategy while focusing on largely violated samples.

VII. Does the re-weighting method exhibit good calibration and confidence?

To further evaluate our model, we test models' calibration using the expected calibration error (ECE) method proposed by Guo et al. [30]. As shown in Table VII, our model demonstrates the best calibration performance on adversarial examples. However, compared to the AT model, our model does not appear to be as well calibrated on clean examples. We believe this may be due to the fact

[7] Adversarial Robustness Toolbox https://adversarial-robustness-toolbox.readthedocs.io/en/latest/#

[8] Foolbox, https://github.com/bethgelab/foolbox

[9] In the official code in GAIRAT, GAIR-RST* is trained with RST attack and CW attack which is not a fair comparison to RST, MART or Ours. Therefore, we re-trained GAIR-RST with RST's official setting.

Table V: **Results against ℓ_∞ attack in MNIST, and CIFAR100.** Clean denotes the accuracy of the natural images. For the AutoAttack (AA), we use the threat model with $\epsilon = 0.031$.

	MNIST		CIFAR100	
Method	Clean	AA	Clean	AA
standard AT [2]	98.74	**88.51**	55.72	24.09
+ Ours (EW-AT)	98.97	88.43 (-0.08)	57.71	**24.57 (+0.48)**
TRADES [3]	98.37	89.30	57.78	25.06
+ Ours (EW-TRADES)	97.46	**89.71 (+0.41)**	55.42	**25.66 (+0.60)**

Table VI: **Results against CW-ℓ_∞ and DeepFool in CIFAR10.** We test our model against CW-ℓ_∞ and DeepFool attacks using the Adversarial Robustness Toolbox (v1.7) and Foolbox (v3.0).

Method	CW-ℓ_∞	DeepFool
standard AT	51.21	53.47
+ Ours (EW-AT)	51.52	**53.58**
TRADES	51.13	57.20
+ Ours (EW-TRADES)	51.48	**57.43**

Table VII: **Results of calibrated performance.** We test our model to see if weighting scheme somehow calibrated the model. We report expected calibration error (ECE) of clean examples and adversarial examples and mean confidence of each models.

	Clean ECE	Adversarial ECE	Mean Confidence
standard AT	14.27	5.04	0.70
EW-AT	15.76	3.75	0.74
MART	23.75	11.04	0.59
GAIRAT	43.87	25.77	0.38

that our model places greater weight on adversarially high entropy examples, which could potentially lead to better calibration only on adversarial examples. In addition, we found that our model has relatively higher average confidence compared to the baseline models (Table VII). The higher average confidence may contribute to better robustness against targeted attacks, as it becomes more difficult to maximize the confidence of high confidence instances to different classes.

Table VIII: **Results of using unlabeled data against ℓ_∞ AutoAttack in CIFAR10.** * indicate performance which is calculated by official checkpoints provided by Carmon et al. [20], Wang et al. [5] and Zhang et al. [9]. GAIR-RST w/o * is reproduced results with same condition as RST. Clean denotes the accuracy of the natural images. For the AutoAttack (AA), we use the threat model with $\epsilon = 0.031$.

Method	Clean	AA
RST* [20]	89.69	59.38
+ MART* [5]	87.50	56.29 (-3.09)
+ GAIR-RST* [9]	89.36	59.64 (+0.26)
+ GAIR-RST [9]	90.63	50.64 (-8.74)
+ EW-TRADES (τ=1.0)	89.92	59.64 (+0.26)
+ EW-TRADES (τ=2.0)	89.48	**59.75 (+0.37)**

