Revisiting Adversarial Robustness Distillation: Robust Soft Labels Make Student Better

Bojia Zi[1,2]*, Shihao Zhao[1,2]*, Xingjun Ma[3]†, Yu-Gang Jiang[1,2]†
[1]Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan Univeristy
[2]Shanghai Collaborative Innovation Center on Intelligent Visual Computing
[3]School of Information Technology, Deakin University, Geelong, Australia

Abstract

Adversarial training is one effective approach for training robust deep neural networks against adversarial attacks. While being able to bring reliable robustness, adversarial training (AT) methods in general favor high capacity models, i.e., the larger the model the better the robustness. This tends to limit their effectiveness on small models, which are more preferable in scenarios where storage or computing resources are very limited (e.g., mobile devices). In this paper, we leverage the concept of knowledge distillation to improve the robustness of small models by distilling from adversarially trained large models. We first revisit several state-of-the-art AT methods from a distillation perspective and identify one common technique that can lead to improved robustness: the use of robust soft labels – predictions of a robust model. Following this observation, we propose a novel adversarial robustness distillation method called Robust Soft Label Adversarial Distillation (RSLAD) to train robust small student models. RSLAD fully exploits the robust soft labels produced by a robust (adversarially-trained) large teacher model to guide the student's learning on both natural and adversarial examples in all loss terms. We empirically demonstrate the effectiveness of our RSLAD approach over existing adversarial training and distillation methods in improving the robustness of small models against state-of-the-art attacks including the AutoAttack. We also provide a set of understandings on our RSLAD and the importance of robust soft labels for adversarial robustness distillation. Code: https://github.com/zibojia/RSLAD.

1. Introduction

Deep Neural Networks (DNNs) have become the standard models for solving complex real-world learning problems, such as image classification [25, 19], speech recognition [46] and natural language processing [45]. However, studies have shown that DNNs are vulnerable to adversarial attacks [43, 15], where imperceptible adversarial perturbations on the input can easily subvert the model's prediction. This raises security concerns on the deployment of DNNs in safety-critical scenarios such as autonomous driving [13, 7, 11] and medical diagnosis [31].

Different types of methods have been proposed to defend DNNs against adversarial attacks [22, 30, 32, 26, 57, 48], amongst which adversarial training (AT) has been found to be the most effective approach [2, 10]. AT can be regarded as a type of data augmentation technique that crafts adversarial versions of the natural examples for model training. AT is normally formulated as a min-max optimization problem with the inner maximization generates adversarial examples while the outer minimization optimizes the model's parameters on the adversarial examples generated during the inner maximization [32, 57, 47].

While being able to bring reliable robustness, AT methods have several drawbacks that may limit their effectiveness in certain application scenarios. Arguably, the most notable drawback is its hunger for high capacity models, i.e., the larger the model the better the robustness [49, 44, 35, 16]. However, there are scenarios where small and lightweight models are more preferable than large models. One example is the deployment of small DNNs in devices with limited memory and computational power such as smart phones and autonomous vehicles [37]. This has motivated the use of knowledge distillation along with AT to boost the robustness of small DNNs by distilling from robust large models [14, 3, 8, 62], a process known as Adversarial Robustness Distillation (ARD).

In this paper, we build upon previous works in both AT and ARD, and investigate the key element that can boost the robustness of small DNNs via distillation. We compare the loss functions adopted by several state-of-the-art AT methods and identify one common technique behind the improved robustness: the use of predictions of an adversarially trained model. We denote this type of supervision as

*Equal contribution: Bojia Zi(bjzi19@fudan.edu.cn) and Shihao Zhao(shzhao19@fudan.edu.cn)

†Correspondence to Xingjun Ma (daniel.ma@deakin.edu.au) and Yu-Gang Jiang (ygj@fudan.edu.cn)

Robust Soft Labels (RSLs). Compared to the original hard labels, RSLs can better represent the robust behaviors of the teacher model, providing more robust information to guide the student's learning. This observation motivates us to design a new ARD method to fully exploit the power of RSLs in boosting the robustness of small student models.

In summary, our main contributions are:

- We identify that the implicit distillation process existing in adversarial training methods is a useful function for promoting robustness and the use of *robust soft labels* can lead to improved robustness.
- We propose a novel adversarial robustness distillation method called Robust Soft Label Adversarial Distillation (RSLAD), which applies *robust soft labels* to replace hard labels in all of its supervision loss terms.
- We empirically verify the effectiveness of RSLAD in improving the robustness of small DNNs against state-of-the-art attacks. We also provide a comprehensive understanding of our RSLAD and the importance of robust soft labels for robustness distillation.

2. Related Work

2.1. Adversarial Attack

Given a DNN model with known parameters, adversarial examples (or attacks) can be crafted by Fast Gradient Sign Method (FGSM) [15], Projected Gradient Descent (PGD) [32], Carlini and Wagner (CW) attack [5] and a number of other methods. Several recent attacks were developed to produce more reliable adversarial robustness evaluation of defense models. These methods were designed to effectively avoid subtle gradient masking or obfuscating effects in improperly defended models. The AutoAttack (AA) [10] is an ensemble of four attacking methods including Auto-PGD (APGD), Difference of Logits Ratio (DLR) attack, FAB-Attack [9] and the black-box Square Attack [1]. The AA ensemble is arguably the most powerful attack to date.

2.2. Adversarial Training

Adversarial training is known as the most effective approach to defend adversarial examples. Recently, a number of understandings [30, 21, 12, 59, 61] and methods [32, 57, 48, 52, 51, 34, 17, 18, 4] have been put forward in this area. Adversarial training can be formulated as the following min-max optimization problem:

$$\begin{aligned} &\underbrace{\arg\min_{\theta} \mathcal{L}_{\text{min}}(f(x',\theta),y)}_{\text{Outer minimization}} \\ \text{where}\quad & x' = \underbrace{\mathop{\arg\max}_{\|x'-x\|_p\le\epsilon} \mathcal{L}_{\text{max}}(f(x',\theta),y)}_{\text{Inner maximization}} \end{aligned} \tag{1}$$

where f is a DNN model with parameters θ, x' is the adversarial example of natural example x within bounded L_p distance ϵ, $\mathcal{L}_{\text{min}}$ is the loss for the outer minimization, $\mathcal{L}_{\text{max}}$ is the loss for the inner maximization. The most commonly adopted L_p norm is the L_∞ norm. In Standard Adversarial Training (SAT) [32], the two losses $\mathcal{L}_{\text{min}}$ and $\mathcal{L}_{\text{max}}$ are set to the same loss, i.e., the most commonly used Cross Entropy (CE) loss. And the inner maximization problem is solved by the PGD attack. For simplicity, we omit the θ from the loss functions for the rest of this paper.

