Learnable Boundary Guided Adversarial Training

Jiequan Cui [1] Shu Liu [2] Liwei Wang [1] Jiaya Jia [1,2]
[1]The Chinese University of Hong Kong [2]SmartMore

{jqcui, lwwang, leojia}@cse.cuhk.edu.hk, liushuhust@gmail.com

Abstract

Previous adversarial training raises model robustness under the compromise of accuracy on natural data. In this paper, we reduce natural accuracy degradation. We use the model logits from one clean model to guide learning of another one robust model, taking into consideration that logits from the well trained clean model embed the most discriminative features of natural data, e.g., generalizable classifier boundary. Our solution is to constrain logits from the robust model that takes adversarial examples as input and makes it similar to those from the clean model fed with corresponding natural data. It lets the robust model inherit the classifier boundary of the clean model. Moreover, we observe such boundary guidance can not only preserve high natural accuracy but also benefit model robustness, which gives new insights and facilitates progress for the adversarial community. Finally, extensive experiments on CIFAR-10, CIFAR-100, and Tiny ImageNet testify to the effectiveness of our method. We achieve new state-of-the-art robustness on CIFAR-100 without additional real or synthetic data with auto-attack benchmark [1]. Our code is available at https://github.com/dvlab-research/LBGAT.

1. Introduction

Deep neural networks have achieved great success in many tasks, especially with the surge of neural architecture search [58, 24, 40, 11, 3]. However, with the concern of security of deep models, several methods [14, 51, 39, 36, 43, 57, 43, 17, 20, 37] have shown that deep models could be vulnerable to adversarial attack. Data that is intentionally created may easily fool strong classifiers.

In response to the vulnerability of deep neural networks, adversarial defense has become an essential topic in computer vision. There are now a sizable body of work exploring different ways to get adversarial settings, including defensive distillation [30], feature squeezing [53], randomization based methods [49, 13] and augmenting the training with adversarial examples [56, 21, 27, 43], *i.e.*, adversarial training. However, training a robust model is still challenging. Recently, adversarial training with PGD attack [27] becomes an effective defense strategy. However, when plotting results of recent work [56, 21, 27] in Fig. 1, it is still noticeable that higher robustness is often accompanied with more accuracy degradation on natural data classification.

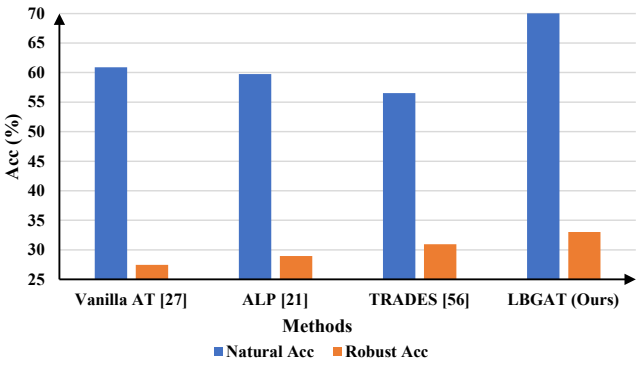
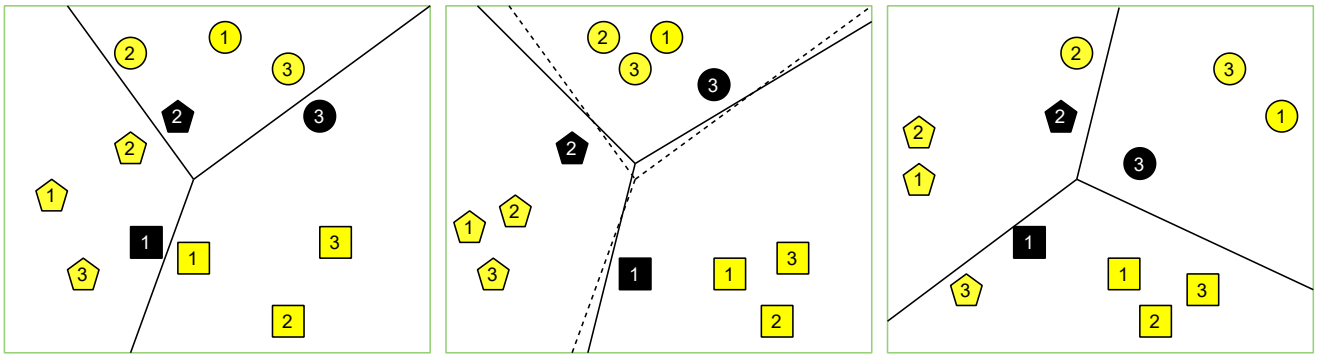


Figure 1: Model robustness on CIFAR-100 evaluated with 20 iterations PGD under white-box attack. "Natural Acc" represents classification accuracy on natural (clean) data. "Robust Acc" represents classification accuracy on adversarial data. Our method (LBGAT+TRADES with $\alpha = 0$) improves robustness with the least natural accuracy degradation.

Different from previous work that mainly pursues various ways to improve robustness, we meanwhile pursue accuracy preservation on natural data. In this paper, we propose a novel adversarial training scheme, which significantly improves classification accuracy on natural data. It also achieves high robustness under black- and white-box attack. We take advantage of logits from a clean model, which is trained only on natural data, to guide the learning of a robust model.

