Adversarial Distillation Training

Anonymous CVPR submission

Paper ID 644

Abstract

Previous adversarial training methods usually raise models' robustness by sacrificing accuracy on natural data. In this paper, our target is to enhance models' robustness with the least accuracy degradation. Logits from one well trained clean model $\mathcal{M}^{natural}$ embed the most discriminative features of natural data. We, therefore, constrain the logits from the robust model $\mathcal{M}^{robust}$ with adversarial examples as inputs to be similar to logits from clean model $\mathcal{M}^{natural}$ fed with corresponding natural data. We name our method as Adversarial Distillation Training (ADT) since $\mathcal{M}^{natural}$ acts as a teacher model to help with the learning of our robust model. Moreover, we generalize the ADT method to Adversarial Online Distillation Training (AODT) by training $\mathcal{M}^{natural}$ and $\mathcal{M}^{robust}$ simultaneously, which further improves the robustness. Extensive experiments have been conducted on the CIFAR-10 dataset and the more challenging CIFAR-100 dataset, showing the advantages of our methods. When equipped with other state-of-the-art adversarial training approaches, e.g., Adversarial Logit Pairing (ALP) [10] and TRADES [31], the performances of our methods can be further improved.

1. Introduction

Deep neural networks have achieved great success in many tasks of artificial intelligence. With a surge of exploration in security of deep models, lots of works [5, 26, 21, 19, 23, 33, 23, 7, 9, 20] have shown that deep models today are vulnerable to adversarial attacks. Data that have been intentionally optimized can easily fool strong classifiers.

In response to the vulnerability of deep neural networks, the adversarial defense has become an essential topic in computer vision. There are now a sizable body of works exploring different angles in the adversarial settings, including defensive distillation [16], feature squeezing [28], randomization based methods [25, 4] and augmenting the training with adversarial examples [31, 10, 14, 23], *i.e.* adversarial training. However, training a robust model is still challenging. Currently, adversarial training with PGD attack [14] has become the most effective defense method with its advantageous performance over other approaches. However, when plotting recent works [31, 10, 14] in Fig. (1), we observe that stronger robustness often comes with more accuracy degradation on natural data classification.

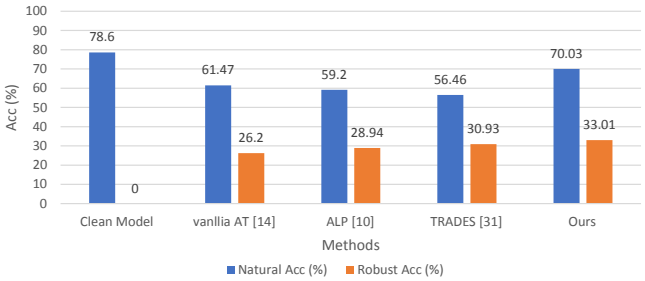
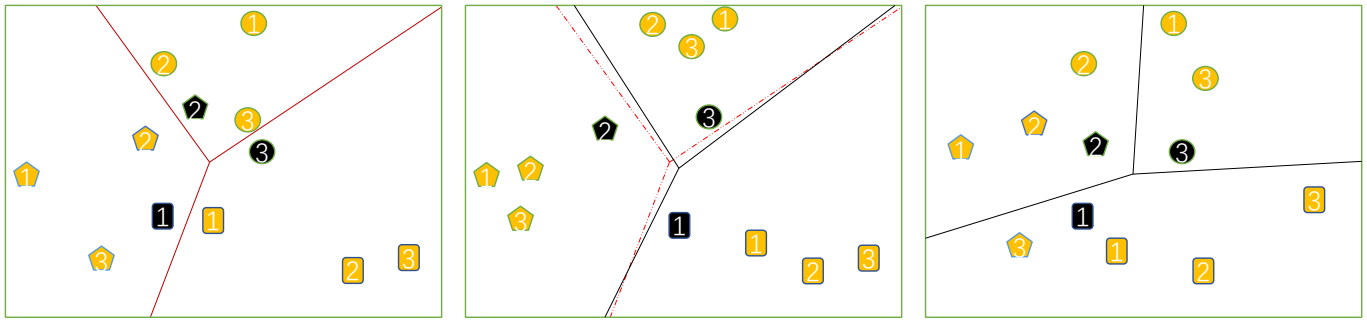


Figure 1: Models' robustness on CIFAR-100 evaluated with 20 iterations PGD under white-box attack. "Natural Acc" represents classification accuracy on natural (clean) data. "Robust Acc" represents classification accuracy on adversarial data. Our method (AODT+TRADES with $\alpha = 0$) improves model's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

Different from previous works that mainly pursue various ways to improve robustness, we ask the following question and focus on seeking the solution to it in this paper:

Can we train a robust model that can improve robustness towards adversarial examples as well as preserve accuracy on natural data?

