Boosting Accuracy and Robustness of Student Models via Adaptive Adversarial Distillation

Bo Huang[1,2], Mingyang Chen[1,2], Yi Wang[3], Junda Lu[4], Minhao Cheng[2], Wei Wang[1,2,*]
[1]The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China
[2]The Hong Kong University of Science and Technology, Hong Kong SAR, China
[3]Dongguan University of Technology, Dongguan, China
[4]Macquarie University, Sydney, Australia
{bhuangas, mchenbt}@connect.ust.hk; wangyi@dgut.edu.cn;
junda.lu@mq.edu.au;{minhaocheng, weiwcs}@ust.hk

Abstract

Distilled student models in teacher-student architectures are widely considered for computational-effective deployment in real-time applications and edge devices. However, there is a higher risk of student models to encounter adversarial attacks at the edge. Popular enhancing schemes such as adversarial training have limited performance on compressed networks. Thus, recent studies concern about adversarial distillation (AD) that aims to inherit not only prediction accuracy but also adversarial robustness of a robust teacher model under the paradigm of robust optimization. In the min-max framework of AD, existing AD methods generally use fixed supervision information from the teacher model to guide the inner optimization for knowledge distillation which often leads to an overcorrection towards model smoothness. In this paper, we propose an adaptive adversarial distillation (AdaAD) that involves the teacher model in the knowledge optimization process in a way interacting with the student model to adaptively search for the inner results. Comparing with state-of-the-art methods, the proposed AdaAD can significantly boost both the prediction accuracy and adversarial robustness of student models in most scenarios. In particular, the ResNet-18 model trained by AdaAD achieves top-rank performance (54.23% robust accuracy) on RobustBench under AutoAttack.

1. Introduction

Although demonstrating great success in dealing with large-scale data, deep neural networks (DNNs) are often over-parameterized in practice and require huge storage as well as computational cost [18, 22, 26]. In many real-time applications, it is desirable to deploy lightweight models in mobile devices with limited resources for prompt inference results. Teacher-student architectures have been considered as a means of computational-effective and high-performing deployment in such applications [23, 29, 45]. Due to limited budget when deploying at the edge, small (student) models are in general lack of sufficient protection mechanisms. Compared with large-scale models, however, they are more prone to the risk of being exposed to a potential attacker, e.g., who crafts adversarial attacks for malicious purpose [3, 21, 43]. Therefore, it is essential to improve adversarial robustness of small models against malicious attacks when applying them to real applications.

As a defense scheme, adversarial training (AT) has been studied and demonstrated effective in improving adversarial robustness for deep models [21, 24, 27, 32, 36]. Several studies have shown that AT is more effective on over-parameterized models with high capacity rather than on small models [27, 31, 48]. Recently, adversarial distillation (AD) was proposed as an alternative scheme for improving adversarial robustness in teacher-student architectures [20, 28, 49, 50]. Like AT, AD can also be formulated as a min-max optimization problem. It aims to enable the student model to inherit not only the prediction accuracy but also the adversarial robustness from a robust teacher model under the paradigm of robust optimization.

Existing AD methods generally utilize teacher models to produce fixed soft labels to guide the distillation optimization process [20, 49, 50]. However, fitting a neighborhood region with a fixed label will inevitably impose an overcorrection towards model smoothness, leading to a severe trade-off between accuracy and robustness [12, 12, 17]. Furthermore, these AD methods do not fully interact with the teacher models to minimize the prediction discrepancy between student and teacher models, thereby limiting the pre-

[*] Corresponding author.

diction and robustness inherited by the student model.

In this paper, we propose adaptive adversarial distillation (AdaAD) which fully involves a robust teacher model to adaptively search for more representative inner results in the knowledge distillation process. Specifically, in the inner optimization of AdaAD, we adaptively search for the points, representing the upper bound of the prediction discrepancy between the two models, as the inner results. And in outer optimization, we minimize the upper bound to perform distillation. In this way, we can enable the student model to better inherit the prediction accuracy and adversarial robustness from the teacher model.

Our main contributions can be summarized as:

- We formulate a new AD objective by maximizing the prediction discrepancy between teacher and student models in the min-max framework, and provide detailed analysis to explain why the proposed method can achieve better distillation performance.
- We design an adaptive adversarial distillation scheme, namely AdaAD, that adaptively searches for optimal *match points* in the inner optimization. This enables a much larger search radius (also known as perturbation limit) in local neighborhoods, which significantly enhances the robustness of student models.
- Extensive experimental results verify that the performance of our method is significantly superior to that of the state-of-the-arts AT and AD methods in various scenarios. In particular, the ResNet-18 model trained over CIFAR-10 dataset by AdaAD achieves top-rank performance (54.23% robust accuracy) on the leaderboard of RobustBench [1] under AutoAttack.