A conceptual illustration is shown in Fig. 2 to explain our motivation. As shown in (a), when only trained on natural (clean) data, the learned model $\mathcal{M}^{natural}$ separates natural data (plotted in yellow) well. But it may fail to classify perturbed data and misclassifies the dark circle into the

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



(a) Classifier boundary of clean model $\mathcal{M}^{natural}$

(b) Classifier boundary of our robust model $\mathcal{M}^{robust}$

(c) Classifier boundary of robust model trained with previous methods

Figure 2: Conceptual illustration of our method vs. previous adversarial training approaches. Solid lines denote real classifier boundary of the trained model, while the dotted line is the classifier boundary of the clean model $\mathcal{M}^{natural}$. Different shapes represent logits of images in various classes. Black color marks adversarial examples.

rectangle category. Previous standard adversarial training methods, *e.g.*, Madry et al. [27], mainly improve the robustness towards adversarial examples. As shown in Fig. 2(c), adversarial examples (plotted in black) can be mostly correctly classified with this strategy. However, some clean data is wrong. Thus, our motivation is to leverage the clean model $\mathcal{M}^{natural}$ to improve the natural data accuracy of $\mathcal{M}^{robust}$.

In order to seek guidance from clean model $\mathcal{M}^{natural}$, we expect the logit output of adversarial example x^{adv} from $\mathcal{M}^{robust}$ to be similar to logits output of corresponding natural data x that goes through $\mathcal{M}^{natural}$. As plotted in Fig. 2(b), the classifier boundary of our $\mathcal{M}^{robust}$ is constrained by that of the clean model, which helps classify the clean data into correct categories. At the same time, adversarial examples are also correctly labeled, benefiting from the adversarial training scheme.

Instead of constraining $\mathcal{M}^{robust}$ with the classifier boundary from one well trained static $\mathcal{M}^{natural}$, we further generalize our method to Learnable Boundary Guided Adversarial Training (LBGAT) by training $\mathcal{M}^{natural}$ and our required model $\mathcal{M}^{robust}$ at the same time to dynamically adjust the classifier boundary of $\mathcal{M}^{natural}$ and learn the most robustness-friendly one to further help $\mathcal{M}^{robust}$ enhance robustness. To show the flexibility of our method, we incorporate our model into state-of-the-art methods Adversarial Logit Pairing (ALP) [21] and TRADES [56] respectively and accomplish remarkable improvement over the baselines. Interestingly, in our exploration, we observe the classifier boundary guidance from $\mathcal{M}^{natural}$ can also enhance model robustness, which gives us new insights and potentially facilitates progress for adversarial robustness.

We conduct experiments on CIFAR-10, CIFAR-100, and Tiny ImageNet to evaluate the performance of our models under both white- and black-box attacks. Our models achieve impressive performance on these datasets and outperform previous work in a large margin. Particularly, we achieve state-of-the-art model robustness on CIFAR-100 without extra real or synthetic data under current the most popular auto-attack.

2. Related Work

2.1. Adversarial Attack

White-box Attack Szegedy et al. [39] observed that CNNs are vulnerable to adversarial examples computed by the proposed box-constrained L-BFGS attack method. Goodfellow et al. [16] attributed the existence of adversarial examples to the linear nature of networks, which yields the fast gradient sign method (FGSM) for efficiently generating adversarial examples.

FGSM was further extended to different versions of iterative attack methods. Kurakin et al. [23] showed that adversarial examples could exist in the physical world with an I-FGSM attack and iteratively applied FGSM multiple times with a small step size. Madry et al. [27] 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.

Dong et al. [14] integrated the momentum term into an iterative process for attack, called MI-FGSM, to stabilize update of directions and escape from poor local maxima during iterations. This method obtains more transferable adversarial examples. Moreover, boundary-based methods like DeepFool [29] and optimization-based methods like C&W [4] were also developed, making adversarial defense more challenging. Recently, the ensemble of diverse attack methods – auto-attack [10] by Croce et al., consisting of

APGD-CE [10], APGD-DLR [10], FAB [9], and Square Attack [1], became popular benchmark for testing model robustness.

Black-box Attack There are also many ways to explore the transferability of adversarial examples for the black-box attack. Liu et al. [25] was the first to study the transferability of targeted adversarial examples. They observed that a large proportion of target adversarial examples were able to transfer with their target labels using the proposed ensemble-based attack method. Dong et al. [14] showed that iterative attack methods incorporating the momentum term achieved better transferability. Further, Xie et al. [52] boosted the transferability of adversarial examples by creating diverse input patterns with random resize and random padding.

2.2. Adversarial Defense

Recent work focuses generally on developing defense methods to improve model robustness, including input transformation-based methods, randomization based methods [49, 13], and adversarial training [56, 21, 27, 43]. Athalye et al. [2] showed that adversarial training with PGD had withstood active attacks. Tramèr et al. [43] raised model robustness under black-box attack by the proposed ensemble adversarial training, *i.e.*, producing adversarial examples by static ensemble models. Madry et al. [27] used the universal first-order adversary, *i.e.*, PGD attack, to obtain adversarial examples in the course of adversarial training. Differently, Kannan et al. [21] enhanced model robustness with adversarial logit pairing, which encourages the logits from natural images and adversarial examples to be similar to each other in the same model.

Moreover, Zhang et al. [56] regularized the output from natural images and adversarial examples with the KL-divergence function, meanwhile using a variant of PGD attack. Xie et al. [50] studied the effect of normalization in adversarial training and proposed the Mixture BN mechanism that uses separate batch normalization layers for natural data and adversarial examples in one model. It still requires the strong assumption of knowing whether an image is natural or adversarial, at inference time, which may not be that practical.

2.3. Knowledge Distillation

Knowledge distillation was first used in [19] by Hinton et al., which was then widely applied to distill knowledge from a teacher model to a student model. The typical application of knowledge distillation is model compression, transferring from a large network or ensembles to a small network that better suits low-cost computing. Since this work, several methods [44, 31, 34, 41, 26, 42, 7] were proposed to further improve performance on model compressing and other tasks.