In this paper, we provide a novel adversarial training scheme, which can significantly improve the classification accuracy on natural data and robustness under black-box attack and white-box attacks. We take advantage of logit from a clean model, which is trained only with natural data, to guide the learning of a robust model. We draw a conceptual illustration in Fig. (2) to explain our motivation. As shown in the left-most plot of Fig. (2), when only trained on natural (clean) data, the learned model $\mathcal{M}^{natural}$ can separate natural data (plotted in yellow color) well, while easily fail to classify perturbed data, e.g., misclassifying the dark circle



Classifier boundaries of clean model $M^{natural}$.

Classifier boundaries of robust model M^{robust} trained with our methods

Classifier boundaries of robust model trained with previous methods

Figure 2: A conceptual illustration of our methods vs. previous adversarial training methods. Solid lines denote real classifier boundaries of the trained model while dotted lines are the classifier boundaries of the clean model $\mathcal{M}^{natural}$. Different shapes represent logits of images in different class and shapes in black color are adversarial examples.

into the rectangle category. What other standard adversarial training methods do, e.g., PGD [14], is mainly to improve the resistance against adversarial examples. Like the rightmost plot in Fig. (2) shows, adversarial samples (plotted in black color) are correctly classified. However, some clean data are wrongly classified due to the learning of $\mathcal{M}^{robust}$. Therefore, our approach is to seek guidance from the clean model $\mathcal{M}^{natural}$ to improve the natural data accuracy of $\mathcal{M}^{robust}$ as well as strengthen its robustness. As plotted in the middle of Fig. (2), the classifier boundary of our $\mathcal{M}^{robust}$ is affected by that of the clean model, which helps to classify the clean data into the correct categories. At the same time, adversarial samples are also correctly labeled, benefiting from the adversarial training scheme.

In order to seek guidance from clean model $\mathcal{M}^{natural}$, we expect the logit output of adversarial sample x^{adv} from $\mathcal{M}^{robust}$ to be the same with the logit output of corresponding x that goes through $\mathcal{M}^{natural}$. The idea of using logits to guide the learning is inspired by recent works of Adversarial Logit Pairing (ALP) [10] and knowledge distillation [8]. Intuitively, logits of a clean model are the most discriminative features of natural data. Our method regularizes logits output of the robust model by a more confident classifier on natural data, which enables improving the accuracy.

We can also understand our approach from the view of knowledge distillation [8]. The clean model trained on natural data usually comes with a high accuracy, which can be seen as a teacher model. Then $\mathcal{M}^{robust}$) is the student model. To fit into the theory of distillation that the same data samples are put into both teacher network and student network, we virtually construct a pseudo teacher model that takes the input of x^{adv} and output the logit, which is the same as the logit of x from $\mathcal{M}^{natural}$. Therefore, our robust model is actually trained to distill from the clean model, *i.e.* to improve accuracy on natural data.

We incorporate this logits guide into the vanilla adversarial training scheme, obtaining our approach named *Adversarial Distillation Training method* (ADT). Inspired by[23] which proposed mutual learning, we further generalize our ADT method to Adversarial Online Distillation Training (AODT) by training $\mathcal{M}^{natural}$ and our required model $\mathcal{M}^{robust}$ at the same time. To show the flexibility of our methods, we plug our models into state-of-the-art methods Adversarial Logit Pairing (ALP) [10] and TRADES [31] respectively and achieve remarkable improvements over other baselines.

We conduct experiments on CIFAR-10 and CIFAR-100 to evaluate the performance of our models under both white-box and black-box attacks. Our models achieve the best performance on both data sets. For example, on CIFAR-10, our "ADT+TRADES($\alpha = 0$)" improves natural data classification accuracy from 84.92% to 89.06%. At the same time, the robust accuracy is remained at 56.75% under white-box attack, surpassing "TRADES($\alpha = 6$)". On CIFAR-100, our "AODT+TRADES($\alpha = 0$)" achieves 70.03% on natural data and 33.01% on adversarial data, outperforming "TRADES($\alpha = 6$)" by 13.53% and 2.08% respectively. Moreover, our most robust model boosts the robustness to 57.78% and 35.50% under the white-box attack, improving "TRADES($\alpha = 6$)" by 1.17% and 4.56% on CIFAR-10 and CIFAR-100 respectively.