2. Related Work

2.1. Adversarial Attacks

Adversarial attacks are roughly categorized into 1) *white-box attacks*, and 2) *black-box attacks*. In white-box attacks, the adversary usually utilizes gradient information from target models to perform iteration optimization for crafting adversarial examples, like Fast Gradient Sign Method (FGSM) [21], Projected Gradient Descent (PGD) [27], Jacobian-based Saliency Map Attacks (JSMA) [33], Carlini-Wagner attacks (CW) [5], and AutoAttack (AA) [16]. Black-box attacks includes transfer-based attacks and query-based attacks. In transfer-based attacks, the adversary trains surrogate models locally to generate adversarial examples, which are then used to successfully attack target models. Query-based attacks, on the other hand, require querying the target models and searching for optimal directions across their discriminative boundary [1, 10, 11, 13, 15].

2.2. Adversarial Training

Adversarial training (AT) augmenting training samples with adversarial ones is one of the most effective and practical methods for training robust models, which cannot be completely defeated by powerful adapted attacks and thus has shown to be promising. [21] initially proposed to use FGSM adversarial examples for training. Then, [27] extended it by replacing FGSM samples with PGD ones, referred to as PGD-AT. Mathematically, PGD-AT can be formulated as a min-max framework to solve a robust optimization problem. An increasing number of approaches are proposed to improve AT, including introducing regularization terms [32,47], using additional data [6,35,39], and handling iteration process [24, 36].

2.3. Adversarial Distillation

AT is known as to require a large model capacity [38], making it challenging for small models with low capacity to attain satisfactory robustness. To mitigate this issue, Adversarial Distillation (AD), which aims at improving the robustness of small models (student models) by distilling knowledge from large, robust models (teacher models), has been extensively studied and shown to be promising. Unlike the conventional knowledge distillation [4, 23], AD emphasizes that student models are expected to inherit robustness from teacher models in addition to clean accuracy. Adversarial Robust Distillation (ARD) [20] was proposed to perform AD by involving the clean predictions of teacher models, which can be viewed as a natural extension of AT from the perspective of knowledge distillation. Then, [50] revisited ARD and proposed Robust Soft Label Adversarial Distillation (RSLAD) to use the robust soft labels in the inner optimization, demonstrating improved robustness. RSLAD highlights the importance of considering robust soft labels in the inner optimization of AD. [49] claimed that teacher models are not consistently reliable in each point and then proposed Introspective Adversarial Distillation (IAD) to conduct reliable AD. In addition, [28] proposed Adversarial Knowledge Distillation (AKD) to enhance AD by several strategies, such as early stopping, label mixing. The above mentioned AD methods perform alignment on the prediction outputs. On the other hand, some studies have explored alignment on feature layers [2, 8, 40, 44] or input gradients [7, 41], for robust student models generation.

3. Methodology

3.1. The Min-Max Framework of AD

We first revisit knowledge distillation in the ordinary setting. In general, knowledge distillation aims at distilling the knowledge of larger teacher models into small student models and is widely adopted in model compression. Assume that the input data point (x_i, y_i) obey the joint data distribu-

[1] https://robustbench.github.io

tion $p_d(x, y)$, the optimization objective of knowledge distillation can be formulated as

$$\mathbb{E}_{p_d(x)}[(1-\alpha)\ell(S(x), p_d(y|x)) + \alpha\tau^2\mathrm{KL}\left(S^\tau(x), T^\tau(x)\right)], \tag{1}$$

where $p_d(y|x)$ denotes the ground-true labels typically provided by the dataset, ℓ is Cross-Entropy loss (CE) widely used in supervision learning, τ is a temperature constant added in softmax transformation, KL is the Kullback-Leibler divergence, $T(\cdot)$ and $S(\cdot)$ denote the teacher and student model, parameterized by θ_T and θ_S, respectively. Given an input x, the output of the student model $S(x)$ is trained to match $T(x)$ generated by the teacher model. In the ordinary knowledge distillation, the student model is expected to inherit clean accuracy from the teacher model without consideration on adversarial robustness.