Goldblum et al. [15] analyzed the application of knowledge distillation in adversarial training and proposed Adversarial Robust Distillation (ARD) to transfer robustness from a large adversarially trained model to a smaller one. In this paper, we propose to use one robustness-friendly boundary learned by one natural model, not necessarily large, to guide the adversarial training without cross-entropy loss. By this way, the robust model can sufficiently inherit the classifier boundary and thus preserves high accuracy on natural data.

3. Our Method

3.1. Boundary Guided Adversarial Training

As suggested by Madry et al. [27], projected gradient descent (PGD) is a universal first-order adversary. Robust methods to defense PGD might be able to resist attack stemming from other first-order methods as well. Similarly, we use adversarial training with PGD as

$$\min_{\theta} \mathbb{E}_{(x,y)\in\hat{p}_{data}} \left(\arg\max_{\delta} \hat{L}(\theta, x+\delta, y) \right) \quad (1)$$

where $\hat{p}_{data}$ is the training data distribution, $\hat{L}(\theta, x, y)$ is the standard cross-entropy loss function with data point x and its corresponding true label y. θ represents parameters of the model, and the maximization with respect to δ is approximated using noisy BIM [23]. We denote the adversarial example $x+\delta$ across the paper as x^{adv}. Following previous work [56, 27], δ is bounded by l_∞.

Our expectation of the robust model is to achieve decent robustness and at the same time keep high accuracy on natural images. As illustrated in Fig. 2, we make use of logits from a clean model to help shape the classifier boundary of the robust model. The logits of our required robust model $\mathcal{M}^{robust}$ with x^{adv} taken as input should be similar to those of $\mathcal{M}^{natural}$ taking x as input. This relation is expressed as

$$\min_{\theta} \mathbb{E}_{(x,y)\in\hat{p}_{data}} L\left(\mathcal{M}^{robust}(x^{adv}), \mathcal{M}^{natural}(x)\right) \quad (2)$$

where L is Mean Square Error (MSE) loss function in our experiments and $\mathcal{M}(x)$ denotes the logits of model $\mathcal{M}$ taking x as input. θ is the parameter of $\mathcal{M}^{robust}$. We randomly initialize $\mathcal{M}^{robust}$ and off-line train $\mathcal{M}^{natural}$ on natural data in our experiments.

Our method can be understood from the perspective of *classifier boundary guidance*. Here we give analysis of why our method can yield high performance on natural data.

Natural Classifier Boundary Guidance Since we assume that $\mathcal{M}^{natural}$ is well trained on natural data, logits from $\mathcal{M}^{natural}$ embed more discriminative features for classification, especially the classifier boundary. According

to Eq. (2), when we impose the logits constraints, the system penalizes more on those pairs (x and x^{adv}) that have more substantial discrepancy in classification. Therefore, this logit guidance makes $\mathcal{M}^{robust}$ inherit decent classifier boundary for adversarial data. Actually, the inherited classifier boundary is still applicable to natural data in following explanation.

It is noteworthy that the adversarial example x^{adv} is located in the l_∞ ball of x. According to the min-max mechanism of PGD [27], when the adversarial training converges, the loss value corresponding to x^{adv} is always larger than the loss value corresponding to x when passing x^{adv} and x into the same model $\mathcal{M}^{robust}$. Therefore, when we pull x^{adv} into the correct class with our proposed logits constraints, x is also squeezed into the correct class. Thus the inherited classifier boundary from $\mathcal{M}^{natural}$ separates natural data well and preserves high natural accuracy.

3.2. Learnable Boundary Guided Adv. Training

For Boundary Guided Adversarial Training (BGAT) method, $\mathcal{M}^{robust}$ is constrained by logits of the static $\mathcal{M}^{natural}$. The well trained $\mathcal{M}^{natural}$ has the most desirable classifier boundary for natural data. Thus, inheriting such classifier boundary, $\mathcal{M}^{robust}$ tends to achieve high performance on natural images.

Nevertheless, the classifier boundary coming from static $\mathcal{M}^{natural}$ might not be the most suitable choice for pursuing robustness. We generalize the BGAT method to Learnable Boundary Guided Adversarial Training (LBGAT) by training $\mathcal{M}^{natural}$ and $\mathcal{M}^{robust}$ simultaneously and collaboratively. The loss function is therefore changed from Eq. (2) to

$$\min_{\theta,\theta^*} \mathbb{E}_{(x,y)\in\hat{p}_{data}} L\left(\mathcal{M}^{robust}(x^{adv}), \mathcal{M}^{natural}(x)\right) + \beta\, CE\left(\sigma(\mathcal{M}^{natural}(x)), y\right) \quad (3)$$

where x^{adv} is the adversarial example corresponding to its natural data x, and y is the true label. $\sigma(\cdot)$ is a softmax function. CE represents cross-entropy loss, $\mathcal{M}^{natural}$ and $\mathcal{M}^{robust}$ are parameterized by θ^* and θ respectively. We use Mean Square Error (MSE) loss as L function. β is the trade-off parameter. In this paper, we choose $\beta = 1$. We randomly initialize $\mathcal{M}^{robust}$ and $\mathcal{M}^{natural}$ in our experiments.

Under the regularization of the proposed logits constraints, *i.e.*, the $L(\cdot)$ loss item in Eq. (3), $\mathcal{M}^{natural}$ adaptively learns one most robustness-friendly classifier boundary during the collaborative training. At the same time, it guarantees least performance degradation on natural data with $CE(\cdot)$ loss item in Eq. (3). Note there is no additional cross-entropy loss for optimizing $\mathcal{M}^{robust}$, which makes the classifier boundary be sufficiently inherited from $\mathcal{M}^{natural}$. More details are listed in Algorithm 1.