2. Related Work

2.1. Adversarial Attacks

White-box attack Adversarial examples have always been an active topic since Szegedy et al. [21] observed

that CNNs are vulnerable to adversarial examples computed by the proposed box-constrained L-BFGS attack method. Goodfellow et al. [6] 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. Later on, FGSM was further extended to different versions of iterative attack methods. Kurakin et al. [11] showed that adversarial examples could exist in the physical world with an I-FGSM attack which iteratively applies FGSM multiple time with a small step size. Madry et al. [14] proposed Projected Gradient Descent (PGD) method as a universal "first-order adversary," i.e., the most active attack is utilizing the local first-order information about the network. Dong et al. [5] integrated the momentum term into an iterative process for attacks, which was called MI-FGSM, to stabilize update directions and escape from poor local maxima during the iterations, thus obtaining more transferable adversarial examples. Moreover, boundary-based methods like DeepFool [15] and optimization-based methods like C&W [2] are also developed, making the adversarial defense more and more challenging.

Black-box attack There are also many works exploring the transferability of adversarial examples for the black-box attack. Liu et al. [12] was the first to study the transferability of targeted adversarial examples. They observed a large proportion of targeted adversarial examples were able to transfer with their target labels using the proposed ensemble-based attack method. Dong et al. [5] showed that iterative attack methods incorporating the momentum term achieved better transferability. Further, Xie et al. [27] boosted the transferability of adversarial examples by creating diverse input patterns with random resize and random padding tricks.

2.2. Adversarial Defense

Many recent works focus on developing defense methods to improve models' robustness, including input transformation-based methods, randomization based methods [25, 4], and adversarial training [31, 10, 14, 23]. Athalye et al. [1] showed that adversarial training with PGD had withstood active attacks. Tramèr et al. [23] raised models' robustness under black-box attack by the proposed ensemble adversarial training, *i.e.* producing adversarial examples by static ensemble models. Madry et al. [14] used the universal first-order adversary, *i.e.* PGD attack, to obtain adversarial examples in the course of adversarial training. Instead, Kannan et al. [10] enhanced models' robustness with adversarial logit pairing technique, which encouraged the logits from natural images and adversarial examples to be similar to each other. Moreover, Zhang et al. [31] regularized the outputs from natural images and adversarial examples with KL-divergence function, meanwhile using a variant of PGD attack method.

2.3. Knowledge distillation

Knowledge distillation first appeared in [8] by Hinton et al., which was then widely used 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 is better suited to low-cost computing. Since then, lots of works [24, 17, 18, 22, 13] were proposed to further improve performance on model compressing and other tasks. In this work, we incorporate the knowledge distillation technique into adversarial training to improve the performance of models. To our best knowledge, this is the first time that adversarial training and knowledge distillation are combined for models' robustness with the help of priors from clean models trained only with natural images.

3. Method

3.1. Adversarial Distillation Training

As suggested by Madry et al. [14], projected gradient descent (PGD) is a universal first-order adversary. Robust methods developed to defense PGD might be able to resist attacks from other first-order attacks as well. Similar to ALP [10], we use adversarial training with PGD in our method:

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

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 [11]. We denote the adversarial sample $x+\delta$ across the paper as x^{adv}. Following previous works [31, 14], δ is bounded by l_∞.

Our expectation of the robust model is to achieve good robustness at the same time keep high accuracy on natural images. ALP [10] achieves this by pairing logits from x and x^{adv} outputted by the same model, *i.e.* x and x^{adv} pass the same model in the mixed mini-batches during training. However, ours is totally different. Logits from a well trained clean model $\mathcal{M}^{natural}$ embed the most discriminative featueres of input data. As illustrated in Fig. (2),we make use of logits from a clean model to help shape the classification boundaries of the robust model, *i.e.* we impose the logits of our required robust model $\mathcal{M}^{robust}$ with x^{adv} as inputs to be similar to the logits of $\mathcal{M}^{natural}$ taking x as inputs, that is:

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

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 inputs. 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.

We provide two different angles to understand our algorithms: *classification logits guidance* and *knowledge distillation* from clean models.

Classification Logits Guidance. According to Eq.(2), when we impose the logits constraints, we penalize more on those pairs (x and x^{adv}), that have more substantial discrepancies in classification. On one hand, this might help to pull x^{adv} into the correct category since we hypothesize $\mathcal{M}^{natural}$ is well trained on natural data and x should be more likely given the correct label.

On the other hand, we should remember that adversarial example x^{adv} is located in the l_∞ ball of x. According to the min-max mechanism of PGD [14], in the current training iteration, 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, it means the x is also likely to be squeezed into the correct class. This can explain why our $\mathcal{M}^{robust}$ is able to improve robustness against adversarial examples while keep accuracy on natural data.