When considering adversarial robustness, the student model is expected to inherit not only accuracy on natural samples, but also robustness on adversarial ones from the teacher model. Along with this line of work, ARD [20] defined distillation objective from the adversarial perspective, which is formulated as

$$\mathbb{E}_{p_d(x)}[(1-\alpha)\ell(S(x), p_d(y|x)) + \alpha\tau^2\mathrm{KL}\left(S^\tau(x^*), T^\tau(x)\right)], \tag{2}$$

where x^* denotes the searching result of the inner optimization, which can be written as

$$x^* = x + \underset{\|\delta\|_p \le \epsilon}{\arg\max}\, \ell\left(S(x+\delta), p_d(y|x)\right), \tag{3}$$

where ϵ is the perturbation size under the L_p-norm constrain. Generally, ϵ can also be regarded as the search radius or the neighborhood region as well. To further improve robustness performance of student models, inherited from teacher models, RSLAD [50] was proposed to demonstrate that student models can obtain better robustness results by using robust soft-labels, i.e., the predictions of teacher models on clean samples, to guide the inner optimization. The inner optimization of RSLAD can be formulated as

$$x^* = x + \underset{\|\delta\|_p \le \epsilon}{\arg\max}\, \mathrm{KL}(S(x+\delta) \,||\, T(x)). \tag{4}$$

In addition, AKD [28] was proposed to enhance AD by label mixing, the objective of AKD is

$$\mathbb{E}_{p_d(x)}\mathrm{CE}(S(x^*), \beta T(x^*) + (1-\beta)p_d(y|x)), \tag{5}$$

where x^* is calculated by Eq. (3) and $\beta \in [0, 1]$ controls the mixing of the distilled labels and the ground-true ones.

3.2. Adaptive Adversarial Distillation (AdaAD)

The desire goal of AD is to enable student models to inherit as much of the prediction accuracy and adversarial robustness from teacher models by distillation training. Ideally, given an input x and its ϵ-neighborhood spaces, it is expected that the predictions of the student model on any point in this space can be maximally aligned with the teacher model. By considering the whole input distribution and the relevant space of tolerant perturbation, the distillation objective in AD could be formally defined as

$$\mathcal{L}_{\mathrm{AD}} = \iint \mathcal{D}(S(x+\delta), T(x+\delta))d\delta dx, \tag{6}$$

where $\mathcal{D}(\cdot)$ represents distance function and δ represents feasible points under the constrain $\|\delta\|_p \le \epsilon$. This objective term Eq. (6) encourages to achieve maximum point-to-point alignment between $S(\cdot)$ and $T(\cdot)$ on the full input distribution $p_d(x)$ along with the adversarial spaces. However, since the input distribution is generally mapped in a high-dimensional space , it is extraordinarily challenging to directly optimize $\mathcal{L}_{\mathrm{AD}}$ in practice.

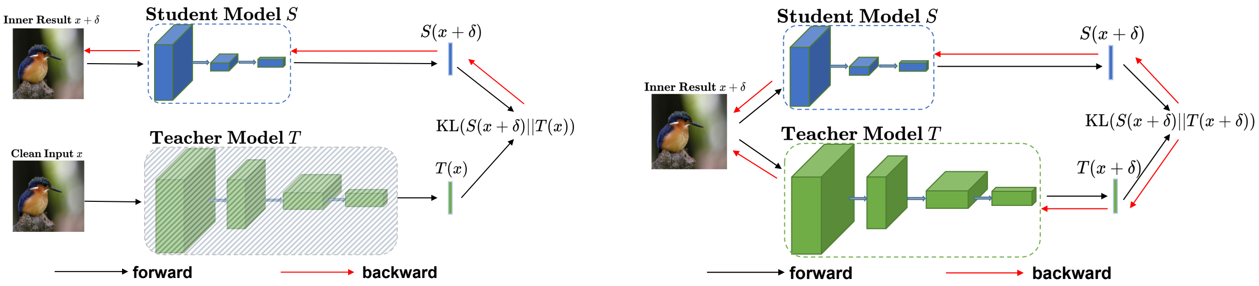
We note that the upper bound of $\mathcal{D}(S(x+\delta), T(x+\delta))$ for any point x and its ϵ-neighborhood spaces can be approximately tractable by leveraging gradient descent optimization. With this observation, it is feasible to convert Eq. (6) into a solvable objective $\mathcal{L}'_{\mathrm{AD}}$ to alternatively optimize $\mathcal{L}_{\mathrm{AD}}$, formulated as

$$\mathcal{L}'_{\mathrm{AD}} = \int \sup_{\|\delta\|_p \le \epsilon} \left(\mathcal{D}(S(x+\delta), T(x+\delta))\right)dx. \tag{7}$$

The optimization of Eq. (7) consists of two steps. The first step is to find the maximum distance value in the ϵ-neighborhood region of the given point x, and the next step is to compute the cumulative integral of the corresponding maximum distance over all sampled x. The two steps can be equivalent to a min-max optimization process, can be written as

$$\min \max_{\|\delta\|_p \le \epsilon} \mathcal{D}(S(x+\delta), T(x+\delta)). \tag{8}$$

