

# Towards the Desirable Decision Boundary by Moderate-Margin Adversarial Training

Xiaoyu Liang<sup>a</sup>, Yaguan Qian<sup>a,\*</sup>, Jianchang Huang<sup>a</sup>, Xiang Ling<sup>b</sup>, Bin Wang<sup>c</sup>, Chunming Wu<sup>d</sup>, Wassim Swaileh<sup>e</sup>

<sup>a</sup>School of Science, Zhejiang University of Science and Technology, China

<sup>b</sup>Institute of Software, Chinese Academy of Sciences, China

<sup>c</sup>The Zhejiang Key Laboratory of Multidimensional Perception Technology, Application and Cybersecurity, China

<sup>d</sup>The College of Computer Science, Zhejiang University, China

<sup>e</sup>IRISA Laboratory UMR 6074, INSA De Rennes, CNRS

---

## Abstract

Adversarial training, as one of the most effective defense methods against adversarial attacks, tends to learn an inclusive decision boundary to increase the robustness of deep learning models. However, due to the large and unnecessary increase in the margin along adversarial directions, adversarial training causes heavy cross-over between natural examples and adversarial examples, which is not conducive to balancing the trade-off between robustness and natural accuracy. In this paper, we propose a novel adversarial training scheme to achieve a better trade-off between robustness and natural accuracy. It aims to learn a moderate-inclusive decision boundary, which means that the margins of natural examples under the decision boundary are moderate. We call this scheme Moderate-Margin Adversarial Training (MMAT), which generates finer-grained adversarial examples to mitigate the cross-over problem. We also take advantage of logits from a teacher model that has been well-trained to guide the learning of our model. Finally, MMAT achieves high natural accuracy and robustness under both black-box and white-box attacks. On SVHN, for example, state-of-the-art robustness and natural accuracy are achieved.

*Keywords:* Adversarial training, adversarial attack, trade-off, decision boundary

---

## 1. Introduction

Deep neural networks (DNNs) have achieved great success in many tasks, including image classification; however, recent research indicates that DNNs are extremely vulnerable to adversarial attacks. The so-called adversarial attack is to mislead a classifier by adding some well-designed perturbations to a clean input. Although several countermeasures have been proposed [1, 2, 3], adversarial training has confirmed to be one of the most effective methods. Nevertheless, the higher Robust Accuracy (RA) of adversarial training is often accompanied by more Natural Accuracy (NA) degradation, which prompts us to achieve a good trade-off between RA and NA.

In fact, although adversarially trained models have learned inclusive decision boundaries that keep adversarial counterparts within the original class, the unwarranted increase in the margin along certain adversarial directions is harmful, as shown in Fig. 1(c). The decision boundary's over-inclusiveness causes a heavy cross-over between natural and adversarial examples, which is not conducive to balancing the RA-NA trade-off. In this paper, we expect to alleviate the excessive adversarial directional margin. We begin our analysis by examining the decision boundaries of the DNNs obtained by different training

methods. Next, we investigate the details of previous work [4, 5, 6] and discover that it mostly treats all examples equally and assigns uniform parameters to them. We then conduct experiments to confirm that this is one of the main causes of excessive directional margin. Thus, we propose a novel adversarial training scheme, namely Moderate-Margin Adversarial Training (MMAT).

Here we separate the training examples clearly, identifying both the critical difference between correctly classified and misclassified examples by a model as well as the characteristics of each correctly classified example. For misclassified examples, it does not make sense to study their robustness. Therefore, we consider a boosted loss function to better learn the examples themselves; for correctly classified examples, we design a new attack strategy based on the variability of their margins. It can significantly reduce the excessive adversarial directional margin for the examples near the decision boundary. In addition, we obtain a well-trained model for the same architecture and use it to guide the training process of our defense model, which can regulate the desirable course of the decision boundary and, to some extent, weaken the over-adjustment of the decision boundary along adversarial directions. Ultimately, we obtain a moderate-inclusive decision boundary shown in Fig. 1(b), which improves classification accuracy on natural examples without compromising robustness. To sum up, our main contributions are:

- 1) We consider the RA-NA trade-off to be the issue of how broad the margins of natural examples should be;
- 2) We propose a new adversarial defense method, MMAT,

---

\*Corresponding author

Email addresses: 212009701006@zust.edu.cn (Xiaoyu Liang), qianyaguan@zust.edu.cn (Yaguan Qian), 212109701007@zust.edu.cn (Jianchang Huang), lingxiang@iscas.ac.cn (Xiang Ling), wangbin2@hikvision.com (Bin Wang), wuchunming@zju.edu.cn (Chunming Wu), wassim.swaileh@cyu.fr (Wassim Swaileh)

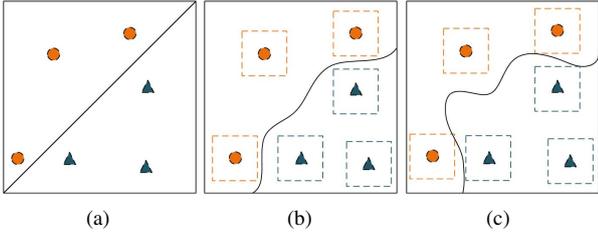


Figure 1: A conceptual illustration of decision boundaries learned via (a) natural training, (b) standard adversarial training, and (c) moderate-margin adversarial training, respectively. The adversarially trained model needs a significantly more complicated decision boundary to memorize adversarial examples of training data.

which generates finer-grained adversarial examples (FAEs) based on projected gradient descent (PGD), reducing the heavy cross-over between natural and adversarial examples;

- 3) Extensive experiments on CIFAR-10 and SVHN show that MMAT has a higher NA while keeping a competitive RA for adversarial examples compared with other adversarial training methods.

## 2. Related work

**Adversarial attacks.** Szegedy et al. [7] first observed that DNNs are highly vulnerable to adversarial examples. Goodfellow et al. [8] proposed Fast Gradient Sign Method (FGSM), which makes use of the gradient of the model to generate adversarial examples. Madry et al. [4] proposed Projected Gradient Descent (PGD) method as a universal "first-order adversary", i.e., the most active attack utilizing the local first-order information about the network. Moreover, boundary-based methods like DeepFool [9] and optimization-based methods like CW [10] were also developed, making adversarial defense more challenging. Recently, the ensemble of diverse attack methods - auto-attack [11] by Croce et al., consisting of APGD-CE [11], APGD-DLR [11], FAB [12], and Square Attack [13], became a popular benchmark for testing the robustness of models. These are white-box attacks since the adversary has full knowledge of DNNs. Numerous works have also been published that investigate the transferability of adversarial examples for black-box attacks. Liu et al. [14] were the first to study the transferability of targeted adversarial examples. Dong et al. [15] showed that iterative attack methods incorporating the momentum term achieved better transferability. Xie et al. [16] boosted the transferability of adversarial examples by creating diverse input patterns with random resizing and random padding.