A body of work has been proposed to further improve the effectiveness of SAT. This includes the use of wider and larger models [49], additional unlabeled data [6], domain adaptation (natural domain versus adversarial domain) [40], theoretically-principled trade-off between robustness and accuracy (known as TRADES) via the use of Kullback–Leibler (KL) divergence loss for $\mathcal{L}_{\text{max}}$, the emphasis of misclassified examples via Misclassification-Aware adveRsarial Training (MART) [48], channel-wise activation suppressing (CAS) [4] and adversarial weight perturbation [50]. In general, elements that have been found in these works that can contribute to robustness include large models, more data, and the use of KL loss for the inner maximization.

AT methods are not perfect. One notable drawback of existing AT methods is that the smaller the model the poorer the robust performance [16]. It is generally hard to improve the robustness of small models like ResNet-18 [19] and MobileNetV2 [37], though many of the above AT methods can bring considerable robustness improvements to large models such as WideResNet-34-10 [57, 48] and WideResNet-70-16 [16]. This tends to limit their effectiveness in scenarios where storage or computational resources are limited, such as mobile devices, autonomous vehicles and drones. In this paper, we leverage knowledge distillation techniques to improve the robustness of small models and improve existing adversarial robustness distillation methods.

2.3. Knowledge Distillation

Knowledge distillation (KD) is one well-known method for deep neural network compression that distills the knowledge of a large DNN into a small, lightweight student DNN [20]. Given a well-trained teacher network T, KD trains the student network S by solving the following optimization problem:

$$\mathop{\text{argmin}}_{\theta_S}(1-\alpha)\mathcal{L}(S(x),y) + \alpha\tau^2\text{KL}(S^\tau(x), T^\tau(x)), \tag{2}$$

where KL is the Kullback-Leibler divergence, τ is a temperature constant added to the softmax operation, $\mathcal{L}$ is the classification loss of the student network with CE is a common choice. KD has been extended in different ways [36, 55, 27, 58] to a variety of learning tasks, such as noisy label learning [53, 60], AI security [14, 3, 28] and natural language processing [33, 41, 29]. Notably, a branch called self-distillation has attracted considerable attention in recent years [23, 58, 54]. Unlike traditional KD methods, self-distillation teaches a student network by itself rather than a separate teacher network.

KD has been applied along with adversarial training to boost the robustness of a student network with an adversarially pre-trained teacher network. The teacher can be a larger model with better robustness[14] (e.g. ARD) or share the same architecture with the student [62] (e.g. IAD). It has been shown that ARD and IAD can produce student networks that are more robust than trained from scratch, indicating that robust features learned by the teacher network can also be distilled [3]. In this paper, we will build upon these works and propose a more effective adversarial robustness distillation method to improve the robustness of small student networks.

3. Proposed Distillation Method

In this section, we revisit state-of-the-art AT and adversarial robustness distillation methods from the perspective of KD, and identify the importance of using robust soft labels for improving robustness. We then introduce our RSLAD method inspired by robust soft labels.

3.1. A Distillation View of Adversarial Training

Following the adversarial training framework defined in equation (1), we summarize, in Table 1, the loss functions and the student and teacher networks used in 4 state-of-the-art AT methods (i.e., SAT [32], TRADES [57] and MART [48]) and two adversarial robustness distillation methods (i.e., ARD [14] and IAD [62]). Compared to SAT which simply adopts the original hard label to supervise the learning, TRADES utilizes the natural predictions of the model via the KL term and gains significant robustness improvement [57]. From this perspective, TRADES is a self-distillation process where the teacher network is the student itself. MART [48] is also a self-distillation process but with a focus on the low probability examples via the (1-$f_y(x)$) weighting scheme on the KL term. In ARD, a more powerful teacher instead of the student itself is used to supervise the learning. The robustness is constantly improved from SAT's no distillation, TRADES/MART's self-distillation to ARD's full distillation [14], as we will also show in Section 4. IAD [62] is also an adversarial distillation method, which makes the distillation process more reliable by using the knowledge of both the teacher and the student networks. In this view, we believe that knowledge distillation implicitly or explicitly adopted in these methods contributes significantly to their success.

Another key difference between SAT and other methods mentioned above is that the latter exploit the teacher network's natural predictions in both of their outer and inner optimization processes, via the KL term. The predictions of a robust teacher model can be considered as a type of *Robust Soft Labels* (RSLs). Previous works (and also our experiments in Section 4) have shown that TRADES and its variants can bring considerable robustness improvement to SAT. From a distillation point of view, this robustness improvement comes from the use of RSLs, contrasting the use of original hard labels y. On the other hand, adversarial robustness distillation is to make the student as similar to the robust teacher as possible. Compared to the original hard labels, RSLs define the full robust behavior of the teacher network, thus convey more robust knowledge learned by the teacher to the student. In Section 4, we will empirically show that RSLs are indeed more beneficial to robustness than the original hard labels or other forms of non-robust soft labels. ARD has a KL term in its outer minimization loss, however, its other loss terms use the original hard labels y. IAD uses the KL terms in its two outer minimization loss terms, but the inner maximization loss still uses the hard labels, leaving space for improvement.

3.2. Robust Soft Label Adversarial Distillation

The proposed Robust Soft Label Adversarial Distillation (RSLAD) framework is illustrated in Figure 1, including a comparison with four existing methods (i.e., TRADES, MART, ARD and IAD). The key difference of our RSLAD to existing methods lies in the use of RSLs produced by the large teacher network to supervise the student's training on both natural and adversarial examples in all loss terms. The original hard labels y are absent in our RSLAD.