Knowledge Distillation. We can also understand our approach from the view of knowledge distillation. $\mathcal{M}^{natural}$ can achieve a high accuracy on natural images, which can be considered as a teacher network. We distill knowledge from $\mathcal{M}^{natural}$ to train our robust model of $\mathcal{M}^{robust}$. To fit into the framework of knowledge distillation, we virtually create a bridge model $\mathcal{M}^{\hat{robust}}$. The bridge model $\mathcal{M}^{\hat{robust}}$ takes the same input of $\mathcal{M}^{robust}$ as x^{adv} and outputs the same logits of $\mathcal{M}^{natural}$ with x as input. In this way, the loss term in Eq.(2) can be re-written as $L\left(\mathcal{M}^{robust}(x^{adv}), \mathcal{M}^{\hat{robust}}(x^{adv})\right)$ and thus can be explained by the distillation scheme.

Therefore, we name our method as the Adversarial Distillation Training (ADT) and list the algorithm details in Algorithm 1. Moreover, $\mathcal{M}^{nature}$ and $\mathcal{M}^{robust}$ need not to be with the same network structure. $\mathcal{M}^{natural}$ can even be an ensemble of models with different structures. We will explore performances of varying network structures of $\mathcal{M}^{natural}$ in supplementary files.

3.2. Adversarial Online Distillation Training

Inspired by [32] which suggets a deep mutual learning strategy, *i.e.* instead of transferring knowledge from a static teacher network to a student network, multiple models can teach each other and learn collaboratively during the training process. We therefore generalize our ADT method to Adversarial Online Distillation Training (AODT) by training $\mathcal{M}^{natural}$ and $\mathcal{M}^{robust}$ simultaneously and collaboratively. The loss function is therefore changed from Eq.(2) to:

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

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. β is the trade-off parameter. In this paper, we choose $\beta = 1$ for experiments. We randomly initialize $\mathcal{M}^{robust}$ and $\mathcal{M}^{natural}$ in our experiments.

In ADT method, $\mathcal{M}^{robust}$ is constrained by fixed logits provided by the static $\mathcal{M}^{natural}$. The well trained $\mathcal{M}^{natural}$ can provide the most desirable classifier boundaries. Thus $\mathcal{M}^{robust}$ tends to achieve higher performance on natural images under this case. However, the fixed logits provided by $\mathcal{M}^{natural}$ may not be the best choice for pursuing the robustness. In AODT method, with loss function 3, the $CE(\cdot)$ loss item encourages $\mathcal{M}^{natural}$ to converge and shape the ideal classifier boundaries for $\mathcal{M}^{robust}$ pursuing strong robustness under the guide of $L(\cdot)$ loss item. Therefore, in experiments, we observe the robustness of $\mathcal{M}^{robust}$ is further improved by AODT.

Algorithm 1 Adversarial Distillation Training (ADT)

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, $\mathcal{M}^{natural}$ trained on natural images.
Output: Robust model $\mathcal{M}^{robust}$ with θ;
Randomly initialize $\mathcal{M}^{robust}$ or initialize $\mathcal{M}^{robust}$ with pre-trained configuration. And load pre-trained $\mathcal{M}^{natural}$.
repeat
  Read mini-batch $B = \{x_1, ..., x_m\}$ from training set;
  Get adversarial examples $B^{adv} = \{x_1^{adv}, ..., x_m^{adv}\}$ by PGD attack with inputs B.
  $output^n = \mathcal{M}^{natural}(B)$.
  $output^r = \mathcal{M}^{robust}(B^{adv})$
  $\theta = \theta - \eta_2 \sum_{i=1}^{m} \nabla_\theta L(output^r, output^n)/m$
until training converged

3.3. Model Flexibility

Our method provides a new training scheme for adversarial training. Thus it is orthogonal to other adversarial training methods. We show the flexibility of our approach by plugging it into other state-of-the-art methods, e.g., Ad-

versarial Logit Pairing (ALP) [10], TRADES method [31] and validating the improvements compared with baselines.

Combined with Adversarial Logit Paring. Adversarial logit pairing (ALP) requires the logits of natural data x and its 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 ADT with ALP as the following loss:

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

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 original ALP loss function with our Eq.(2).

Combined with TRADES. The proposed TRADES algorithm [31] explores the trade-off between model's robustness and its accuracy on natural data by optimizing one regularized surrogate loss. We plug our ADT into the TRADES algorithm and thus have:

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

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 [31]. We replace the cross-entropy loss item of $CE(\sigma(\mathcal{M}^{robust}(x)), y)$ in original TRADES loss with our Eq. (2).

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

4. Experiments

In this section, we verify the effectiveness of our methods by conducting both white-box and black-box attacks following the same experimental settings in [31], i.e. applying $FGSM^k$ (white-box or black-box) attack with 20 iterations, perturbation size $\epsilon = 0.031$ and the step size 0.003.

Datasets. To evaluate the robustness of our models, we conduct extensive experiments on CIFAR-10 and CIFAR-100 datasets. CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. CIFAR-100 dataset, which is more challenge than CIFAR-10, has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. Following [31], we perform standard data augmentation including random crops with 4 pixels of padding and random horizontal flips during training.