**Adversarial training.** Many of the early adversarial defenses [17, 18, 19] are broken by stronger attacks. Among various defense strategies, adversarial training is currently the best defense method against adversarial attacks. The initial idea of adversarial training is first brought to light by [7], where neural networks are trained on a mixture of adversarial examples and clean examples. Goodfellow et al. [8] went further and proposed FGSM to produce adversarial examples during training. Mathematically, adversarial training is formulated as a minimax optimization problem. Madry et al. [4] employed a multi-step

gradient-based attack known as PGD attack for solving the inner problem. Wang et al. [20] investigated the influence of misclassified examples and proposed MART to further improve robustness. However, it is often observed that these methods compromise the accuracy of unperturbed examples. Such observations had led prior work to speculate on a trade-off between the two fundamental notions of natural accuracy and robustness. Zhang et al. [5] regularized the output from natural images and adversarial examples with the KL divergence function and proposed TRADES, which can trade natural accuracy off against adversarial robustness. Zhang et al. [21] proposed searching for the least adversarial data for AT, which could be called FAT. Cui et al. [22] proposed LBGAT to adapt the logits of one model trained on clean data to guide adversarial training. Rade et al. [6] incorporated additional wrongly labelled examples during training and presented HAT to provide a notable improvement in accuracy without compromising robustness.

## 3. Problem statement

### 3.1. Preliminaries

Suppose  $\mathbb{D} = \{(x_i, y_i)\}_{i=1}^N$  is a dataset where  $x_i \in \mathbb{R}^d$  is a natural example, and  $y_i \in 1, 2, \dots, K$  is its corresponding label. A DNN  $f_\theta$  is a function with parameter  $\theta$  to predict an input example  $x_i$ :  $f_\theta(x_i) = \arg \max_{k=1, \dots, K} z_k(x_i)$ ,  $\mathbf{p}_k(x_i) = e^{z_k(x_i)} / \sum_{k'=1}^K e^{z_{k'}(x_i)}$ , where  $z_k(x_i)$  is the logits output of the network with respect to class  $k$ , and  $\mathbf{p}_k(x_i)$  is the probability (softmax on logits) belonging to class  $k$ . An adversarial example  $x_i'$  is crafted by adding a small perturbation  $\delta_i$  to  $x_i$ , which misleads DNNs as  $f_\theta(x_i') \neq y_i$ , while  $f_\theta(x_i') = y_i$ . Small perturbation means that  $x_i$  satisfy  $\{x_i' \in \mathbb{B}(x_i, \epsilon) : \|x_i' - x_i\| \leq \epsilon\}$ , where  $\|\cdot\|$  is a generic distance norm that can be  $\|\cdot\|_\infty$  and  $\|\cdot\|_2$ .

**Decision boundary.** Given a classifier  $f_\theta$ , a dataset  $\mathbb{D} = (x_i, y_i)_{i=1}^N$ , we can define the decision boundary  $\mathbb{D}' \subseteq \mathbb{R}^d$  of  $f_\theta$  as the set:

$$\mathbb{D}' := \{x_i \in \mathbb{R}^d | \exists l \neq j, \mathbf{p}_l(x_i) = \mathbf{p}_j(x_i) = \max_k \mathbf{p}_k(x_i)\} \quad (1)$$

**Margin.** Given a unit vector  $\mathbf{v} \in \mathbb{S}^{d-1}$ , the margin  $m$  at  $x_i$  along the direction  $\mathbf{v}$  is given by:

$$m(x_i, \mathbf{v}) = \arg \min_a |a| \quad s.t. x_i + a\mathbf{v} \in \mathbb{D}' \quad (2)$$

**Adversarial directions.** Given a classifier  $f_\theta$ , a dataset  $\mathbb{D} = (x_i, y_i)_{i=1}^N$ , we define the set of adversarial directions as  $V = \{\mathbf{v}_i / \|\mathbf{v}_i\|_2\}_{i=1}^N$  where  $\mathbf{v}_i$  is obtained by solving:

$$\mathbf{v}_i = \max_{\delta: \|\delta\|_p \leq \epsilon} L(f_\theta(x_i + \delta), y_i) \quad (3)$$

where  $L(\cdot)$  is an arbitrary loss function. This optimization problem is usually solved via PGD.

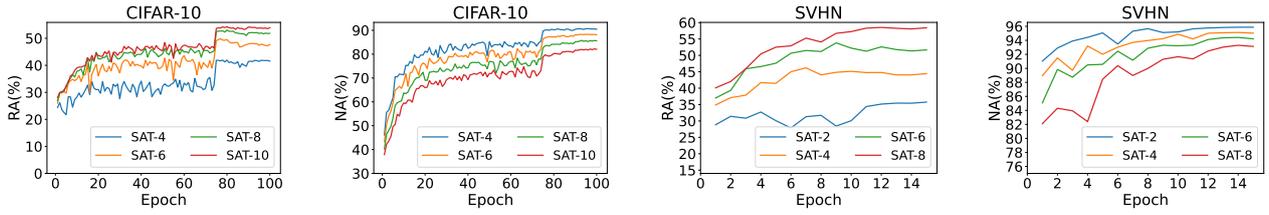


Figure 2: The RA and NA of standard adversarially trained models. A trained model with  $\epsilon = k/255$  is denoted by SAT- $k$ . Here we train ResNet-18 on CIFAR-10 and PreActResNet-18 on SVHN. During training, we generate adversarial examples by using 10 steps of size  $\epsilon/4$ , where  $\epsilon$  is the perturbation budget, and different  $\epsilon$  represents different attack strengths.

### 3.2. Motivation

A naturally trained model can attain perfect NA, but its average margin has a clear downward trend over the training process. Eventually, a large portion of the training examples as well as test examples lie close to the decision boundary after training [23]. The relatively low average margins lead to the vulnerability of DNNs to adversarial examples. In the adversarial setting, even though a slight downward trend is inevitable, it is not as severe as that in the natural setting, and this downward trend starts in later epochs compared to natural training. The adversarially trained models with larger average margins are robust while triggering a superfluous increase in the margin along adversarial directions as compared to the nominal increase required to attain robustness [6]. The excessive directional margin might be the reason for a reduction in natural accuracy. This argument is in line with the previous works by [24], [25].

We then obtain several robust models using the standard adversarial training (SAT) method [4], and further investigate the differences in RA and NA among them. As shown in Fig. 2, the RA grows as  $\epsilon$  increases, but the gain of this growth becomes gradually smaller. Meanwhile, the process is accompanied by a continued drop in NA. We realize that one reason for this phenomenon is that all examples share a uniform  $\epsilon$ . With the increase of  $\epsilon$ , more examples have excessive directional margin, especially those near the decision boundary, which significantly cross over the decision boundary and are located in the area of natural examples, causing a serious cross-over mixture problem [21]. Therefore, we prefer designing finer-grained parameters  $\epsilon_i$  for different examples to alleviate the problem and exploit the potential of larger perturbation budgets.