As the student network in RSLAD is still trained using AT, it also has the inner maximization and the outer minimization processes. To bring RSLs into its full play, we apply RSLs in both of the two processes. The loss functions used by our RSLAD are summarized in the last row of Table 1. Note that, in our RSLAD, the temperature constant commonly exists in distillation methods is fixed to $\tau = 1$ as we find it is no longer necessary when RSLs are used. Same as TRADES, MART, ARD and IAD, we use the natural RSLs (i.e. the predictions of a robust model for natural examples) as the soft label to supervise the model training.

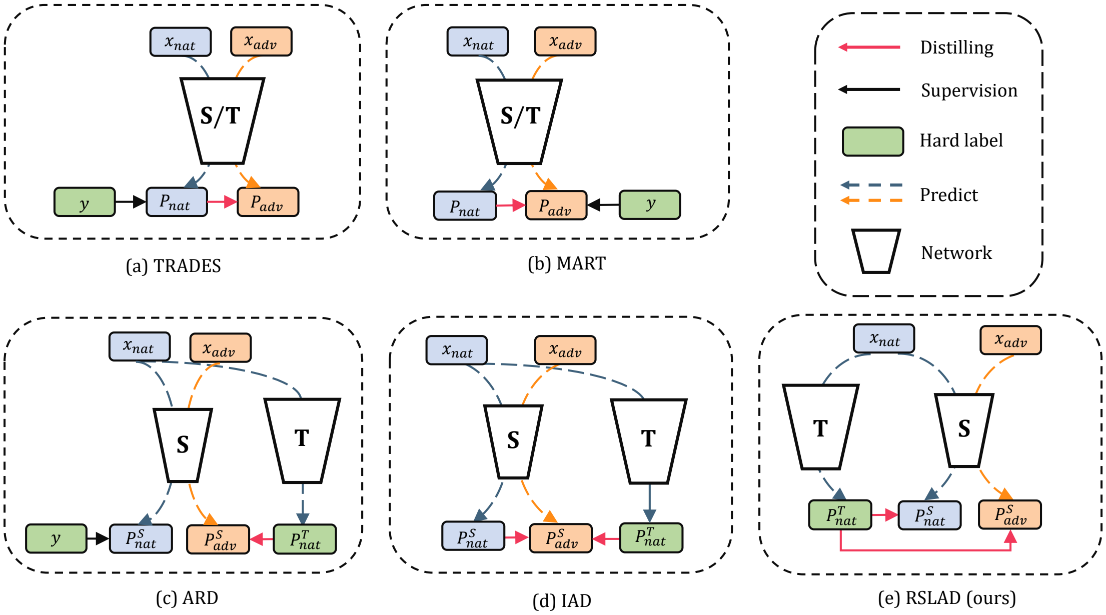
The overall optimization framework of our RSLAD is defined as following:

$$\begin{gathered}\arg\min_{\theta_S}(1-\alpha)\mathrm{KL}(S(x), T(x)) + \alpha\mathrm{KL}(S(x'), T(x)) \\ \text{where} \quad x' = \mathop{\arg\max}_{\|x'-x\|_p \leq \epsilon} \mathrm{KL}(S(x), T(x))\end{gathered} \tag{3}$$

where $S(x)$ and $T(x)$ are the abbreviations for $S(x, \theta_S)$ and $T(x, \theta_T)$, respectively. Since the RSLs produced by the adversarially trained teacher network $T(x)$ are also used to supervise the clean training part of the student's outer minimization, here we replace the commonly used CE loss by

Table 1: A unified view of 6 defense methods from the perspective of knowledge distillation. $\mathcal{L}_{\text{min}}$ is the loss function for the outer minimization while $\mathcal{L}_{\text{max}}$ is the loss function for the inner maximization. S and T represent the student and the teacher network respectively. λ in TRADES, MART and α in ARD, RSLAD are parameters balancing the two loss terms in $\mathcal{L}_{\text{min}}$. τ is a temperature constant added to the softmax operation. β in IAD is a hyper-parameter to sharpen the prediction.

Method	$\mathcal{L}_{\text{min}}$	$\mathcal{L}_{\text{max}}$	Student/Teacher
SAT	$\text{CE}(f(x'), y)$	$\text{CE}(f(x'), y)$	-
TRADES	$\text{CE}(f(x), y) + \lambda\text{KL}(f(x'), f(x))$	$\text{KL}(f(x'), f(x))$	S: $f(\cdot)$; T: $f(\cdot)$
MART	$\text{BCE}(f(x'), y) + \lambda(1 - f_y(x))\text{KL}(f(x'), f(x))$	$\text{CE}(f(x'), y)$	S: $f(\cdot)$; T: $f(\cdot)$
ARD	$(1-\alpha)\text{CE}(S^{\tau}(x), y) + \alpha\tau^2\text{KL}(S^{\tau}(x'), T^{\tau}(x))$	$\text{CE}(S(x'), y)$	S: $S(\cdot)$; T: $T(\cdot)$
IAD	$T_y(x')^{\beta}\text{KL}(S^{\tau}(x'), T^{\tau}(x)) + (1 - T_y(x')^{\beta})\text{KL}(S^{\tau}(x'), S^{\tau}(x))$	$\text{CE}(S(x'), y)$	S: $S(\cdot)$; T: $T(\cdot)$
RSLAD (ours)	$(1-\alpha)\text{KL}(S(x), T(x)) + \alpha\text{KL}(S(x'), T(x))$	$\text{KL}(S(x'), T(x))$	S: $S(\cdot)$; T: $T(\cdot)$



(a) TRADES

(b) MART

(c) ARD

(d) IAD

(e) RSLAD (ours)

Figure 1: An overview of the proposed **RSLAD** framework, in comparison with 4 existing methods including **TRADES**, **MART**, **ARD** and **IAD**. Black solid arrows represent training with hard labels y; yellow and blue dashed arrows represent predicting process for natural and adversarial examples respectively; red solid arrows represent distillation using *robust soft labels*. **S** and **T** represent the student and the teacher network respectively. P_{nat} and P_{adv} are the predictions of the model for the natural examples x_{nat} and adversarial examples x_{adv}. Note that no hard labels y are used in our RSLAD.