Training details. We use the same neural network architecture as [31], i.e. the wide residual network WRN-34-10. Following [31], 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, the number of training epochs as 100 with transition epochs as {75,90} on the training dataset. Similarly, SGD optimizer with momentum 0.9 and weight decay 2e-4 are adopted.

4.1. Effectiveness of our methods

We first show the effectiveness of our methods compared with vanilla Adversarial Training (AT). The evaluation of models' robustness is under the white-box attack using the same setting as described at the beginning of Sec.4. Both our ADT and AODT methods can significantly surpass vanilla AT, as demonstrated by results in Table 2. As we analyzed in Sec. 3.2, ADT method can achieves higher natural accuracy while the AODT method tends to have stronger robustness.

4.2. Combing with ALP and TRADES

To verify the flexibility of our method, we show that, combined with our ADT and AODT methods, ALP and TRADES can further improve performance. For ALP, ADT+ALP and AODT+ALP methods, we adopt $\alpha = 1$ following the setting in [10]. For the TRADES method, we adopt $\alpha = 6$ with which TRADES achieves the best robustness, as demonstrated in [31].

The evaluation of models' robustness is under the white-box attack following the same setting as described at the beginning of Sec.4. We summarize the results in Table 3. AODT+ALP method outperforms ALP methods by 2.92% and 6.31% respectively on natural accuracy and robust accuracy under a white-box attack. Meanwhile, the TRADES+ADT method also surpassing the TRADES method on both natural accuracy and robustness under a white-box attack, which demonstrates the great flexibility of our method.

4.3. Robustness on CIFAR-10 and CIFAR-100

White-box attack We evaluate the robustness of our models under white-box attack using the same setting as described in the beginning of Sec.4. For CIFAR-10, our ADT+TRADES($\alpha = 0$) achieves 89.06% accuracy on natural images which outperforms TRADES($\alpha = 6$) by 4.14% at the same time remaining 56.75% robust accuracy which surpasses TRADES($\alpha = 6$). For CIFAR-100, our AODT+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.

Defense	Under which attack	CIFAR-10		CIFAR-100	
		Acc$_n$	**Acc**$_r$	**Acc**$_n$	**Acc**$_r$
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%
ADT+TRADES($\alpha = 0$)	$FGSM^{20}(PGD)$	**89.06%**	56.75%	**71.27%**	28.70%
AODT+ALP	$FGSM^{20}(PGD)$	85.05%	57.60%	62.67%	35.25%
AODT+TRADES($\alpha = 0$)	$FGSM^{20}(PGD)$	88.22%	57.55%	70.03%	33.01%
AODT+TRADES($\alpha = 6$)	$FGSM^{20}(PGD)$	81.98%	**57.78%**	60.43%	**35.50%**
TRADES($\alpha = 6$)+Q	$FGSM^{20}(PGD)$	84.45%	58.13%	56.47%	32.03%
ADT+TRADES+Q($\alpha = 0$)	$FGSM^{20}(PGD)$	**86.89%**	**60.53%**	**71.40%**	30.00%
AODT+ALP+Q	$FGSM^{20}(PGD)$	83.44%	59.98%	62.70%	36.32%
AODT+TRADES+Q($\alpha = 0$)	$FGSM^{20}(PGD)$	85.99%	59.86%	70.05%	34.42%
AODT+TRADES+Q($\alpha = 6$)	$FGSM^{20}(PGD)$	80.59%	59.40%	60.76%	**36.39%**
TRADES($\alpha = 6$)+Q	$FGSM^{1000}(PGD)$	84.45%	56.78%	56.47%	31.56%
ADT+TRADES+Q($\alpha = 0$)	$FGSM^{1000}(PGD)$	**86.89%**	**58.92%**	**71.40%**	28.00%
AODT+ALP+Q	$FGSM^{1000}(PGD)$	83.44%	58.68%	62.70%	35.49%
AODT+TRADES+Q($\alpha = 0$)	$FGSM^{1000}(PGD)$	85.99%	58.27%	70.05%	32.87%
AODT+TRADES+Q($\alpha = 6$)	$FGSM^{1000}(PGD)$	80.59%	58.41%	60.76%	**35.99%**
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%
ADT+TRADES($\alpha = 0$)	$CW^{20}(PGD)$	**89.06%**	**56.90%**	**71.27%**	28.69%
AODT+ALP	$CW^{20}(PGD)$	85.05%	55.78%	62.67%	**31.97%**
AODT+TRADES($\alpha = 0$)	$CW^{20}(PGD)$	88.22%	56.38%	70.03%	31.14%
AODT+TRADES($\alpha = 6$)	$CW^{20}(PGD)$	81.98%	55.53%	60.64%	31.50%
TRADES($\alpha = 6$)+Q	$CW^{20}(PGD)$	84.45%	69.27%	56.47%	41.85%
ADT+TRADES+Q($\alpha = 0$)	$CW^{20}(PGD)$	**86.89%**	**71.75%**	**71.40%**	48.33%
AODT+ALP+Q	$CW^{20}(PGD)$	83.44%	69.39%	62.70%	46.43%
AODT+TRADES+Q($\alpha = 0$)	$CW^{20}(PGD)$	85.99%	70.31%	70.05%	**49.00%**
AODT+TRADES+Q($\alpha = 6$)	$CW^{20}(PGD)$	80.59%	68.31%	60.76%	45.91%
TRADES($\alpha = 6$)+Q	$CW^{1000}(PGD)$	84.45%	68.90%	56.47%	41.18%
ADT+TRADES+Q($\alpha = 0$)	$CW^{1000}(PGD)$	**86.89%**	**70.71%**	**71.40%**	48.00%
AODT+ALP+Q	$CW^{1000}(PGD)$	83.44%	68.60%	62.70%	45.72%
AODT+TRADES+Q($\alpha = 0$)	$CW^{1000}(PGD)$	85.99%	69.69%	**70.05%**	**48.23%**
AODT+TRADES+Q($\alpha = 6$)	$CW^{1000}(PGD)$	80.59%	67.68%	60.76%	45.29%
TRADES($\alpha = 1$)	FGSM	88.64%	86.04%	62.37%	58.32%
TRADES($\alpha = 6$)	FGSM	84.92%	82.07%	56.50%	53.82%
ADT+TRADES($\alpha = 0$)	FGSM	**89.06%**	**86.88%**	**71.24%**	**66.59%**
AODT+ALP	FGSM	85.05%	83.19%	62.67%	59.87%
AODT+TRADES($\alpha = 0$)	FGSM	88.22%	86.14%	70.03%	66.42%
AODT+TRADES($\alpha = 6$)	FGSM	81.98%	80.08%	60.64%	58.16%