Moreover, we can stimulate a slight push to the decision boundary during adversarial training to reduce the potential excess directional margin. This can be achieved by fine-tuning the robust model using the decision boundary learned by a natural model. To implement the idea, we consider a *weak* robust model as our teacher model, such as SAT-6 in Fig. 2, which inherits the decision boundary of the natural model better in the process of improving robustness.

### 4. Moderate-Margin Adversarial Training

To learn a moderate-inclusive decision boundary, we propose a novel adversarial training method named MMAT. This method achieves a good trade-off by designing two loss terms  $L_1$  and  $L_2$  respectively for adversarial and natural examples, and design a new attack strategy to generate our finer-grained adversarial examples.

---

#### Algorithm 1 The configuration of $\epsilon_i$

---

**Input:** A dataset  $\mathbb{D}$ , a strategy network  $f_s$   
**Output:**  $\epsilon_i$

- 1: Train a strategy network  $f_s$  on  $\mathbb{D} = (\mathbf{x}_i, y_i)_{i=1}^N$
- 2:  $\mathbb{D}_A \leftarrow \emptyset, \mathbb{D}_B \leftarrow \emptyset, \mathbb{D}_C \leftarrow \emptyset,$
- 3: **while**  $f_s(\mathbf{x}_i) \neq y_i$  **do**
- 4:    $\epsilon_i \leftarrow 0$
- 5: **end while**
- 6: **while**  $f_s(\mathbf{x}_i) = y_i$  **do**
- 7:   calculating the margin of all examples:  
 $M = \{m(\mathbf{x}_i)\}_{i=1}^N$
- 8:   **if**  $m(\mathbf{x}_i) \leq M_{P_{40}}$  **then**
- 9:      $\mathbb{D}_A \leftarrow \mathbb{D}_A \cup \{\mathbf{x}_i\}$
- 10:      $\epsilon_i \leftarrow \epsilon_A = \max_m m(\mathbf{x}_i), \mathbf{x}_i \in \mathbb{D}_A$
- 11:   **else if**  $m(\mathbf{x}_i) \leq M_{P_{70}}$  **then**
- 12:      $\mathbb{D}_B \leftarrow \mathbb{D}_B \cup \{\mathbf{x}_i\}$
- 13:      $\epsilon_i \leftarrow \epsilon_B = \frac{1}{|\mathbb{D}_B|} \sum_{\mathbf{x}_i \in \mathbb{D}_B} m(\mathbf{x}_i)$
- 14:   **else if**  $m(\mathbf{x}_i) \leq M_{P_{100}}$  **then**
- 15:      $\mathbb{D}_C \leftarrow \mathbb{D}_C \cup \{\mathbf{x}_i\}$
- 16:      $\epsilon_i \leftarrow \epsilon_C = \min_m m(\mathbf{x}_i), \mathbf{x}_i \in \mathbb{D}_C$
- 17:   **end if**
- 18: **end while**

---

#### 4.1. Loss function

The first term of loss function,  $L_1$ , is for the adversarial examples. Obviously, adversarial examples' classification is more arduous than natural examples', the decision boundary is no longer flat, and its local curvature profoundly affects the final performance of the defense model, as shown in Fig. 1, so the boosted cross entropy loss (BCE) is chosen here instead of the commonly used cross entropy loss. Thus  $L_1$  is defined as follows:

$$L_1 = -\log(\mathbf{p}_{y_i}(\mathbf{x}'_i)) - \log(1 - \max_{k \neq y_i} \mathbf{p}_k(\mathbf{x}'_i)) \quad (4)$$

where  $\mathbf{p}_k(\mathbf{x}'_i)$  is the probability belonging to class  $k$ , the first term  $-\log(\mathbf{p}_{y_i}(\mathbf{x}'_i))$  is the commonly used CE loss, denoted CE  $(\mathbf{p}(\mathbf{x}'_i), y_i)$ , and the second term  $-\log(1 - \max_{k \neq y_i} \mathbf{p}_k(\mathbf{x}'_i))$  is a margin term used to improve the classifier's decision margin, allowing it to be a robust classifier with high confidence.

The second term,  $L_2$ , is for natural examples. The teacher model  $f_t$  is a well-trained *weak* robust model (like SAT with  $\epsilon = 6/255$ ). The decision boundary of  $f_\theta$  will be fine-tuned in

---

**Algorithm 2** Moderate-Margin Adversarial Training

---

**Input:** Training dataset  $\mathbb{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ **Parameter:** Batch-size  $m$ , learning rate  $\eta$ , scaling parameter  $\lambda$ , attack step size  $\alpha$  and number of attack iterations  $T$ 

- 1: Train a teacher network  $f_t$  on  $\mathbb{D}$
  - 2: **repeat**
  - 3: Read a mini-batch  $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$  from  $\mathbb{D}$
  - 4: **for**  $j = 1, 2, \dots, m$  **do**
  - 5:  $\mathbf{x}_{ij}' \leftarrow \mathbf{x}_{ij} + 0.001 \times \mathcal{N}(\mathbf{0}, \mathbf{I})$ , where  $\mathcal{N}(\mathbf{0}, \mathbf{I})$  is the Gaussian distribution with zero mean and identity variance
  - 6: Grade  $\mathbf{x}_{ij}$  based on its margin and set  $\epsilon_{ij}$  for it
  - 7: **for**  $t = 1, 2, \dots, T$  **do**
  - 8:  $\mathbf{x}_{ij}' \leftarrow \prod_{\mathcal{B}(\mathbf{x}_{ij}, \epsilon_{ij})} (\mathbf{x}_{ij}' + \alpha \cdot \text{sign} \nabla_{\mathbf{x}_{ij}'} CE)$
  - 9: **end for**
  - 10: **end for**
  - 11:  $\theta \leftarrow \theta - \frac{\eta}{m} \sum_{j=1}^m \nabla_{\theta} (L_1 + L_2 / \lambda)$
  - 12: **until** training completed
- 

real time during training, and the fine-tuning will prevent the undesirable excessive rise in the margin to some extent, thus making it possible to achieve a better trade-off. We make use of logits from the teacher model to help reshape the decision boundary of our robust model, which reduces damage to the NA. Here we use MSE loss as  $L_2$ :

$$L_2 = \|\mathbf{z}^t(\mathbf{x}_i) - \mathbf{z}^\theta(\mathbf{x}_i)\|_2^2 \quad (5)$$

where  $\mathbf{z}^\theta(\mathbf{x}_i)$ ,  $\mathbf{z}^t(\mathbf{x}_i)$  is logit output of input  $\mathbf{x}_i$  of  $f_\theta$  and  $f_t$  respectively. When the student  $f_\theta$  is trained with MSE loss, its representations attempt to follow the shape of the teacher's representations. Besides, MSE is efficient in transferring the teacher's information to a student [26].