Algorithm 1 Learnable Boundary Guided Adversarial Training (LBGAT)

Input: step size η_1 and learning rate η_2, batch size m, number of iterations K in inner optimization, model $\mathcal{M}^{robust}$ parameterized by θ, $\mathcal{M}^{natural}$ parameterized by θ^*. β is one hyper-parameter.
Output: robust model $\mathcal{M}^{robust}$ with θ.
Initialize $\mathcal{M}^{robust}$ and $\mathcal{M}^{natural}$ randomly or with pre-trained configuration.
repeat
  Read mini-batch $X = \{x_1, ..., x_m\}$, $Y = \{y_1, ..., y_m\}$ from training set;
  Get adversarial examples $X^{adv} = \{x_1^{adv}, ..., x_m^{adv}\}$ by PGD attack with input X, Y;
  $output^n = \mathcal{M}^{natural}(X)$;
  $output^r = \mathcal{M}^{robust}(X^{adv})$;
  $loss_{ce} = cross - entropy(\sigma(output^n), Y)$;
  $loss_{reg} = L(output^n, output^r)$;
  $\theta^* = \theta^* - \eta_2 \sum_{i=1}^{m} \nabla_{\theta^*}(\beta loss_{ce} + loss_{reg})/m$;
  $\theta = \theta - \eta_2 \sum_{i=1}^{m} \nabla_{\theta}(\beta loss_{ce} + loss_{reg})/m$;
until training converges

3.3. Boundary Guidance Improving Robustness

Zhang et al. [56] identified a trade-off between performance on natural data and robust accuracy. Xie et al. [48] observed that adversarial examples were helpful to model generalization ability on natural images. However, using models trained only with natural data to enhance model robustness remains unexplored. We instead notice that proper classifier boundary learned by the naturally trained model not only helps preserve high natural accuracy but also enhances model robustness (**2.44%** improvement on CIFAR-100 dataset under the strongest auto-attack [10] shown in Table 5. We attribute the improvement to the guidance of natural classifier boundary with the following explanation.

Empirically, as shown in Fig. 1, an adversarially trained model usually suffer from natural accuracy degradation, which means the adversarially trained model can not model the relations among different classes as well as the naturally trained model.

For example, with an image of a dog, the naturally trained model can misclassify it as a cat with the probability of 0.5. Under some case, we can accept this result because some dogs are very like a cat in real life. However, the adversarially trained model can misclassify a dog into a truck with high confidence because attackers can change the prediction of an image into any other class. And this is not acceptable for us because a dog is very different from a truck. Thus, with the guide of classifier boundary from a naturally trained model, the adversarially trained model can avoid such issues to some degree in training optimization.

3.4. Model Flexibility

Our method provides a new training scheme for adversarial training. It does not conflict or overlap with other adversarial training methods. We show the flexibility of our approach by using it in other state-of-the-art methods, *e.g.*, Adversarial Logit Pairing (ALP) [21] and TRADES [56]. We validate the improvement over these baselines.

Combined with Adversarial Logit Paring Adversarial logit pairing (ALP) requires the logits of natural data x and the corresponding adversarial example x^{adv} to be the same in one model, which is achieved by adding an extra mean square loss item between two logits output. We combine our BGAT with ALP as the loss of

$$\min_{\theta} \quad \mathbb{E}_{(x,y)\in\hat{p}_{data}} L\left(\mathcal{M}^{robust}(x^{adv}), \mathcal{M}^{natural}(x)\right) + \alpha\, MSE\left(\mathcal{M}^{robust}(x^{adv}), \mathcal{M}^{robust}(x)\right) \tag{4}$$

where α is a trade-off parameter. $\sigma(\cdot)$ is a softmax function and y is the true label. θ is the parameter of $\mathcal{M}^{robust}$. We replace the cross-entropy loss item $CE(\sigma(\mathcal{M}^{robust}(x^{adv})), y)$ in the original ALP loss function with our Eq. (2).

Combined with TRADES The proposed TRADES algorithm [56] explores the trade-off between model robustness and accuracy on natural data by optimizing one regularized surrogate loss. We use our BGAT in the TRADES algorithm as

$$\min_{\theta} \quad \mathbb{E}_{(x,y)\in\hat{p}_{data}} L\left(\mathcal{M}^{robust}(x^{adv}), \mathcal{M}^{natural}(x)\right) + \alpha\, D_{KL}\left(\sigma(\mathcal{M}^{robust}(x^{adv}))||\sigma(\mathcal{M}^{robust}(x))\right) \tag{5}$$

where α is still a trade-off parameter. θ is the parameter of $\mathcal{M}^{robust}$. $\sigma(\cdot)$ is softmax function and y is the true label. $D_{KL}(\cdot)$ is the boundary error term, pushing classifier boundary away from data point x, originally defined in TRADES [56]. We replace the cross-entropy loss item of $CE(\sigma(\mathcal{M}^{robust}(x)), y)$ in original TRADES loss with our Eq. (2).

It is noted that our LBGAT method can also be combined with both ALP and TRADES methods by simply replacing the first loss item in Eqs. (4) and (5) with Eq. (3).

4. Experiments

In this section, we verify the effectiveness of our methods by conducting both white- and black-box attack following the same experimental settings in [56], *i.e.*, applying $FGSM^k$ (white-box or black-box) attack with 20 iterations, perturbation size $\epsilon = 0.031$ with step size 0.003.

Table 1: Ablation study for boundary inheritance on CIFAR-10. 20 iterations PGD white-box attack is applied. We adopt ResNet18 as $\mathcal{M}^{natural}$ for LBGAT method. Acc_n represents accuracy on natural images while Acc_r represents robustness of models.