KL divergence to formulate the degree of distributional difference between the two models' output probabilities.

The goal of RSLAD is to learn a small student network that is as robust as an adversarially pre-trained teacher network, which is also to retain as much as possible the teacher's knowledge and robustness. We note that the commonly used hard labels in adversarial training can lose information learned by the teacher network to some extent, due to the fact that binarizing the teacher's output probabilities into hard labels tends to lose its true distribution. However, not all soft labels are robust. We will empirically show that smooth labels produced by label smoothing or soft labels produced by naturally trained non-robust models cannot improve robustness.

4. Experiments

We first describe the experimental setting, then evaluate the white-box robustness of 4 baseline defense methods and our RSLAD. We also conduct an ablation study, visualize the attention map learned by different methods, compare 3 types of soft labels, and explore how to choose a better teacher network.

4.1. Experimental Settings

We conduct our experiments on two benchmark datasets including CIFAR-10 and CIFAR-100 [24], and consider 5

Table 2: Robustness of the teacher networks used in our experiments.

Dataset	Teacher	Clean	FGSM	PGD_{SAT}	$\text{PGD}_{\text{TRADES}}$	CW_{∞}	AA
CIFAR-10	WideResNet-34-10	84.92%	60.87%	55.33%	56.61%	53.98%	53.08%
CIFAR-100	WideResNet-34-10	57.16%	33.58%	30.61%	31.34%	27.74%	26.78%
CIFAR-100	WideResNet-70-16	60.86%	35.68%	33.56%	33.99%	42.15%	30.03%

Table 3: White-box robustness results on CIFAR-10 dataset. MN-V2 and RN-18 are abbreviations of MobileNetV2 and ResNet-18 respectively. The maximum adversarial perturbation is $\epsilon = 8/255$. The best results are **blodfaced**.

Model	Method	**Best Checkpoint**						**Last Checkpoint**					
		Clean	FGSM	PGD_{SAT}	$\text{PGD}_{\text{TRADES}}$	CW_{∞}	AA	Clean	FGSM	PGD_{SAT}	$\text{PGD}_{\text{TRADES}}$	CW_{∞}	AA
RN-18	Natural	**94.65%**	19.26%	0.0%	0.0%	0.0%	0.0%	**94.65%**	19.26%	0.0%	0.0%	0.0%	0.0%
	SAT	83.38%	56.41%	49.11%	51.11%	48.67%	45.83%	84.44%	55.37%	46.22%	48.72%	47.14%	43.64%
	TRADES	81.93%	57.49%	52.66%	53.68%	50.58%	49.23%	82.20%	57.86%	52.30%	53.66%	50.69%	49.27%
	ARD	83.93%	59.31%	52.05%	54.20%	51.22%	49.19%	84.23%	59.33%	51.52%	53.74%	51.24%	48.90%
	IAD	83.24%	58.60%	52.21%	54.18%	51.25%	49.10%	83.90%	58.95%	51.35%	53.15%	50.52%	48.48%
	RSLAD	83.38%	**60.01%**	**54.24%**	**55.94%**	**53.30%**	**51.49%**	83.33%	**59.90%**	**54.14%**	**55.61%**	**53.22%**	**51.32%**
MN-V2	Natural	**92.95%**	14.47%	0.0%	0.0%	0.0%	0.0%	**92.78%**	14.59%	0.0%	0.0%	0.0%	0.0%
	SAT	82.48%	56.44%	50.10%	51.74%	49.33%	46.32%	82.89%	56.43%	49.71%	51.48%	49.07%	45.92%
	TRADES	80.57%	56.05%	51.06%	52.36%	49.36%	47.17%	80.57%	56.05%	51.06%	52.36%	49.36%	47.17%
	ARD	83.20%	58.06%	50.86%	52.87%	50.39%	48.34%	83.42%	57.94%	50.63%	52.44%	50.09%	48.01%
	IAD	81.91%	57.00%	51.88%	53.23%	50.45%	48.40%	83.49%	57.44%	49.77%	51.85%	49.41%	46.98%
	RSLAD	83.40%	**59.06%**	**53.16%**	**54.78%**	**51.91%**	**50.17%**	83.11%	**59.08%**	**53.04%**	**54.50%**	**51.60%**	**49.90%**

baseline methods: SAT [32], TRADES [57], ARD [14], IAD [62] and natural training.

Student and Teacher Networks. We consider two student networks including ResNet-18 [19] and MobileNetV2 [37], and two teacher networks including WideResNet-34-10 [56] for CIFAR-10 and WideResNet-70-16 [16] for CIFAR-100. The CIFAR-10 teacher WideResNet-34-10 is trained using TRADES, while for CIFAR-100, we use the WideResNet-70-16 model provided by Gowal et al. [16].

Training Setting. We train the networks using Stochastic Gradient Descent (SGD) optimizer with initial learning rate 0.1, momentum 0.9 and weight decay 2e-4. We set batch size to 128. For our RSLAD, we set the total number of training epochs to 300, and the learning rate is divided by 10 at the 215th, 260th and 285th epoch. A 10 step PGD (PGD-10) with random start size 0.001, step size 2/255 is used to solve the inner maximization of our RSLAD. For baseline methods SAT, TRADES and ARD, we strictly follow their original settings. IAD uses the same structure for the teacher and student networks. Here, we reproduce their method by using a more powerful teacher to fit our settings. Training perturbation is bounded to the L_{∞} norm $\epsilon = 8/255$ for both datasets. For natural training, we train the networks for 100 epochs on clean images with standard data augmentations and the learning rate is divided by 10 at the 75th and 90th epochs.