#### 4.2. Finer-grained adversarial examples

In this subsection, we elaborate on the generation of our finer-grained adversarial examples. Intuitively, we can learn *worse* examples with larger perturbation budgets than  $\epsilon_c$  to defend against attacks with that  $\epsilon_c$ . However, previous adversarial training mostly treats all examples equally to generate adversarial examples, which are also called rough adversarial examples here. It results in: 1) increasing the perturbation budget is accompanied by a significant decrease in natural accuracy; 2) large perturbation budgets are extremely unreasonable for misclassified examples or even correctly classified examples near the decision boundary; and 3) excessive perturbation budgets are not necessarily beneficial for robustness improvement.

**Idea of attack strategy.** Instead of using a uniform  $\epsilon$  for all examples, our new attack strategy aims to design finer-grained attack parameters  $\epsilon_i$  for them. As we know, the network parameter  $\theta$  is updated each epoch during the training process to obtain the current model  $f_{\theta_{new}}$ . Natural training examples can be divided into two subsets with respect to  $f_{\theta_{new}}$ , with one subset of correctly classified examples  $\mathbb{D}^+ = \{\mathbf{x}_i : i \in [N], f_{\theta_{new}}(\mathbf{x}_i) = y_i\}$ , and one subset of misclassified examples  $\mathbb{D}^- = \{\mathbf{x}_i : i \in$

$[N], f_{\theta_{new}}(\mathbf{x}_i) \neq y_i\}$ . As shown in Fig. 3, there is a fundamental difference between correctly classified and misclassified examples of a model, with the latter being inherently difficult to learn and inappropriate for further adversarial attack, i.e.,  $\epsilon_i = 0$ . Actually, the margin distribution of correctly classified examples in Fig. 3 also shows significant differences between these examples, so we calculate margins for the examples correctly classified by the model  $f_{\theta_{new}}$  for each epoch, and thus determine their attack parameters. Apparently, our design is more reasonable, which takes into account the multiple characteristics of the examples. In addition, for each epoch, we select a new subset  $\mathbb{D}^+$  and calculate the corresponding margin based on the current network  $f_{\theta_{new}}$ . Because both the subset and the margin are changing, this new attack scheme belongs to a dynamic adjustment and demonstrates great flexibility.

**Measure of example's margin.** A key point of our attack strategy is the calculation of margin. Since calculating the exact margin in the input space is intractable, we have to search for an approximate solution. To be more precise, we search for a small perturbation  $\delta$  such that  $f_\theta(\mathbf{x}_i + \delta) \neq f_\theta(\mathbf{x}_i)$ . Ideally, we then have  $m(\mathbf{x}_i, \mathbf{v}) \approx \|\delta\|_\infty$ , although  $\mathbf{x}_i + \delta$  is not necessarily an element of  $\mathbb{D}'$ . We will obtain the margin estimate with the help of DeepFool [9], an iterative adversarial attack which stops as soon as a perturbation  $\delta$  has been found with  $f_\theta(\mathbf{x}_i + \delta) \neq f_\theta(\mathbf{x}_i)$ . In every iteration step of DeepFool, it linearizes the decision boundary around an example to gradually push the example over the closest boundary for minimal perturbation. This perturbation  $\delta_i$  can be written as:

$$\delta_i := \frac{|\mathbf{p}_f(\mathbf{x}_i) - \mathbf{p}_{y_i}(\mathbf{x}_i)|}{\|\nabla \mathbf{p}_f(\mathbf{x}_i) - \nabla \mathbf{p}_{y_i}(\mathbf{x}_i)\|_1} * (\nabla \mathbf{p}_f(\mathbf{x}_i) - \nabla \mathbf{p}_{y_i}(\mathbf{x}_i)) \quad (6)$$

with index:

$$\hat{l} = \hat{l}(\mathbf{x}_i) := \arg \min_{k \neq y_i} \frac{|\mathbf{p}_f(\mathbf{x}_i) - \mathbf{p}_k(\mathbf{x}_i)|}{\|\nabla \mathbf{p}_f(\mathbf{x}_i) - \nabla \mathbf{p}_k(\mathbf{x}_i)\|_1} \quad (7)$$

Let  $l$  denote the stopping index of this iterative scheme for  $\mathbf{x}_i$ , i.e.,  $f_\theta(\mathbf{x}_i^l) \neq f_\theta(\mathbf{x}_i)$ . The desired adversarial perturbation is then defined as  $\delta = \sum_{i=0}^{l-1} \delta_i$ . Note that we only use successful adversarial perturbations for the margin approximation. Thus, if the DeepFool attack is not able to find a small adversarial perturbation for a given image, we regard its semantic features as more obvious, and the vulnerability can be ignored without considering its margin. We have  $m = \|\delta\|_\infty$  now.

**Implementation of attack strategy.** We will use a strategy model to develop our attack strategy, an example-dependent strategy to set the parameter  $\epsilon_i$  when performing an adversarial attack, as shown in the Algorithm 1. Since most previous adversarial training work has used SAT-8 ( $\epsilon = 8/255$ ) as a baseline and followed its attack setup, here we use a well-trained SAT-8 as a strategy model and consider the attack scheme to be migratory and applicable to other defense methods. The margin distribution of correctly classified examples of this strategy model is shown in Fig. 3(c). In fact, it is feasible for us to rank the margin values of examples in ascending order and to rate them

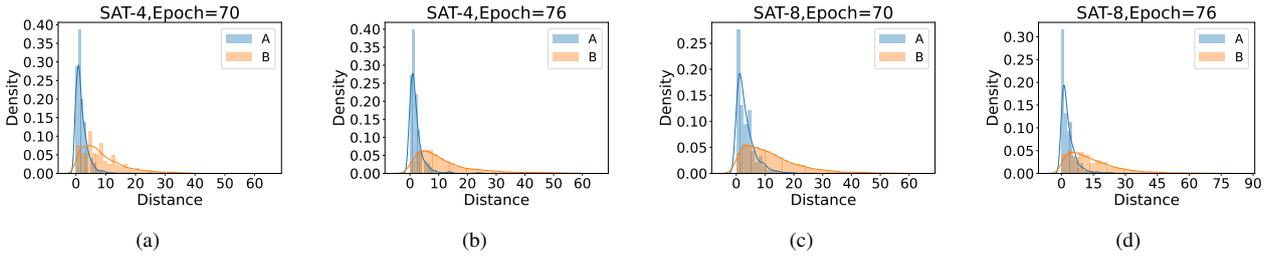


Figure 3: The distribution of examples’ distance from different decision boundaries. **A** represents correctly classified examples by the model, while **B** represents misclassified examples. Here we pick the decision boundaries learned by SAT-4 and SAT-8 at different epochs.