Methods	**Acc**$_n$	**Acc**$_r$
vanilla AT	86.82%	52.87%
TRADES ($\alpha = 6$)	84.92%	56.61%
LBGAT ($\alpha = 0$) (KL)	88.00%	56.10%
LBGAT ($\alpha = 0$) w/	88.35%	55.50%
LBGAT ($\alpha = 0$) w/o	88.22%	57.55%

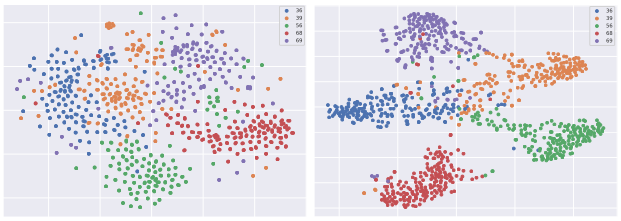
Datasets. To evaluate the robustness of our models, we conduct extensive experiments on CIFAR-10, CIFAR-100 and Tiny ImageNet datasets. CIFAR-10 dataset consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images. CIFAR-100 has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. Tiny Imagenet [12], which is with more complex data, is a miniature of ImageNet dataset. It has 200 classes. Each class has 500 training images, 50 validation images. In our experiments, we resize the image to 32x32 and normalize pixel values to [0,1]. Following [56], we perform standard data augmentation including random crops with 4 pixels of padding and random horizontal flip during training.

Training Details. We use the same neural network architecture as [56], *i.e.*, the wide residual network WRN-34-10. Following [56], We set perturbation $\epsilon = 0.031$, perturbation step size $\eta_1 = 0.007$, number of iterations $K = 10$, learning rate $\eta_2 = 0.1$, batch size $m = 128$, and number of training epochs 100 with transition epochs $\{75, 90\}$ on the training dataset. Similarly, SGD optimizer with momentum 0.9 and weight decay $2e-4$ is adopted.

4.1. Ablation Studies

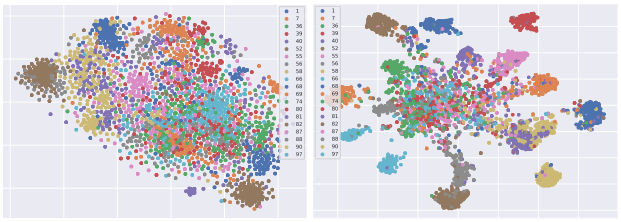
4.1.1 Natural Classifier Boundary Inheritance

To show the importance of boundary inheritance from $\mathcal{M}^{natural}$, we conduct ablation experiments with and without cross-entropy loss for $\mathcal{M}^{robust}$ in Eq. (3). Experimental results are summarized in Table 1. "w/o" additional cross-entropy loss for $\mathcal{M}^{robust}$ enjoys 2.05% higher robust accuracy than "w/", which further manifests vast importance of the natural classifier boundary inheritance. We also replace MSE loss with KL-Divergence loss in Eq. (3). KL-Divergence loss encourages the outputs of $\mathcal{M}^{robust}$ and $\mathcal{M}^{natural}$ to enjoy the same distribution while MSE loss encourages the outputs of $\mathcal{M}^{robust}$ and $\mathcal{M}^{natural}$ to have



(a) Visualization for TRADES. (b) Visualization for LBGAT.

Figure 3: Feature Visualization for LBGAT and TRADES on 5 random selected classes.



(a) Visualization for TRADES. (b) Visualization for LBGAT.

Figure 4: Feature Visualization for LBGAT and TRADES on 20 random selected classes.

the same classifier boundary. After replacing MSE with KL-Divergence, we observe performance degradation.

4.1.2 Feature Visualization

We randomly sample 5 or 20 classes in CIFAR-100. The numbers in the pictures are class indexes. For each sampled class, we collect the logit features of clean images and the corresponding adversarial examples. As shown in the figures below, LBGAT can inherit a good classifier boundary from a naturally trained model, benefiting performance on both natural data and adversarial data of the adversarially trained model.

4.1.3 Separate Batch Normalization

Xie et al. pointed that clean and adversarial images are drawn from two different domains and disentangling the mixture distribution for normalization can enhance model robustness. However, in this paper, we explore the interaction of information from those two domains, *i*.e., using classifier boundary information from clean images to assist the learning for adversarial examples.

Here we go deeper to explore whether the convolution weights can be shared in $\mathcal{M}^{natural}$ and $\mathcal{M}^{robust}$ with experiments on CIFAR-100. The experimental results are shown in Table 2. Unfortunately, we observe robustness drops.

Table 2: Ablation study for separate batch normalization. Robustness is evaluated under auto-attack. †denotes models trained with shared convolution and separate batch normalization.

Methods	**Acc**$_n$	**Acc**$_r$	**Datasets**
LBGAT ($\alpha = 0$)	70.03%	27.05%	CIFAR-100
LBGAT ($\alpha = 6$)	60.43%	29.34%	CIFAR-100
LBGAT ($\alpha = 0$) †	64.89%	24.02%	CIFAR-100
LBGAT ($\alpha = 6$) †	60.62%	27.26%	CIFAR-100

Table 3: Comparison with vanilla AT method. For BGAT, we use the ensemble of WideResNet and InceptionResNetV2 as $\mathcal{M}^{natural}$. ResNet18 as $\mathcal{M}^{natural}$ is for LBGAT on CIFAR-10 and CIFAR-100. Acc_n represents accuracy on natural images, while Acc_r represents the robustness of models.