Table 1: Comparison of our method with previous defense models under white-box attacks on CIFAR-10 and CIFAR-100. Q denote the input quantization trick [28] that applied at inference time. The choice of hyper parameter for quantization is obtained by grid search for all methods. We use the ensemble of WideResNet[29] and InceptionResNetV2 as $\mathcal{M}^{natural}$. ResNet152 and ResNet18 are dopted as $\mathcal{M}^{natural}$ for AODT+TRADES and AODT+ALP methods separately. To rule out randomness, the numbers are averaged over 2 independently trained models. Acc_n represents accuracy on natural images while Acc_r represents robustness of models.

Moreover, our AODT+TRADES($\alpha = 6$) further boosts the robustness to 57.78% and 35.50% on CIFAR-10 and CIFAR-100 separately. Also, we apply several other attack methods to evaluate our models. Compared with TRADES, our proposed methods always can achieve better accuracy on natural images and stronger robustness on both CIFAR-10 and CIFAR-100 datasets. The details of our results are shown in Table 1. Note that the CW attack denotes using CW-loss within the PGD framework here. The evaluation under CW attacks are also with 20 iterations, step size 0.003 and perturbation $\epsilon = 0.031$. To show the advantages of our methods more intuitively, we plot the comparison results on the more challenging CIFAR-100 dataset in Fig. 3, which indicates that models trained with our method can

Method	Acc_n	Acc_r	Dataset
vanilla AT	60.90%	27.46%	CIFAR-100
ADT	67.72%	30.20%	CIFAR-100
AODT	66.29%	34.30%	CIFAR-100
vanilla AT	86.82%	52.87%	CIFAR-10
ADT	89.00%	56.07%	CIFAR-10
AODT	87.08%	56.60%	CIFAR-10

Table 2: Effectiveness of our methods by comparison with vanilla AT method. For ADT method, we use the ensemble of WideResNet and InceptionResNetV2 model as $\mathcal{M}^{natural}$. ResNet18 and ResNet152 as $\mathcal{M}^{natural}$ are for AODT method on CIFAR-10 and CIFAR-100 respectively. To rule out randomness, we get numbers by running two independently trained models and taking the average. Acc_n represents accuracy on natural images, while Acc_r represents the robustness of models.