into three grades according to a certain proportion. The examples at each grade are recorded as  $\mathbb{D}_A, \mathbb{D}_B, \mathbb{D}_C$ . The example set closest to the decision boundary,  $\mathbb{D}_A$ , is the key to determine the degree of destruction of natural accuracy and is our key concern, accounting for 40%;  $\mathbb{D}_B$ , the example set second closest to the decision boundary, the average margin is slightly greater than  $8/255$ ;  $\mathbb{D}_C$ , the example far from the decision boundary, basically completely avoid the cross-over between the adversarial examples and the original natural examples, so  $\mathbb{D}_B$  and  $\mathbb{D}_C$  can well improve the robustness of the model, and the two account for 30% respectively. The three grades are practically meaningful and sufficiently representative of the properties of correctly classified examples.  $\mathbb{D}_A$ , in particular, aims to ensure natural accuracy at the expense of robustness;  $\mathbb{D}_B$  has the greatest trade-off advantage; and  $\mathbb{D}_C$  can further exploit the potential robustness without requiring an excessive range of budgets. As a result, we set  $\epsilon_A, \epsilon_B$ , and  $\epsilon_c$  to the values shown in the Algorithm 1. Finally, the FAEs are solved by:

$$\tilde{x}'_i = \arg \max_{x'_i \in \mathbb{B}(x_i, \epsilon_i)} CE(p(x'_i), y_i) \quad (8)$$

Based on the two proposed loss terms and the FAEs, we can state our final objective function for MMAT:

$$\min_{\theta} \frac{1}{N} \left( \sum_{i \in \mathbb{D}} L_1 + \sum_{i \in \mathbb{D}} L_2 / \lambda \right) \quad (9)$$

where  $\lambda$  is a tunable scaling parameter that balances the two parts of the hybrid loss.

The Algorithm 2 depicts our Moderate-Margin Adversarial Training. Our attack strategy is based on the margins of examples. The teacher model is used to guide the real-time decision boundary adjustment of our model. As a result, MMAT can also be framed as performing student-teacher learning, particularly self-distillation [27] to mimic certain properties of the model itself.

## 5. Experiment

### 5.1. Experimental setup

**Training methods.** We consider neural networks trained via Natural Training (NT), Standard Adversarial Training (SAT) [28], TRADES [5], MART [20], LBGAT [6], and HAT [22].

The trade-off parameter,  $\beta$ , is present in TRADES, MART, LBGAT, and HAT, and a higher  $\beta$  indicates that the robust accuracy is given more weight.

**White-box attacks.** We evaluate robustness with four types of attacks, *i.e.*, FGSM, PGD<sup>20</sup>, CW<sub>∞</sub>, and Auto-Attack (AA), on two datasets. All attacks are constrained by the same perturbation budget, *i.e.*,  $\epsilon = 8/255$ . Note that the CW<sub>∞</sub> attack denotes using CW-loss within the PGD framework here. The goal of Auto-Attack is to reliably evaluate model robustness with an ensemble of diverse strong attack methods. We use the open-source code from [29] to test our models. Besides, for PGD, the training attack is PGD<sup>10</sup> with random start and step size  $\alpha = \epsilon/4$ , while the test attack is PGD<sup>20</sup> with random start, perturbation budget  $\epsilon = 8/255$  and step size  $\alpha = \epsilon/10$ .

**Black-box attacks.** We train ResNet-18 on CIFAR-10 and PreActResNet-18 on SVHN with different adversarial training methods. Then we use the well-trained model as the source model or the target model. Adversarial examples are generated from the source model to attack the target model. On both datasets, the PGD<sup>20</sup> (black-box) is applied to attack various defense models. And the setup follows that specified in the white-box attack.

**Training details.** To evaluate the performance of different models, we conduct extensive experiments on ResNet-18 [30] using CIFAR-10 [31] and PreActResNet-18 using SVHN [32]. For CIFAR-10, following [20], the batch size is 128 and the number of training epochs is 100. All models are trained using the SGD optimizer with Nesterov momentum 0.9 [33], weight

Table 1: The performance of ResNet-18 is trained using MMAT with different attack settings on CIFAR-10. We pick the checkpoint which has the best robust accuracy on the test set. The difference (Diff.) between best and final accuracy indicates degradation in performance during training.

		RA			NA		
		Best	Final	Diff.	Best	Final	Diff.
$Z_1$	2	55.58	55.14	0.44	85.32	85.84	-0.52
	3	53.77	53.31	0.46	86.36	86.54	-0.18
	4	52.13	51.49	0.64	87.71	87.80	-0.09
$\epsilon_A$	3/255	55.16	54.57	0.59	85.15	85.54	-0.39
	5/255	55.58	55.14	0.44	85.32	85.84	-0.52
	7/255	55.69	55.19	0.50	84.77	85.25	-0.48

Table 2: Comparisons with other defense methods under white-box attacks. We pick the checkpoint which has the best robust accuracy on the test set. Note that we still follow the basic experimental setup described in Section 5.1. Hyper-parameters and training details of other defense methods are configured as per their original papers:  $\epsilon = 6$  for TRADES and MART; ResNet18 is adopted as  $M_{\text{natural}}$  for LBGAT; a well naturally trained  $f_{\text{std}}$  for HAT.

Adversary	Method	CIFAR-10		SVHN	
		RA(%)	NA(%)	RA(%)	NA(%)
FGSM	SAT	57.34	83.50	71.17	92.47
	TRADES	56.72	81.57	73.32	91.25
	MART	59.68	81.39	72.93	91.43
	LBGAT	56.75	85.28	69.55	92.75
	HAT	61.10	85.95	76.66	92.88
	MMAT	<b>59.78</b>	<b>85.32</b>	<b>72.75</b>	<b>93.45</b>
PGD <sup>20</sup>	SAT	52.69	83.50	58.52	92.47
	TRADES	52.90	81.57	60.83	91.25
	MART	55.33	81.39	60.70	91.43
	LBGAT	52.10	85.28	59.84	92.75
	HAT	52.49	85.95	61.65	92.88
	MMAT	<b>55.58</b>	<b>85.32</b>	<b>62.20</b>	<b>93.45</b>
CW <sub>∞</sub>	SAT	49.23	83.50	54.64	92.47
	TRADES	49.41	81.57	56.16	91.25
	MART	49.91	81.39	53.40	91.43
	LBGAT	48.81	85.28	54.92	92.75
	HAT	48.73	85.95	55.27	92.88
	MMAT	<b>50.33</b>	<b>85.32</b>	<b>55.35</b>	<b>93.45</b>
AA	SAT	47.67	83.50	50.12	92.47
	TRADES	48.33	81.57	53.44	91.25
	MART	48.03	81.39	49.83	91.43
	LBGAT	47.29	85.28	52.02	92.75
	HAT	47.28	85.95	51.06	92.88
	MMAT	<b>48.49</b>	<b>85.32</b>	<b>52.13</b>	<b>93.45</b>