VIII. Discussion

In this paper, we showed that existing weighting schemes for adversarial training yield high-entropy examples with uncertain predictions, thus making them vulnerable to targeted attacks such as AutoAttack. Based on this observation, and the direct association of the entropy to its vulnerability to targeted attacks, we propose to focus on the entropy of each sample in the adversarial training: *Entropy-Weighted Adversarial Training* (EWAT). EWAT is a simple yet effective weighted adversarial training scheme that weighs each instance by its entropy. We show that simple entropy weighting in various datasets, architecture, and experimental settings (e.g. adversarial method, and additional dataset) helps to improve the robustness. EWAT is simple and can be used to weigh the instance-wise adversarial loss of any conventional adversarial training algorithms, such as standard AT and TRADES. Moreover, while the existing instance weighting scheme for adversarial training suffers from vulnerability against logit scaling attack and AutoAttack, entropy is the key to weighing the examples more robustly against them over standard AT and TRADES with even weights across the samples. Further, EWAT also achieves competitively robust accuracy against untargeted PGD attacks to standard AT and TRADES. We show interesting findings of entropy in adversarial learning and show a simple manner to leverage the weighting against several attacks robustly. We believe that we have provided a novel empirical study that shows the vulnerabilities of existing weighting schemes, as well as new insights that link the sample's uncertainty to its vulnerability against targeted attacks, which may lead to follow-up works that exploit our findings.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-00153) and by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST)).

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[2] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[3] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International Conference on Machine Learning*, 2019.

[4] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[5] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, "Improving adversarial robustness requires revisiting misclassified examples," in *International Conference on Learning Representations*, 2019.

[6] G. W. Ding, Y. Sharma, K. Y. C. Lui, and R. Huang, "Mma training: Direct input space margin maximization through adversarial training," in *International Conference on Learning Representations*, 2020.

[7] S. Gowal, S.-A. Rebuffi, O. Wiles, F. Stimberg, D. A. Calian, and T. A. Mann, "Improving robustness using generated data," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4218–4233, 2021.

[8] S.-A. Rebuffi, S. Gowal, D. A. Calian, F. Stimberg, O. Wiles, and T. Mann, "Fixing data augmentation to improve adversarial robustness," *Advances in Neural Information Processing Systems*, 2021.

[9] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. Kankanhalli, "Geometry-aware instance-reweighted adversarial training," *International Conference on Learning Representations*, 2020.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[11] M. Toneva, A. Sordoni, R. T. d. Combes, A. Trischler, Y. Bengio, and G. J. Gordon, "An empirical study of example forgetting during deep neural network learning," *International Conference on Learning Representations*, 2019.

[12] D. Hitaj, G. Pagnotta, I. Masi, and L. V. Mancini, "Evaluating the robustness of geometry-aware instance-reweighted adversarial training," *arXiv preprint arXiv:2103.01914*, 2021.

[13] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2206–2216.

[14] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2574–2582.

[15] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.

[16] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," *International Conference on Learning Representations*, 2017.

[17] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *International Conference on Learning Representations*, 2018.

[18] G. S. Dhillon, K. Azizzadenesheli, Z. C. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, and A. Anandkumar, "Stochastic activation pruning for robust adversarial defense," *International Conference on Learning Representations*, 2018.

[19] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International Conference on Machine Learning*, 2018.

[20] Y. Carmon, A. Raghunathan, L. Schmidt, P. Liang, and J. C. Duchi, "Unlabeled data improves adversarial robustness," *Advances in Neural Information Processing Systems*, 2019.

[21] D. Wu, S.-T. Xia, and Y. Wang, "Adversarial weight perturbation helps robust generalization," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[22] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," in *http://cs231n.stanford.edu/*, 2015.

[23] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," *Citeseer*, 2009.

[24] J. Tack, S. Yu, J. Jeong, M. Kim, S. J. Hwang, and J. Shin, "Consistency regularization for adversarial robustness," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8414–8422.

[25] T. Chen, Z. Zhang, S. Liu, S. Chang, and Z. Wang, "Robust overfitting may be mitigated by properly learned smoothening," in *International Conference on Learning Representations*, 2020.

[26] F. Croce and M. Hein, "Minimally distorted adversarial examples with a fast adaptive boundary attack," in *International Conference*

on Machine Learning. PMLR, 2020, pp. 2196–2205.

[27] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: a query-efficient black-box adversarial attack via random search," in *European Conference on Computer Vision*. Springer, 2020, pp. 484–501.

[28] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems*, 1990, pp. 396–404.

[29] S. Zagoruyko and N. Komodakis, "Wide residual networks," *British Machine Vision Conference*, 2016.

[30] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1321–1330.