Method	Acc_n	Acc_r	Dataset
ALP	59.75%	28.94%	CIFAR-100
ADT+ALP	63.46%	31.27%	CIFAR-100
AODT+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
ADT+TRADES($\alpha = 0$)	71.27%	28.70%	CIFAR-100
AODT+TRADES($\alpha = 0$)	70.03%	33.01%	CIFAR-100
AODT+TRADES($\alpha = 6$)	60.43%	35.50%	CIFAR-100
ALP	85.55%	54.59%	CIFAR-10
ADT+ALP	86.58%	55.74%	CIFAR-10
AODT+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
ADT+TRADES($\alpha = 0$)	89.06%	56.75%	CIFAR-10
AODT+TRADES($\alpha = 0$)	88.22%	57.55%	CIFAR-10
AODT+TRADES($\alpha = 6$)	81.98%	57.78%	CIFAR-10

Table 3: Our method are orthogonal to ALP and TRADES. For ADT method, we use the ensemble of WideResNet and InceptionResNetV2 model as $\mathcal{M}^{natural}$. ResNet152 and ResNet18 are adopted as $\mathcal{M}^{natural}$ for AODT+TRADES and AODT+ALP method respectively. To rule out randomness, the numbers are averaged over 2 independently trained models. Acc_n represents accuracy on natural images while Acc_r represents robustness of models.

have stronger robustness while preserving high natural accuracy.

Black-box attack We verify the robustness of our models under black-box attacks. 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

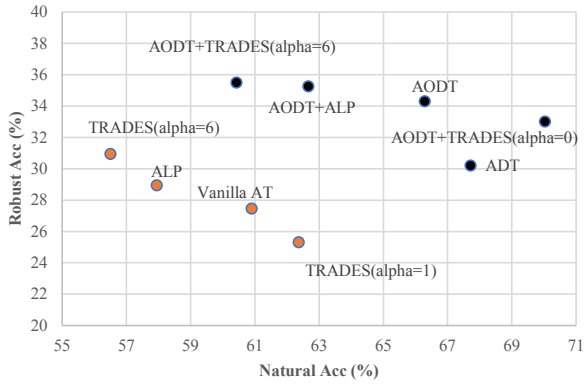


Figure 3: Comparisons with state-of-the-art defense methods on CIFAR-100 with 20 iterations PGD under white-box attack. Black dots represent our methods, while orange points represent other recent methods. More details refer to Tables 1, 2, 3.

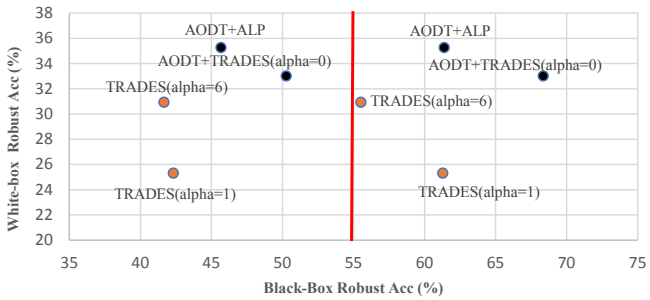


Figure 4: "White-box Robust Acc" represents classification accuracy under white-box attack. "Black-box Robust Acc" represents classification accuracy under black-box attack. Models on the right of the red line are evaluated with clean model as the source model, while models on the left of the red line models are evaluated with robust model as the source model. More details refer to Table 4.

architecture[29], are adopted. We denote these models by naturally trained models (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 [31] on both datasets with their open-sourced 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 and the step size is set to 0.003. Note that the setup is the same as the setup specified in the white-box attack.

The results are summarized in Table 4. To evaluate, we use source models to generate adversarial perturbations: we

Target Model	BAcc_{robust}	WAcc_{robust}	$\text{Acc}_{natural}$	Source Model	Dataset
TRADES($\alpha = 1$)	87.41%	49.14%	88.64%	Nature	CIFAR-10
TRADES($\alpha = 6$)	83.30%	56.61%	84.92%	Natural	CIFAR-10
ADT+TRADES($\alpha = 0$)	**87.91%**	56.75%	**89.06%**	Natural	CIFAR-10
AODT+TRADES($\alpha = 0$)	87.00%	**57.55%**	88.22%	Natural	CIFAR-10
TRADES($\alpha = 1$)	66.18%	49.14%	88.64%	ADT+TRADES(Ours)	CIFAR-10
TRADES($\alpha = 6$)	67.18%	56.61%	84.92%	ADT+TRADES(Ours)	CIFAR-10
ADT+TRADES($\alpha = 0$)	**70.47%**	56.75%	**89.06%**	TRADES($\alpha = 6$)	CIFAR-10
AODT+TRADES($\alpha = 0$)	68.45%	**57.55%**	88.22%	TRADES($\alpha = 6$)	CIFAR-10
TRADES($\alpha = 1$)	61.29%	25.31%	62.37%	Nature	CIFAR-100
TRADES($\alpha = 6$)	55.52%	30.93%	56.51%	Natural	CIFAR-100
AODT+ALP	61.38%	**35.25%**	62.67%	Natural	CIFAR-100
AODT+TRADES($\alpha = 0$)	**68.35%**	33.01%	**70.03%**	Natural	CIFAR-100
TRADES($\alpha = 1$)	42.32%	25.31%	62.37%	AODT+TRADES($\alpha = 0$)	CIFAR-100
TRADES($\alpha = 6$)	41.67%	30.93%	56.51%	AODT+TRADES($\alpha = 0$)	CIFAR-100
AODT+ALP	45.68%	**35.25%**	62.67%	TRADES($\alpha = 6$)	CIFAR-100
AODT+TRADES(α=0)	**50.27%**	33.01%	**70.03%**	TRADES($\alpha = 6$)	CIFAR-100