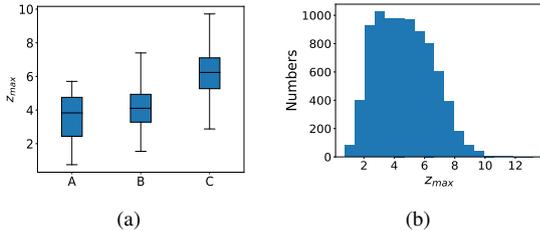


Figure 4: The analysis of  $z_{\max}$  of correctly classified examples by the well-trained strategy network. (a): The horizontal axis represents the grades based on the margins of these examples. Examples with higher grades correspond to larger  $z_{\max}$ . (b): The distribution of  $z_{\max}$ . Here we sample 10000 examples from the training examples.

decay 0.0035 and an initial learning rate of 0.01, which is divided by 10 at the 75-th and 90-th epochs. Data augmentation is performed. When performing data augmentation, we randomly crop the image to  $32 \times 32$  with 4 pixels of padding, then perform random horizontal flips. For SVHN, following [6], the batch size is still 128. We use the SGD optimizer with Nesterov momentum 0.9 and weight decay 0.0005. We further employ cyclic learning rates [34] with cosine annealing and a maximum learning rate of 0.05 for SVHN. Here we only run 15 epochs.

## 5.2. Understanding the proposed MMAT

In this section, we investigate MMAT from three different perspectives: 1) the perturbation budgets of FAEs; 2) the guiding role of teacher models; and 3) replacing components of the BCE term.

**Perturbation budget of FAEs.** Since adversarial training is time-consuming, taking 3-30 times longer to form a robust network than a non-robust equivalent [35], [36], we prefer to use

a metric without extra computation to determine the grade of examples instead of the margin. As shown in Fig. 4(a),  $z_{\max}$  (the largest element of a network’s logit output after forward-propagation) of examples meet this requirement. In Fig. 4(b), the distribution of  $z_{\max}$  implies the hierarchical nature of these examples; we simply use  $Z_1 = 2$ ,  $Z_2 = 6$  to divide these examples into three grades, i.e.,  $\mathbb{D}_A = \{\mathbf{x}_i | z_{\max}(\mathbf{x}_i) \leq 2\}$ ,  $\mathbb{D}_B = \{\mathbf{x}_i | z_{\max}(\mathbf{x}_i) \leq 6\}$ ,  $\mathbb{D}_C = \{\mathbf{x}_i | z_{\max}(\mathbf{x}_i) > 6\}$ . And, according to Algorithm 1,  $\epsilon_A = 5/255$ , we have  $\epsilon_B = 10/255$ ,  $\epsilon_A = 15/255$ , which means that  $\epsilon_i$  of each example  $\mathbf{x}_i$  can be set. Furthermore, we investigate the effects of  $\mathbb{D}_A$  under various  $Z_1$  and  $\epsilon_A$ , with the results summarized in Table 1. Apparently, larger  $Z_1$  or smaller  $\epsilon_A$  appears to favor natural accuracy over robustness. The results demonstrate the powerful regulation capability of MMAT in terms of the RA-NA trade-off. We also discovered that MMAT can enable larger perturbation budget  $\epsilon_i$  by alleviating the cross-over mixture problem [21].

**Role of teacher models.** We first investigate the parameter  $a$  ( $1/\lambda$ ) in the MMAT objective function defined in (9) which controls the strength of the regularization. A larger  $a$  indicates that the MSE term plays a more significant role, and the teacher model has a more obvious influence on the course of the decision boundary. As illustrated in Fig. 5(a) and (b), as  $a$  increases, the RA in the training process tends to decrease, especially when  $a = 1$ , while NA tends to increase. For  $a = 1/4$  and  $a = 1/8$ , the model has higher RA and NA, which confirms that the fine-tuning effect of the teacher model can maintain high NA without compromising the robustness of the student model. Next, fixing  $a = 1/4$ , we further explore the effect of different teacher models. As the teacher model becomes more robust with lower NA, the student model also shows this trend. Thus, we confirm that fine-tuning of a suitable teacher model facilitates the desirable course of decision boundary to achieve a good trade-off.

**Replacing components of  $L_1$  term.** Recall that loss term  $L_1$  is defined by BCE. As we show in Fig. 6(a) and (b), learning with CE instead of our BCE suffers from insufficient learning with lower robustness throughout the entire training process. Similarly, when replacing CE with KL in the inner maximization of the adversarial min-max framework, the robustness has been weak throughout the training process. Fig. 6(c) and (d) show the contribution of our BCE term with FAEs. Specifically, we generate adversarial examples with three strategies. Both the RA and NA of AE-1 are lower than our original MMAT, suggesting that learning misclassified examples is more necessary than learning their adversarial counterparts. Moreover, AE-2 leads to a robustness degradation, indicating that our FAE associated with characteristics of natural examples better exploits the model’s robustness potential.

## 5.3. Comparisons with other defense methods

In this section, we assess MMAT and other defense methods mentioned in Section 5.1 against white-box and black-box attacks, as well as perform a deep analysis to confirm the effectiveness of our method.

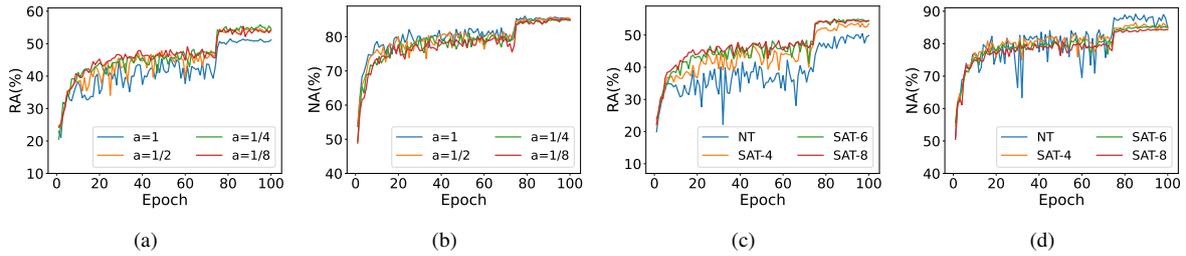


Figure 5: The guiding role of teacher models. (a)-(b): Our models are trained with different scaling parameters  $a$  ( $1/l$ ) under the supervision of SAT-6. (c)-(d): Our models are trained under the supervision of four teacher models, each representing a different level of NA and obtained through various training methods.