Evaluation Attacks. After training, we evaluate the model against 5 adversarial attacks: FGSM, PGD_{SAT}, $\text{PGD}_{\text{TRADES}}$, CW_{∞}(optimized by PGD) and AutoAttack(AA). The PGD_{SAT} attack is the original attack proposed in Madry et al. [32], while $\text{PGD}_{\text{TRADES}}$ is the one used in Zhang et al. [57]. They both are PGD attacks but differ in their hyper-parameters (e.g. step size). We consider these two attacks separately following Carmon et al. [6]. Note these attacks are commonly used adversarial attacks in adversarial robustness evaluation. Maximum perturbation used for evaluation is also set to $\epsilon = 8/255$ for both datasets. The perturbation steps of PGD_{SAT}, $\text{PGD}_{\text{TRADES}}$ and CW_{∞} are all 20. The robustness of the teacher models against the 5 attacks are reported in Table 2, indicating the maximum robustness the student model can get. Besides the white-box evaluation, we also conduct a black-box evaluation which will be described later.

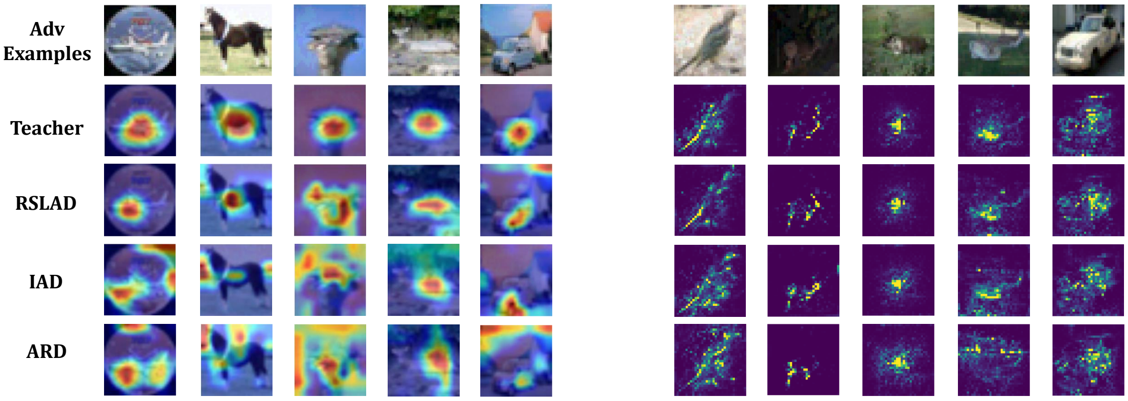
4.2. Adversarial Robustness Evaluation

White-box Robustness. The white-box robustness of our RSLAD and other baseline methods are reported in Table 3 for CIFAR-10 and Table 4 for CIFAR-100. Following previous works, we report the results at both the best checkpoint and the last checkpoint. The best checkpoint of naturally training (i.e., showing as 'Natural' in both Tables) is selected based on the performance on clean test examples, and the best checkpoints of SAT, TRADES, ARD, IAD, and our RSLAD are selected based on their robustness against the $\text{PGD}_{\text{TRADES}}$ attack.

As shown in Table 3 and Table 4, our RSLAD method demonstrates the state-of-the-art robustness on both CIFAR-10 and CIFAR-100 against all 5 attacks at either the best or the last checkpoints. For ResNet-18, RSLAD improves the robustness by 1.74% and 1.32% on

Table 4: White-box robustness results on CIFAR-100 dataset. MN-V2 and RN-18 are abbreviations of MobileNetV2 and ResNet-18 respectively. The maximum adversarial perturbation is $\epsilon = 8/255$. The best results are **blodfaced**.

Model	Method	**Best Checkpoint**						**Last Checkpoint**					
		Clean	FGSM	PGD_{SAT}	$\text{PGD}_{\text{TRADES}}$	CW_∞	AA	Clean	FGSM	PGD_{SAT}	$\text{PGD}_{\text{TRADES}}$	CW_∞	AA
RN-18	Natural	**75.55%**	9.48%	0.0%	0.0%	0.0%	0.0%	**75.39%**	9.57%	0.0%	0.0%	0.0%	0.0%
	SAT	57.46%	28.56%	24.07%	25.39%	23.68%	21.79%	57.51%	26.41%	21.7%	23.30%	22.15%	20.44%
	TRADES	55.23%	30.48%	27.79%	28.53%	25.06%	23.94%	54.62%	30.06%	27.35%	28.00%	24.34%	23.42%
	ARD	60.64%	33.41%	29.16%	30.30%	27.85%	25.65%	60.86%	32.64%	28.15%	29.34%	26.79%	24.74%
	IAD	57.66%	33.26%	29.59%	30.58%	27.37%	25.12%	58.82%	33.22%	28.50%	29.97%	26.79%	24.79%
	RSLAD	57.74%	**34.20%**	**31.08%**	**31.90%**	**28.34%**	**26.70%**	57.82%	**34.06%**	**30.68%**	**31.57%**	**28.16%**	**26.34%**
MN-V2	Natural	**74.58%**	7.19%	0.0%	0.0%	0.0%	0.0%	**74.58%**	7.19%	0.0%	0.0%	0.0%	0.0%
	SAT	56.85%	31.95%	28.33%	29.50%	26.85%	24.71%	58.50%	32.05%	27.80%	28.88%	26.74%	24.31%
	TRADES	56.20%	31.37%	29.21%	29.83%	25.06%	24.16%	56.56%	31.35%	28.85%	29.38%	25.00%	24.04%
	ARD	59.83%	33.05%	29.13%	30.26%	27.86%	25.53%	61.66%	32.98%	27.74%	29.33%	26.77%	24.34%
	IAD	56.14%	32.81%	29.81%	30.73%	27.99%	25.74%	58.07%	32.61%	27.55%	28.81%	26.24%	23.72%
	RSLAD	58.97%	**34.03%**	**30.40%**	**31.36%**	**28.22%**	**26.12%**	58.76%	**34.02%**	**30.17%**	**31.14%**	**28.10%**	**26.31%**



(a) Heatmap

(b) Saliency Map

Figure 2: Attention and saliency maps on adversarial examples. **Teacher:** WideResNet-34-10 trained by TRADES; **ARD:** the ResNet-18 student trained using ARD with the **Teacher** network; **RSLAD:** the ResNet-18 student trained using our RSLAD with the **Teacher** network. Heatmaps are generated by Grad-Cam[38] while saliency maps are generated by [39].