Table 4: Comparison of our method with previous defense models under black-box attacks on CIFAR-10 and CIFAR-100. To rule out randomness, the numbers are averaged over 2 independently trained models. Acc_n represents accuracy on natural images while Acc_r represents robustness of models.

compute the perturbation directions according to the gradients of the source models on the input images. It shows that our models are more robust against black-box attacks transferred from naturally trained models and TRADES [31]'s models while keeping stronger robustness under white-box attack and higher performance on natural images. Moreover, our models can generate stronger adversarial examples for black-box attacks compared with naturally trained models and TRADES [31]'s models. A more clear comparison between our method and TRADES method can be seen in Fig. 4, which exhibits the results on the more challenging dataset CIFAR-100.

4.4. Choice of L function

We have tried four commonly used functions in knowledge distillation as the L function to explore their effects on our AODT methods. For convenience, $\mathcal{M}^{robust}(x^{adv})$, $\mathcal{M}^{natural}(x)$, softmax($\mathcal{M}^{robust}(x^{adv})/T$), softmax($\mathcal{M}^{natural}(x)/T$), softmax($\mathcal{M}^{robust}(x^{adv})$), softmax($\mathcal{M}^{natural}(x)$) are denoted by O^r, O^n, O^r_{st}, O^n_{st}, O^r_s and O^n_s respectively. The functions are as follows:

$$L_1 = MSE(O^r, O^n) \tag{6}$$

$$L_2 = MSE(O^r, O^n) + CE(O^r_s, y) \tag{7}$$

$$L_3 = (1-\beta)CE(O^r_s, y) + 2\beta T^2 CE(O^r_{st}, O^n_{st}) \tag{8}$$

$$L_4 = D_{KL}(O^r_s || O^n_s) + CE(O^r_s, y) \tag{9}$$

where CE is the cross-entropy loss, y is the true label, D_{KL} is KL-divergence loss and T is hyper-parameter in knowledge distillation loss. Following previous works [30, 3], we set T=4 and β=0.1 for knowledge distillation loss L_3 [24]. ResNet18 as $\mathcal{M}^{natural}$ is adopted in experiments on CIFAR-10 dataset. The models are evaluated under white-box attack with the same setting as mentioned in the beginning of Sec. 4. The results are summarized in Table 5. We observe that AODT method equipped with MSE loss achieves the best performance and β should be set smaller to stabilize the training for L_3.

L function	Acc_n	Acc_r
MSE(L1)	87.53%	57.35%
MSE+CE(L2)	88.35%	55.70%
KD(L3)	83.56%	47.18%
$D_{KL} + CE$(L4)	88.41%	54.78%

Table 5: Performance of AODT methods with different L functions on CIFAR-10.

5. Conclusion

In this paper, we propose the Adversarial Distillation Training (ADT) method for improving models' robustness while keeping a high accuracy on natural data. Previous methods usually enhance models' robustness at the cost of natural accuracy degradation. By distilling from a well trained clean model $\mathcal{M}^{natural}$, our proposed ADT method boosts performance on natural data of the trained model $\mathcal{M}^{robust}$ and achieve strong robustness. Moreover, the generalized AODT method further strengthens the robustness by joint training $\mathcal{M}^{natural}$ and $\mathcal{M}^{robust}$ together. Finally, extensive experiments on CIFAR-10 and CIFAR-100 show the flexibility and effectiveness of our methods.

References

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

[2] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *SP*, 2017. 3

[3] Elliot J. Crowley, Gavin Gray, and Amos J. Storkey. Moonshine: Distilling with cheap convolutions. In *NIPS*, 2018. 8

[4] 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

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

[6] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 3

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

[8] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. 2, 3

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

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

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

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

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

[14] 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

[15] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *CVPR*, 2016. 3

[16] 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 Symposium on Security and Privacy, SP*, 2016. 1

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

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

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

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

[21] 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

[22] 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

[23] 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, 2, 3

[24] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *ICCV*, October 2019. 3, 8

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

[26] 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

[27] 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

[28] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *NDSS*, 2018. 1, 6

[29] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 6, 7

[30] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017. 8

[31] 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, 5, 7, 8

[32] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *CVPR*, 2018. 4

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