Methods	**Acc**$_n$	**Acc**$_r$	**Datasets**
vanilla AT	60.90%	27.46%	CIFAR-100
BGAT	67.72%	30.20%	CIFAR-100
LBGAT	66.29%	34.30%	CIFAR-100
vanilla AT	86.82%	52.87%	CIFAR-10
BGAT	89.00%	55.40%	CIFAR-10
LBGAT	87.08%	56.60%	CIFAR-10

4.1.4 Effectiveness of Our Method

We first verify the effectiveness of our method compared with vanilla Adversarial Training (AT). Evaluation of model robustness is under the white-box attack using the same setting as described at the beginning of Sec. 4. Both our BGAT and LBGAT methods significantly outperform vanilla AT shown by results in Table 3. As analyzed in Sec. 3.2, the BGAT method can achieve higher natural accuracy while the LBGAT method tends to have stronger robustness. Since we aim to achieve the strongest robustness while preserving natural accuracy as high as possible, we use LBGAT by default.

4.1.5 Combing with ALP and TRADES

To verify the flexibility of our method, we show that combined with our BGAT and LBGAT methods, ALP and TRADES further improve performance. For ALP, BGAT+ALP and LBGAT+ALP methods, we adopt $\alpha = 1$ following the setting in [21]. For the TRADES method, we adopt $\alpha = 6$, with which TRADES achieves the best robustness, as demonstrated in [56].

The evaluation is under the white-box attack following the same setting as described at the beginning of Sec. 4. We summarize the results in Table 4. Equipped with regularization items of ALP and TRADES, our method can further enhance model robustness. For CIFAR-100, LBGAT+ALP

outperforms ALP by 2.92% and 6.31% respectively on natural accuracy and robust accuracy under the white-box attack respectively. Meanwhile, the BGAT+TRADES method also outperforms TRADES in terms of both natural accuracy and robustness under the white-box attack for CIFAR-10, which manifests the great flexibility of our method.

Table 4: Our method is supplementary to ALP and TRADES. For BGAT, we use the ensemble of WideResNet and InceptionResNetV2 model as $\mathcal{M}^{natural}$. ResNet18 is adopted as $\mathcal{M}^{natural}$ for LBGAT+TRADES and LBGAT+ALP. Acc_n represents accuracy on natural images while Acc_r represents robustness of models.

Methods	Acc_n	Acc_r	Datasets
ALP	59.75%	28.94%	CIFAR-100
BGAT+ALP	63.46%	31.27%	CIFAR-100
LBGAT+ALP	62.67%	35.25%	CIFAR-100
TRADES ($\alpha = 1$)	62.37%	25.31%	CIFAR-100
TRADES ($\alpha = 6$)	56.51%	30.94%	CIFAR-100
BGAT+TRADES ($\alpha = 0$)	71.27%	28.70%	CIFAR-100
LBGAT+TRADES ($\alpha = 0$)	70.03%	33.01%	CIFAR-100
LBGAT+TRADES ($\alpha = 6$)	60.43%	35.50%	CIFAR-100
ALP	85.55%	54.59%	CIFAR-10
BGAT+ALP	86.58%	55.74%	CIFAR-10
LBGAT+ALP	85.05%	57.60%	CIFAR-10
TRADES ($\alpha = 1$)	88.64%	49.14%	CIFAR-10
TRADES ($\alpha = 6$)	84.92%	56.61%	CIFAR-10
BGAT+TRADES ($\alpha = 0$)	89.06%	56.75%	CIFAR-10
LBGAT+TRADES ($\alpha = 0$)	88.22%	57.55%	CIFAR-10
LBGAT+TRADES ($\alpha = 6$)	81.98%	57.78%	CIFAR-10

4.2. Robustness on CIFAR-10 and CIFAR-100

White-box Regular Attacks. We evaluate the robustness of our models under the white-box attack using the same setting as described at the beginning of Sec. 4. For CIFAR-10, our LBGAT+TRADES ($\alpha = 0$) achieves 88.22% accuracy on natural images, which outperforms TRADES ($\alpha = 6$) by 3.3% at the same time remaining 57.55% robust accuracy, 0.94% higher than that of TRADES ($\alpha = 6$).

For CIFAR-100, our LBGAT+TRADES ($\alpha = 0$) achieves 70.03% accuracy on natural images and 33.01% robust accuracy, improving TRADES ($\alpha = 6$) by 13.53% and 2.08% respectively. Moreover, our LBGAT+TRADES ($\alpha = 6$) further boosts robustness to 57.78% and 35.50% on CIFAR-10 and CIFAR-100 respectively.

We also apply several other regular attack methods, like FGSM and CW, to evaluate our models. Compared with TRADES, our proposed methods consistently achieve better accuracy on natural images and stronger robustness on both CIFAR-10 and CIFAR-100 datasets. The details of our results are presented in Table 5. Note that the CW attack denotes using CW-loss within the PGD framework here. The evaluation under CW attack is also with 20 iterations, step size 0.003 and perturbation $\epsilon = 0.031$.

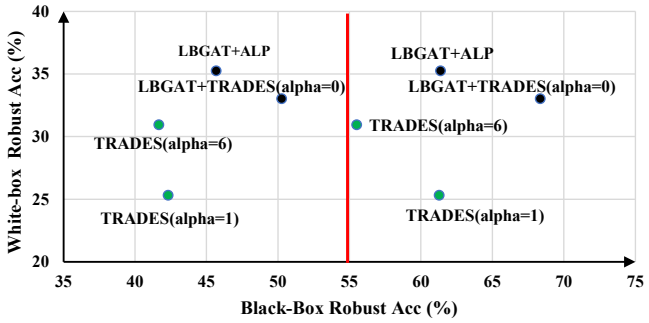


Figure 5: "White-box Robust Acc" represents classification accuracy under white-box attack. "Black-box Robust Acc" represents classification accuracy under black-box attack. Models on the right of the red line are evaluated with the clean model as the source one, while models on the left of the red line models are evaluated with the robust model as the source. More details are included in Table 7 of Appendix A.