Table 5: Black-box robustness results on CIFAR-10 dataset. The maximum adversarial perturbation is $\epsilon = 8/255$. The best results are **blodfaced**.

Method	**ResNet-18**			**MobileNetV2**		
	PGD-20	CW_∞	Square	PGD-20	CW_∞	Square
SAT	60.84%	60.52%	54.27%	60.46%	59.83%	53.94%
TRADES	62.20%	61.75%	55.13%	60.90%	60.23%	53.46%
ARD	63.49%	63.05%	56.89%	62.13%	61.85%	55.60%
IAD	62.78%	62.26%	56.62%	61.57%	61.25%	55.45%
RSLAD	**64.11%**	**63.84%**	**57.90%**	**63.30%**	**63.20%**	**56.70%**

CIFAR-10 and CIFAR-100 respectively, compared to previous SOTA under $\text{PGD}_{\text{TRADES}}$ attack. For MobileNetV2, RSLAD brings 1.55% and 0.63% improvements against the $\text{PGD}_{\text{TRADES}}$ attack. The improvements are more pronounced against the AutoAttack, which is the most powerful attack to date. Particularly, our RSLAD outperforms ARD by even 2.30% for the ResNet-18 student on CIFAR-10. This verifies that our RSLAD is more stable and robust in training robust small DNNs than all the baseline methods. We also observe that, under all settings, TRADES holds clear advantage over SAT, but can still be largely outperformed by distillation methods (i.e., ARD and our RSLAD).

Black-box Robustness. Here, we evaluate the black-box robustness of our RSLAD, SAT, TRADES, ARD and IAD. We test both the transfer attack and query-based attack. This experiment is conducted on CIFAR-10 dataset. For transfer attack, we craft the test adversarial examples using 20 step PGD (PGD-20) and CW_∞ on an adversarially pre-trained ResNet-50 surrogate model. The maximum perturbation is also set to $8/255$. For query-based attack, we use one strong and query-efficient attack, i.e., the Square attack, to attack the models. We evaluate both the transfer attack and query-based attacks on the best checkpoints of the two student models (i.e., ResNet-18 and MobileNetV2). The results are

Table 6: Ablation study with ResNet-18 student network distilled using variants of our RSLAD and ARD [14]. **ARD-300**: ARD training under our RSLAD setting (i.e. 300 epochs); **ARD$_{\mathbf{min}}$**: the outer maximization part of ARD; **ARD$_{\mathbf{max}}$**: the inner minimization part of ARD; **RSLAD$_{\mathbf{min}}$**: the outer minimization part of our RSLAD; **RSLAD$_{\mathbf{max}}$**: the inner maximization part of our RSLAD.

Distillation Method	Clean	FGSM	PGD$_{\text{SAT}}$	PGD$_{\text{TRADES}}$	CW$_{\infty}$	AA
ARD-300	84.40%	59.81%	52.36%	54.49%	51.58%	49.70%
ARD$_{\text{min}}$+RSLAD$_{\text{max}}$	**84.70%**	**60.77%**	52.99%	54.84%	52.09%	50.35%
RSLAD$_{\text{min}}$+ARD$_{\text{max}}$	84.44%	59.89%	53.10%	55.01%	52.15%	49.94%
RSLAD	83.38%	60.01%	**54.24%**	**55.94%**	**53.30%**	**51.49%**

presented in Table 5. As can be observed, our RSLAD surpasses all the 4 baseline methods against all 3 black-box attacks, demonstrating the superiority of our robust soft label distillation approach. The general trend across different types of defense methods is consistent with that in the white-box setting: for robustifying small DNNs, TRADES is better than SAT while distillation methods are better than TRADES.

4.3. A Comprehensive Understanding of RSLAD

Ablation of RSLAD. To better understand the impact of each component of our RSLAD to robustness, we conduct a set of ablation studies with the existing distillation method ARD on CIFAR-10 with the ResNet-18 student network (the teacher is the same WideResNet-34-10 network as used in the above experiments). We replace the inner maximization and outer minimization losses used by ARD by the ones used in our RSLAD, then test the robustness of the trained student network. We also run an experiment with ARD under our RSLAD setting for 300 epochs (it was 200 epochs in the original paper). The ablation results are reported in Table 6. Compared to ARD, there is a certain improvement when either the inner loss or the outer loss of our RSLAD is used. The best robustness is achieved when both losses in ARD are switched to our RSLAD losses. This confirms the importance of each component of RSLAD, and the robust soft labels used in these components. We also find that the outer maximization has more impact on the overall robustness than the inner minimization: replacing the inner part of ARD by RSLAD leads to a more robust student than the outer part. An additional comparison between RSLAD and the baselines trained for 300 epochs can be found in Appendix D.

Attention Maps Learned by RSLAD. Here we use attention maps and saliency maps to visually inspect the similarity of the knowledge learned by the student to that of the teacher network. Given the same adversarial examples, higher similarity indicates more successful distillation and better aligned robustness to the teacher model. We take the ResNet-18 student distilled from the WideResNet-34-10 teacher on CIFAR-10 dataset as an example, and visualize the attention maps (generated by Grad-CAM [38]) and saliency maps (generated by [39]) in Figure 2. As can be observed, the attention maps of the student trained using our RSLAD are noticeably more similar to that of the teacher's than baseline methods ARD and IAD. This indicates that the student trained by our RSLAD can indeed mimic the teacher better and has gained more robust knowledge from the teacher. A parameter analysis of our RSLAD can be found in the appendix.