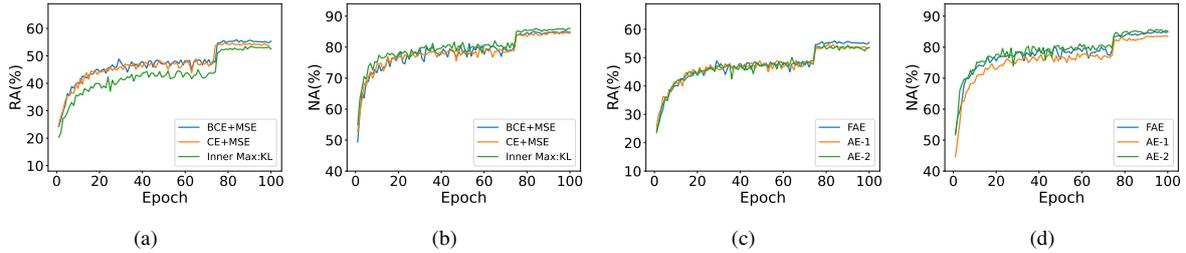


Figure 6: We replace components of the BCE term and compare the performance of the models trained with MMAT variants. In each plot, the solid blue line represents the original MMAT method. (a)-(b): We replace BCE loss with CE loss. In the adversarial min-max framework, we use KL instead of CE for inner maximization. (c)-(d): We substitute other attack strategies for our FAEs. FAE represents our original MMAT. AE-1 is an attack strategy that no longer distinguishes whether an example is correctly classified or not, but directly ranks all examples. AE-2 is an attack that adheres to the uniform perturbation budget of SAT and only attacks correctly classified examples.

Table 3: Comparisons with other defense models under black-box PGD<sup>20</sup> attack on CIFAR-10 and SVHN. Source and target models are the same as those used for white-box attacks. Here, MMAT is used as a source model to attack other defense models, and it is also used as a target model to be attacked by other defense models.

Dataset	Source Model	Target model	RA(%)	NA(%)
CIFAR-10	MMAT	SAT	63.43	83.50
	MMAT	TRADES	63.26	81.57
	MMAT	MART	63.89	81.39
	MMAT	LBGAT	63.83	85.28
	MMAT	HAT	64.64	85.95
	SAT	MMAT	63.95	85.32
	TRADES	MMAT	63.90	85.32
	MART	MMAT	65.64	85.32
	LBGAT	MMAT	64.17	85.32
	HAT	MMAT	64.11	85.32
SVHN	MMAT	SAT	64.60	92.47
	MMAT	TRADES	66.39	91.25
	MMAT	MART	65.58	91.43
	MMAT	LBGAT	67.20	92.75
	MMAT	HAT	65.66	92.88
	SAT	MMAT	65.49	93.45
	TRADES	MMAT	66.88	93.45
	MART	MMAT	68.31	93.45
	LBGAT	MMAT	65.47	93.45
	HAT	MMAT	73.41	93.45

**Evaluation results under white-box attacks.** Table 2 shows the evaluation results of defense models under white-box attacks on CIFAR-10 and SVHN. Our model achieves high RA and NA across multiple datasets and attack methods. On CIFAR-10, TRADES and MART have relatively low NA but perform well in terms of robustness. LBGAT and HAT have higher NA, but their robustness is even weaker than SAT. It is worth noting that our model’s robustness exceeds that of MART, while our NA remains very high, second only to HAT. Because it is easier to classify than CIFAR-10, the NA is generally high, and the difference between them on SVHN is small.

Despite this, our method has the best NA. In terms of robustness, TRADES achieves the best results under  $CW_\infty$  and AA attacks, LBGAT comes in second only to TRADES, MART loses its original advantage, while HAT and MMAT perform admirably under all four attacks.

**Evaluation results under black-box attacks.** The evaluation results of defense models under black-box attacks on CIFAR-10 and SVHN are reported in Table 3. Compared with the white-box (PGD<sup>20</sup>) results, all defense methods achieve much better robustness against black-box attacks. Compared with other models, ours is capable of generating stronger adversarial examples. Moreover, our model exhibits high robustness and natural accuracy on both datasets, demonstrating our method’s excellent performance. For a fair comparison, we keep the training and evaluation settings the same as in MART [20] and HAT [6], and we ensure that the important training details involved in different methods are consistent with the original paper. Thus, we assert that MMAT is the closest to the optimal trade-off among these classical defense methods.

**Discussion of MMAT’s effectiveness.** In Fig. 7, we collect the logit features of natural examples and find that our learned features have a larger distance between classes while being more clustered within the same class. The more distinguishable feature embedding justifies our improvement of both RA and NA. In particular, the distribution of MMAT is similar to that of SAT and MART, whose robustness is great, and MMAT can well distinguish ten classes’ features of the dataset, like LBGAT, which has high natural accuracy. Then we study the distributions of examples’ margins obtained by multiple models. Fig. 8 exhibits the distributions. We observe that the margins obtained by our models are consistently in the medium range of all models at

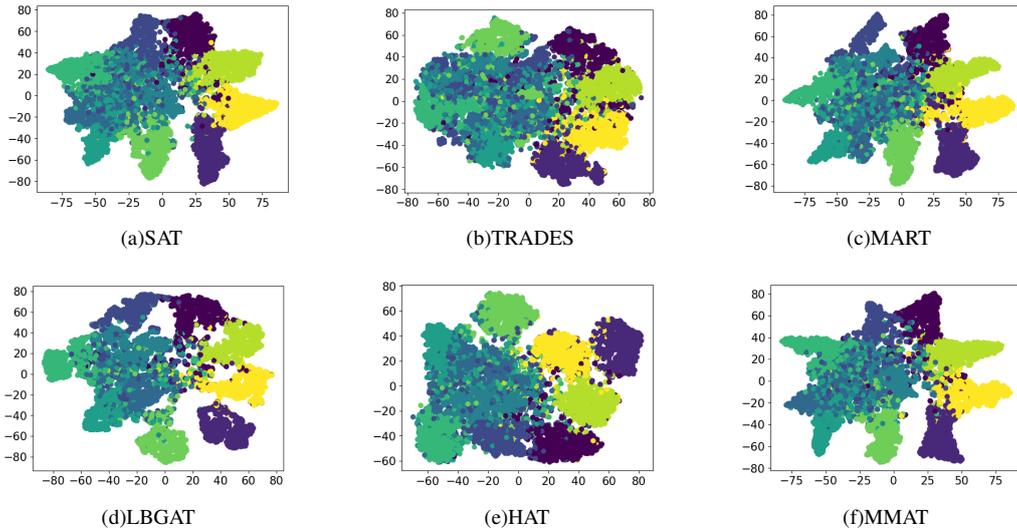


Figure 7: t-SNE results of different methods trained on CIFAR-10. Different colors represent different classes. (a)-(f) represent different training methods, respectively.