White-box Auto-Attack (AA). Auto-Attack [10] is to reliably evaluate model robustness with an ensemble of diverse strong attack methods, including APGD-CE, APGD-DLR, FAB, and Square Attack. We use the open-source code from [10] to test our models with perturbation size 0.031. The results are listed in Table 5. Compared with TRADES ($\alpha = 6$), our LBGAT+TRADES ($\alpha = 0$) model improves natural accuracy by **13.53%** and **3.30%** on CIFAR-100 and CIFAR-10 separately, while achieving comparable robustness. Our LBGAT+TRADES ($\alpha = 6$) model further boosts robust accuracy, obtaining 29.34% and 53.14% on CIFAR-100 and CIFAR-10, outperforming TRADES ($\alpha = 6$) by **2.44%** and **0.5%** respectively.

Black-box Attacks. We verify the robustness of our models under the black-box attack. We first train models without using adversarial training on the CIFAR-10 and CIFAR-100 datasets. The same network architectures that are specified at the beginning of this section, *i.e.*, the WRN-34-10 architecture [54], are adopted. We denote these models by naturally trained models as (Natural).

The accuracy of the naturally trained WRN-34-10 model is 95.80% on the CIFAR-10 dataset and 78.76% on the CIFAR-100 dataset. We also implement the method proposed in [56] on both datasets with their open-source codebase. For both datasets, the $FGSM^k$ (black-box) method is applied to attack various defense models. We set $\epsilon = 0.031$ and apply $FGSM^k$ (black-box) attack with 20 iterations with step size set to 0.003. Note that the setup is the same as that specified in the white-box attack.

Table 5: Comparison of our method with previous defense models under white-box attack on CIFAR-10 and CIFAR-100. We use ResNet18 as $\mathcal{M}^{natural}$ for LBGAT method. Acc_n represents accuracy on natural images while Acc_r represents robustness of models. AA is the strongest attack, *i.e.*, auto-attack [10]. * denotes the model is WRN-34-20.

Defense	**Attack**	CIFAR-10		CIFAR-100	
		Acc$_n$	**Acc**$_r$	**Acc**$_n$	**Acc**$_r$
Baseline	None	95.80%	0%	78.76%	0%
TRADES ($\alpha = 1$)	$FGSM^{20}(PGD)$	88.64%	49.14%	62.37%	25.31%
TRADES ($\alpha = 6$)	$FGSM^{20}(PGD)$	84.92%	56.61%	56.50%	30.93%
LBGAT+ALP	$FGSM^{20}(PGD)$	85.05%	57.60%	62.67%	35.25%
LBGAT+TRADES ($\alpha = 0$)	$FGSM^{20}(PGD)$	88.22%	57.55%	70.03%	33.01%
LBGAT+TRADES ($\alpha = 6$)	$FGSM^{20}(PGD)$	81.98%	57.78%	60.43%	35.50%
TRADES ($\alpha = 1$)	$CW^{20}(PGD)$	88.64%	50.93%	62.37%	24.53%
TRADES ($\alpha = 6$)	$CW^{20}(PGD)$	84.92%	54.98%	56.50%	28.43%
LBGAT+ALP	$CW^{20}(PGD)$	85.05%	55.78%	62.67%	31.97%
LBGAT+TRADES ($\alpha = 0$)	$CW^{20}(PGD)$	88.22%	56.38%	70.03%	31.14%
LBGAT+TRADES ($\alpha = 6$)	$CW^{20}(PGD)$	81.98%	55.53%	60.64%	31.50%
TRADES ($\alpha = 1$)	AA	**88.64%**	48.11%	62.37%	22.24%
TRADES ($\alpha = 6$)	AA	84.92%	52.64%	56.50%	26.87%
LBGAT+TRADES ($\alpha = 0$)	AA	**88.22%**	**52.86%**	**70.03%**	**27.05%**
LBGAT+TRADES ($\alpha = 6$)	AA	81.98%	**53.14%**	60.43%	**29.34%**
LBGAT+TRADES ($\alpha = 0$)*	AA	**88.70%**	**53.58%**	**71.00%**	**27.66%**
LBGAT+TRADES ($\alpha = 6$)*	AA	83.61%	**54.45%**	62.55%	**30.20%**

The results on CIFAR-100 are summarized in Table 7 of Appendix A. We use source models to generate adversarial perturbations where the perturbation directions are according to the gradients of the source models on the input images. Our models are more robust against black-box attack transferred from naturally trained models and TRADES [56], while yielding stronger robustness under white-box attack and higher performance on natural images. Specifically, our best model is **12.83%** and **8.60%** higher than TRADES ($\alpha = 6$) with the naturally trained model and robust model as the source model separately on CIFAR-100. For robustness under black-box attack with one robust source model, our model is tested under TRADES ($\alpha = 6$) while TRADES is tested under our LBGAT trained model. More comparison between our method and TRADES is shown in Fig. 5, which exhibits results on the more challenging dataset CIFAR-100.

4.3. Robustness on Tiny-ImageNet.

To further demonstrate the effectiveness of our method on more complex data, we conduct experiments on Tiny ImageNet. Table 6 shows the experimental results. Our method is better than ALP and TRADES, surpassing baselines with a large margin. Specifically, our LBGAT+TRADES ($\alpha = 0$) outperforms the most robust baseline TRADES ($\alpha = 6$) by **9.29%** on natural data, meanwhile LBGAT+TRADES ($\alpha = 6$) is **3.00%** higher than it on adversarial data, which verifies the effectiveness of our approach again.

Table 6: Results on Tiny ImageNet [12]. The same evaluation setting with CIFAR is applied under 20-iteration PGD white-box attack. We adopt ResNet18 as $\mathcal{M}^{natural}$ for LBGAT methods. Acc_n represents accuracy on natural images while Acc_r represents robustness of models.