4.4. Further Explorations

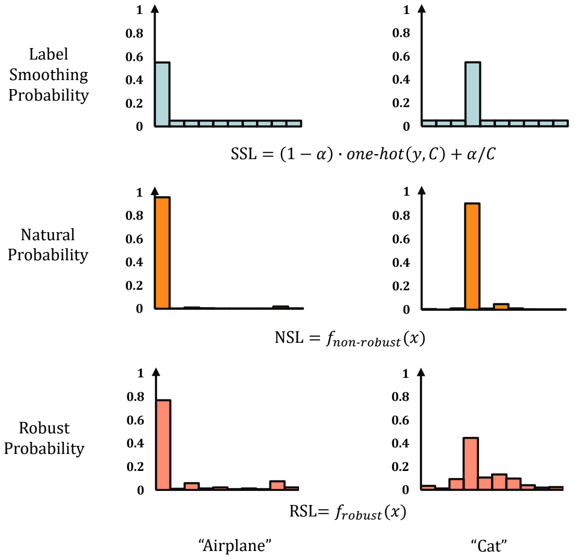


Figure 3: Probability distributions of 3 types of soft labels. f_{robust} represents the robust model, which is the adversarial trained model, $f_{non-robust}$ represents the non-robust model, which is the standard trained model. C represent the number of class in dataset, $one\text{-}hot(\cdot,\cdot)$ stands for the function that convert the label y to one-hot vector, α is a parameter which adjust the max number of the vector.

Different Types of Soft Labels. Here, we compare three types of soft labels: 1) smooth soft labels (SSLs) crafted by label smoothing [42]; 2) natural soft labels (NSLs) produced by a naturally trained teacher model; and 3) robust soft labels (RSLs) produced by an adversarially

Table 7: White-box robustness of ResNet-18 student trained using our RSLAD with three types of soft labels (i.e., SSL, NSL and RSL). The best results are **boldfaced**.

Soft Labels	Best Checkpoint			Last Checkpoint		
	Clean	$\text{PGD}_{\text{TRADES}}$	AA	Clean	$\text{PGD}_{\text{TRADES}}$	AA
SSL	**85.67%**	53.12%	47.88%	**85.26%**	49.70%	43.92%
NSL	85.02%	47.12%	42.87%	84.99%	46.69%	42.08%
RSL	83.38%	**55.94%**	**51.49%**	83.33%	**55.61%**	**51.32%**

trained robust teacher model. This experiment is conducted with ResNet-18 student and WideResNet-34-10 teacher on CIFAR-10 dataset, with our RSLAD. The probability distributions of the three types of soft labels for two example CIFAR-10 classes (i.e., 'Airplane' and 'Cat') are plotted in Figure 3. Different to RSLs, SSLs implement a fixed smoothing transformation to the original hard labels, while NSL probabilities are more concentrated around the ground truth label. The white-box robustness of the student network trained using our RSLAD with these 3 types of soft labels are shown in Table 7. One key observation is that the robustness drops drastically when non-robust labels including SSLs or NSLs are used in place of robust labels. This means that soft labels are not all beneficial to robustness, and non-robust labels especially the NSLs produced by non-robust models can significantly harm robustness distillation.

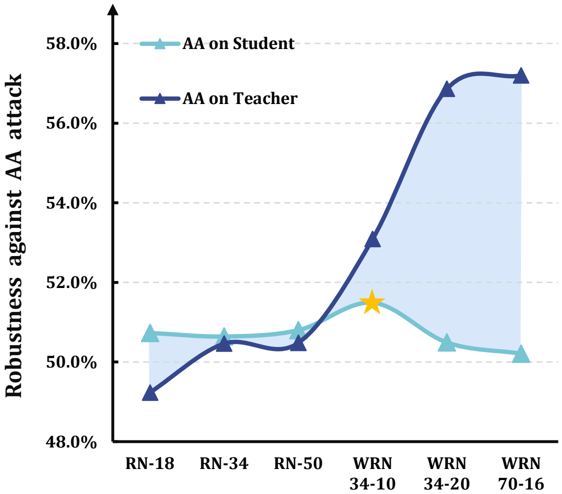


Figure 4: Robustness against AA attack of ResNet-18 (RN-18) students trained using our RSLAD with 6 different teachers. RN: ResNet; WRN: WideResNet. The RN-18, RN-34, RN-50, WRN-34-10 teachers are trained using TRADES, while the rest teacher models are from Gowal et al. [16]. This experiment is done on CIFAR-10 dataset.

How to Choose a Good Teacher? Here, we provide some empirical understandings on the impact of the teacher on the robustness of the student. We conduct this experiment on CIFAR-10 with the ResNet-18 student network and investigate its robustness when distilled using our RSLAD from 6 different teacher networks: ResNet-18, ResNet-34, ResNet-50, WideResNet-34-10, WideResNet-34-20 and WideResNet-70-16. The results are plotted in Figure 4. Surprisingly, we find that the student's robustness does not increase monotonically with that of the teacher's, instead, it first rises then drops. We call this phenomenon *robust saturation*. When the teacher network becomes too complex for the student to learn, the robustness of the student tends to drop. As shown in the figure, the robustness gap between the student and the teacher increases when the complexity of the teacher network goes beyond WideResNet-34-10. Interestingly, the student's robustness can surpass that of the teacher's when the teacher is smaller than WideResNet-34-10, especially when the teacher has the same architecture (i.e., ResNet-18) as the student. We call this phenomenon the *robust underfitting* of adversarial training methods, where robustness can be improved by training the model the second time while using the model trained the first time as a teacher. The robust underfitting region is where distillation can help boost the robustness. The best robustness of the ResNet-18 student is achieved when the WideResNet-34-10 (∼4.5× larger than ResNet-18) teacher is used. These results indicate that choosing a moderately large teacher model can lead to the maximum robustness gain in adversarial robustness distillation.

5. Conclusion

In this paper, we investigated the problem of training small robust models via knowledge distillation. We revisited several state-of-the-art adversarial training and robustness distillation methods from the perspective of distillation. By comparing their loss functions, we identified the importance of robust soft labels (RSLs) for improved robustness. Following this view, we proposed a novel adversarial robustness distillation method named Roust Soft Label Adversarial Distillation (RSLAD) to fully exploit the advantage of RSLs. The advantage of RSLAD over existing adversarial training and distillation methods were empirically verified on two benchmark datasets under both the white-box and the black-box settings. We also provided several insightful understandings of our RSLAD, different types of soft labels, and more importantly, the interplay between the teacher and student networks. Our work can help build adversarially robust lightweight deep learning models.