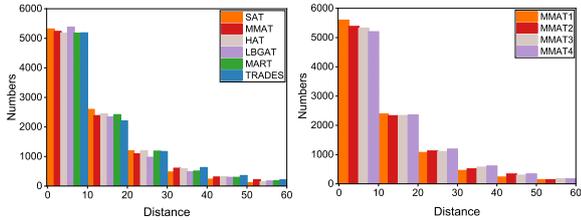


Figure 8: The distribution of examples' margins (the values scaled to 255 times) to the decision boundaries of multiple models. We still use the models trained in Table 2, and calculate the margin of each model separately. Here we only randomly select 10,000 training examples.

different margin levels, in line with our quest for moderate margins. In addition, the example size with small margins of SAT significantly exceed those of MMAT, while the example size with larger margins gradually less than us, which explains why our method can improve both robust accuracy and natural accuracy over SAT. Another interesting finding is that TRADES consistently has the lowest example size when the margin is small, and the highest example size as the margin gradually increases, which makes us understand better why it can stand out under the strong attacks of  $CW_\infty$  and AA.

## 6. Conclusion

Based on previous work and our experimental findings, we discover that adversarially trained models cause an excessive increase in the margin along adversarial directions. This partly contributes to the much-debated RA-NA trade-off. The commonly used uniform perturbation budget,  $\epsilon$ , which ignores the characteristics of examples and leads to the severe issue of cross-over mixture, is a major contributor. To address the issue, our proposed MMAT considers the multiple characteristics of training examples and is fine-tuned by a teacher model to obtain a moderately inclusive decision boundary. Experiments show MMAT achieves a superior RA-NA trade-off compared to existing defenses.

## 7. Acknowledgments

This work was supported by the Key Program of Zhejiang Provincial Natural Science Foundation of China (LZ22F020007), Major Research Plan of the National Natural Science Foundation of China (92167203), National Key R&D Program of China (2018YFB2100400).

## References

- [1] N. Papernot, P. D. McDaniel, X. Wu, S. Jha, A. Swami, Distillation as a defense to adversarial perturbations against deep neural networks, in: IEEE Symposium on Security and Privacy, SP 2016, pp. 582–597.
- [2] F. Tramèr, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, P. D. McDaniel, Ensemble adversarial training: Attacks and defenses, in: 6th International Conference on Learning Representations, ICLR 2018.
- [3] S. Moosavi-Dezfooli, A. Fawzi, J. Uesato, P. Frossard, Robustness via curvature regularization, and vice versa, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, pp. 9078–9086.
- [4] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, 2018.
- [5] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, M. I. Jordan, Theoretically principled trade-off between robustness and accuracy, in: Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Vol. 97, 2019, pp. 7472–7482.
- [6] R. Rade, S.-M. Moosavi-Dezfooli, Helper-based adversarial training: Reducing excessive margin to achieve a better accuracy vs. robustness trade-off, in: ICML 2021 Workshop on Adversarial Machine Learning, 2021.
- [7] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, R. Fergus, Intriguing properties of neural networks, in: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, 2014.
- [8] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, 2015.
- [9] S. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: A simple and accurate method to fool deep neural networks, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 2016, pp. 2574–2582.

- [10] N. Carlini, D. A. Wagner, Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017, 2017, pp. 39–57.
- [11] F. Croce, M. Hein, Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, in: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Vol. 119 of Proceedings of Machine Learning Research, 2020, pp. 2206–2216.
- [12] F. Croce, M. Hein, Minimally distorted adversarial examples with a fast adaptive boundary attack, in: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Vol. 119 of Proceedings of Machine Learning Research, 2020, pp. 2196–2205.
- [13] M. Andriushchenko, F. Croce, N. Flammarion, M. Hein, Square attack: A query-efficient black-box adversarial attack via random search, in: Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Vol. 12368, 2020, pp. 484–501.
- [14] Y. Liu, X. Chen, C. Liu, D. Song, Delving into transferable adversarial examples and black-box attacks, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, 2017.
- [15] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, J. Li, Boosting adversarial attacks with momentum, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, 2018, pp. 9185–9193.
- [16] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, A. L. Yuille, Improving transferability of adversarial examples with input diversity, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, 2019, pp. 2730–2739.
- [17] J. Buckman, A. Roy, C. Raffel, I. J. Goodfellow, Thermometer encoding: One hot way to resist adversarial examples, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, 2018.
- [18] P. Samangouei, M. Kabkab, R. Chellappa, Defense-gan: Protecting classifiers against adversarial attacks using generative models, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, 2018.
- [19] G. S. Dhillon, K. Azizzadenesheli, Z. C. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, A. Anandkumar, Stochastic activation pruning for robust adversarial defense, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, 2018.
- [20] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, Q. Gu, Improving adversarial robustness requires revisiting misclassified examples, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, 2020.
- [21] J. Zhang, X. Xu, B. Han, G. Niu, L. Cui, M. Sugiyama, M. S. Kankanhalli, Attacks which do not kill training make adversarial learning stronger, in: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Vol. 119, 2020, pp. 11278–11287.
- [22] J. Cui, S. Liu, L. Wang, J. Jia, Learnable boundary guided adversarial training, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, 2021, pp. 15701–15710.
- [23] D. Mickisch, F. Assion, F. Greßner, W. Günther, M. Motta, Understanding the decision boundary of deep neural networks: An empirical study, CoRR abs/2002.01810.
- [24] S. Jetley, N. A. Lord, P. H. S. Torr, With friends like these, who needs adversaries?, in: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, 2018, pp. 10772–10782.
- [25] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, A. Madry, Adversarial examples are not bugs, they are features, in: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 2019, pp. 125–136.
- [26] T. Kim, J. Oh, N. Kim, S. Cho, S. Yun, Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, 2021, pp. 2628–2635.
- [27] G. E. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, CoRR abs/1503.02531.
- [28] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, 2018.
- [29] B. Zoph, V. Vasudevan, J. Shlens, Q. V. Le, Learning transferable architectures for scalable image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8697–8710.
- [30] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 2016, pp. 770–778.
- [31] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images.
- [32] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning.
- [33] Y. E. Nesterov, A method for solving the convex programming problem with convergence rate  $O(1/k^2)$ , in: Dokl. akad. nauk Sssr, Vol. 269, 1983, pp. 543–547.
- [34] L. N. Smith, N. Topin, Super-convergence: Very fast training of residual networks using large learning rates, CoRR abs/1708.07120.
- [35] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. P. Dickerson, C. Studer, L. S. Davis, G. Taylor, T. Goldstein, Adversarial training for free!, in: NeurIPS, 2019, pp. 3353–3364.
- [36] E. Wong, L. Rice, J. Z. Kolter, Fast is better than free: Revisiting adversarial training, in: 8th International Conference on Learning Representations, ICLR 2020, 2020.