Methods	**Acc**$_n$	**Acc**$_r$	**Datasets**
vanilla AT	30.65%	6.81%	Tiny ImageNet
LBGAT	36.50%	14.00%	Tiny ImageNet
ALP	30.51%	8.01%	Tiny ImageNet
LBGAT+ALP	33.67%	14.55%	Tiny ImageNet
TRADES ($\alpha = 6$)	38.51%	13.48%	Tiny ImageNet
LBGAT+TRADES ($\alpha = 0$)	**47.80%**	14.31%	Tiny ImageNet
LBGAT+TRADES ($\alpha = 6$)	39.26%	**16.42%**	Tiny ImageNet

5. Conclusion

In this paper, we have proposed the Learnable Boundary Guided Adversarial Training (LBGAT) method, to improve model robustness without losing much accuracy on natural data. Our approach can be understood from the perspective of natural classifier boundary guidance. Moreover, an interesting phenomenon that the boundary guidance from a naturally trained model can also enhance model robustness is observed during our exploration. Finally, extensive experiments on CIFAR-10, CIFAR-100, and more challenging Tiny ImageNet datasets proved the effectiveness of our methods.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *ECCV*, 2020. 3

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018. 3

[3] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *ICLR*, 2020. 1

[4] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE SP*, 2017. 2

[5] Alvin Chan, Yi Tay, Yew Soon Ong, and Jie Fu. Jacobian adversarially regularized networks for robustness. In *ICLR*, 2019. 12

[6] Jinghui Chen, Yu Cheng, Zhe Gan, Quanquan Gu, and Jingjing Liu. Efficient robust training via backward smoothing. *arXiv preprint arXiv:2010.01278*, 2020. 11, 12

[7] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5008–5017, June 2021. 3

[8] Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *CVPR*, 2020. 12

[9] F. Croce and M. Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*, 2020. 3

[10] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020. 2, 3, 4, 7, 8, 12

[11] Jiequan Cui, Pengguang Chen, Ruiyu Li, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast and practical neural architecture search. In *ICCV*, 2019. 1

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5, 8

[13] Guneet S. Dhillon, Kamyar Azizzadenesheli, Zachary C. Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *ICLR*, 2018. 1, 3

[14] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018. 1, 2, 3

[15] Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. 2019. 3

[16] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[18] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *ICML*, 2019. 11, 12

[19] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. 3

[20] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 1

[21] Harini Kannan, Alexey Kurakin, and Ian J. Goodfellow. Adversarial logit pairing. *CoRR*, abs/1803.06373, 2018. 1, 2, 3, 5, 6

[22] Nupur Kumari, Mayank Singh, Abhishek Sinha, Harshitha Machiraju, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Harnessing the vulnerability of latent layers in adversarially trained models. In *IJCAI*, 2019. 12

[23] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *ICLR*, 2017. 2, 3

[24] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: differentiable architecture search. In *ICLR*. OpenReview.net, 2019. 1

[25] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *ICLR*, 2017. 3

[26] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *CoRR*, abs/1710.07535, 2017. 3

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 1, 2, 3, 4

[28] Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, and Baishakhi Ray. Metric learning for adversarial robustness. In *NeurIPS*, 2019. 12

[29] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *CVPR*, 2016. 2

[30] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE SP*, 2016. 1

[31] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, 2019. 3

[32] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. In *NeurIPS*, 2019. 12

[33] Leslie Rice, Eric Wong, and J Zico Kolter. Overfitting in adversarially robust deep learning. In *ICML*, 2020. 12

[34] Bharat Bhusan Sau and Vineeth N. Balasubramanian. Deep model compression: Distilling knowledge from noisy teachers. *CoRR*, abs/1610.09650, 2016. 3

[35] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *NeurIPS*, 2019. 12

[36] Yucheng Shi, Siyu Wang, and Yahong Han. Curls & whey: Boosting black-box adversarial attacks. In *CVPR*, June 2019. 1

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[38] Chawin Sitawarin, Supriyo Chakraborty, and David Wagner. Improving adversarial robustness through progressive hardening. *arXiv preprint arXiv:2003.09347*, 2020. 12

[39] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 1, 2

[40] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, 2019. 1

[41] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *ICLR*, 2017. 3

[42] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2020. 3

[43] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. Ensemble adversarial training: Attacks and defenses. In *ICLR*, 2018. 1, 3

[44] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *ICCV*, 2019. 3

[45] Jianyu Wang and Haichao Zhang. Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. In *ICCV*, 2019. 12

[46] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020. 12

[47] Chang Xiao, Peilin Zhong, and Changxi Zheng. Enhancing adversarial defense by k-winners-take-all, 2019. 12

[48] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *CVPR*, 2020. 4

[49] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan L. Yuille. Mitigating adversarial effects through randomization. In *ICLR*, 2018. 1, 3

[50] Cihang Xie and Alan Yuille. Intriguing properties of adversarial training. *arXiv preprint arXiv:1906.03787*, 2019. 3

[51] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. Improving transferability of adversarial examples with input diversity. In *CVPR*, June 2019. 1

[52] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. Improving transferability of adversarial examples with input diversity. In *CVPR*, 2019. 3

[53] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *NDSS*, 2018. 1

[54] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 7

[55] Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Accelerating adversarial training via maximal principle. In *NeurIPS*, 2019. 12

[56] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019. 1, 2, 3, 4, 5, 6, 7, 8, 12

[57] Tianhang Zheng, Changyou Chen, and Kui Ren. Distributionally adversarial attack. In *AAAI*, 2019. 1

[58] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *CVPR*, 2018. 1