ACKNOWLEDGEMENT

This work was supported in part by National Natural Science Foundation of China (#62032006) and STCSM (#20511101000).

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. 2

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

[3] Tao Bai, Jinnan Chen, Jun Zhao, Bihan Wen, Xudong Jiang, and Alex Kot. Feature distillation with guided adversarial contrastive learning. *arXiv preprint arXiv:2009.09922*, 2020. 1, 3

[4] Yang Bai, Yuyuan Zeng, Yong Jiang, Shu-Tao Xia, Xingjun Ma, and Yisen Wang. Improving adversarial robustness via channel-wise activation suppressing. In *ICLR*, 2020. 2

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *S&P*, 2017. 2

[6] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *NeurIPS*, 2019. 2, 5

[7] Siheng Chen, Baoan Liu, Chen Feng, Carlos Vallespi-Gonzalez, and Carl Wellington. 3d point cloud processing and learning for autonomous driving: Impacting map creation, localization, and perception. *IEEE Signal Processing Magazine*, 38(1):68–86, 2020. 1

[8] Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *ICLR*, 2021. 1

[9] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*, 2020. 2

[10] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020. 1, 2

[11] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A Kai Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *CVPR*, 2020. 1

[12] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, D. Tsipras, Brandon Tran, and A. Madry. Adversarial robustness as a prior for learned representations. *arXiv: Machine Learning*, 2019. 2

[13] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *CVPR*, 2018. 1

[14] Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *AAAI*, 2020. 1, 3, 5, 7

[15] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2

[16] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020. 1, 2, 5, 8

[17] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020. 2

[18] Minghao Guo, Yuzhe Yang, Rui Xu, Ziwei Liu, and Dahua Lin. When nas meets robustness: In search of robust architectures against adversarial attacks. In *CVPR*, 2020. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 5

[20] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[21] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019. 2

[22] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. Comdefend: An efficient image compression model to defend adversarial examples. In *CVPR*, 2019. 1

[23] Kyungyul Kim, ByeongMoon Ji, Doyoung Yoon, and Sangheum Hwang. Self-knowledge distillation: A simple way for better generalization. *arXiv preprint arXiv:2006.12000*, 2020. 3

[24] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012. 4

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 2012. 1

[26] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *ICLR Workshop*, 2017. 1

[27] Xuewei Li, Songyuan Li, Bourahla Omar, Fei Wu, and Xi Li. Reskd: Residual-guided knowledge distillation. In *arXiv:2006.04719*, 2020. 2

[28] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *arXiv preprint arXiv:2101.05930*, 2021. 3

[29] Wenhao Lu, Jian Jiao, and Ruofei Zhang. Twinbert: Distilling knowledge to twin-structured bert models for efficient retrieval. *arXiv preprint arXiv:2002.06275*, 2020. 3

[30] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *ICLR*, 2018. 1, 2

[31] Xingjun Ma, Yuhao Niu, Lin Gu, Yisen Wang, Yitian Zhao, James Bailey, and Feng Lu. Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, 110:107332, 2021. 1

[32] 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, 5

[33] Ndapandula Nakashole and Raphael Flauger. Knowledge distillation for bilingual dictionary induction. In *EMNLP*, 2017. 3

[34] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. *NeurIPS*, 2019. 2

[35] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *ICLR*, 2020. 1

[36] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 2

[37] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 1, 2, 5

[38] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 6, 7

[39] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 6, 7

[40] Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Improving the generalization of adversarial training with domain adaptation. In *arXiv:1810.00740*, 2019. 2

[41] Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*, 2020. 3

[42] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 7

[43] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1

[44] Jonathan Uesato, Jean-Baptiste Alayrac, Po-Sen Huang, Robert Stanforth, Alhussein Fawzi, and Pushmeet Kohli. Are labels required for improving adversarial robustness? *arXiv preprint arXiv:1905.13725*, 2019. 1

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1

[46] Yisen Wang, Xuejiao Deng, Songbai Pu, and Zhiheng Huang. Residual convolutional ctc networks for automatic speech recognition. *arXiv preprint arXiv:1702.07793*, 2017. 1

[47] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, 2019. 1

[48] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020. 1, 2, 3

[49] Boxi Wu, Jinghui Chen, Deng Cai, Xiaofei He, and Quanquan Gu. Do wider neural networks really help adversarial robustness? *arXiv preprint arXiv:2010.01279*, 2020. 1, 2

[50] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, 2021. 2

[51] Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020. 2

[52] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019. 2

[53] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020. 3

[54] Ting-Bing Xu and Cheng-Lin Liu. Data-distortion guided self-distillation for deep neural networks. In *AAAI*, 2019. 3

[55] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 2

[56] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5

[57] Hongyang Zhang, Yaodong Yu, J. Jiao, E. Xing, L. Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019. 1, 2, 3, 5

[58] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *CVPR*, 2019. 2, 3

[59] Tianyuan Zhang and Zhanxing Zhu. Interpreting adversarially trained convolutional neural networks. In *ICML*, 2019. 2

[60] Zizhao Zhang, Han Zhang, Sercan O Arik, Honglak Lee, and Tomas Pfister. Distilling effective supervision from severe label noise. In *CVPR*, 2020. 3

[61] Shihao Zhao, Xingjun Ma, Yisen Wang, James Bailey, Bo Li, and Yu-Gang Jiang. What do deep nets learn? class-wise patterns revealed in the input space. *arXiv preprint arXiv:2101.06898*, 2021. 2

[62] Jianing Zhu, Jiangchao Yao, Bo Han, Jingfeng Zhang, Tongliang Liu, Gang Niu, Jingren Zhou, Jianliang Xu, and Hongxia Yang. Reliable adversarial distillation with unreliable teachers. *arXiv preprint arXiv:2106.04928*, 2021. 1, 3, 5