Based on the above analysis, we propose to utilize a min-max framework to derive the suboptimal solution of minimizing $\mathcal{L}_{\mathrm{AD}}$, which can be viewed as an alternative of maximum point-to-point alignment between student and teacher models under the paradigm of robust training. Specifically, we first use gradient descent algorithm to adaptively search the upper bound of the prediction discrepancy between the student and teacher model in the inner optimization, in which the gradient of the prediction discrepancy between two models w.r.t the input is derived by involving both two models in the backpropagation. Then we minimize the upper bound in outer optimization to perform distillation. Since the inner optimization can adaptively search for the upper bound of discrepancy between two models' predictions on ϵ-neighborhood region, we therefore name the proposed method as adaptive adversarial distillation (AdaAD).



(a) RSLAD

(b) AdaAD

Figure 1. The comparison on the inner optimization of RSLAD and the proposed AdaAD

Mathematically, given an input x and tolerant perturbation size ϵ, the proposed inner optimization aims at searching for the "support" point x^* with maximum prediction discrepancy between student and teacher models in the neighborhood region of the data point x, formulated as

$$x^* = x + \arg\max_{||\delta||_p \leq \epsilon} \mathrm{KL}\left(S(x+\delta) \,||\, T(x+\delta)\right). \quad (9)$$

We adopt KL divergence as the distance function to measure the discrepancy between two models' output probabilities. In practice, we leverage projection gradient descent strategy [27] to search for "support" instance x^* for training. After the searching process of x^* has been done, the upper bound of the prediction discrepancy between the student and teacher model in the ϵ-neighborhood region of the data point x could be approximated as $\mathrm{KL}(S(x^*) \,||\, T(x^*))$. Then, the outer optimization is to minimize the approximated upper bound of the two models' output discrepancy to perform distillation, defined as

$$\arg\min_{\theta_S} \mathrm{KL}(S(x^*) \,||\, T(x^*)). \quad (10)$$

With considering introducing the distillation temperature τ and combining the inner and outer optimization, the proposed overall optimization could be formulated as

$$\arg\min_{\theta_S} (1-\alpha)\mathrm{KL}(S(x) \,||\, T(x)) + \alpha\mathrm{KL}(S^\tau(x^*) \,||\, T^\tau(x^*)), \quad (11)$$

where x^* is derived by Eq. (9), the clean output matching $\mathrm{KL}(S(x) \,||\, T(x))$ can also be considered in the outer optimization, and the hyper-parameter α is used to control the balance between the two matching part.

Considering that teacher models with high capacity may still be unable to make the accurate prediction on some input points and their neighborhood region, teacher models become unreliable. Thus, it is undesirable to conduct AD for those points on which teacher models make wrong predictions. [49] demonstrated that teacher models progressively become unreliable during AD training and accordingly proposed Introspective Adversarial Distillation (IAD) to encourage student models to partially instead of fully trust teacher models for AD. Because the objective of the proposed AdaAD is to maximally align with the teacher model and is a general method for AD, AdaAD can be naturally coupled with IAD to make the distillation process more reliable. The combination of AdaAD and IAD is referred to as AdaIAD.

3.3. Inner Optimization Difference

Fig. 1 illustrates the inner optimization of RSLAD and AdaAD. RSLAD differs from AdaAD in two key aspects. First, RSLAD employs fixed soft predictions as the supervision target for guiding the inner optimization. Second, RSLAD does not require backward propagation of teacher models during the inner optimization, whereas AdaAD does. The former imposes local invariance, leading to a significant adversarial trade-off between accuracy and robustness, which we will discuss in detail in Sec. 3.5. The latter means that RSLAD treats the teacher model as a black-box model, and does not utilize the gradient information from the teacher model to optimize the inner result x^* during the inner optimization. As a result, the predictions discrepancy between the two models on the inner results is suboptimal. Hence, the inner optimization of RSLAD can only be considered a rough estimate of the upper bound in Eq. (7), thereby limiting distilled performance by student models.

Note that AdaAD will be equivalent to RSLAD when the search radius $\epsilon = 0$. Nonetheless, setting $\epsilon = 0$ will make AD methods convert into conventional knowledge distillation defined as Eq. (1), which cannot ensure robust optimization. When ϵ is appropriately selected and larger, RSLAD and AdaAD are not equivalent because the soft predictions of the teacher model on the ϵ-neighborhood region are not constant due to the high-dimensional property [21].

3.4. Larger Search Radius ϵ

As aforementioned, PGD-AT, RSLAD, and other AD methods use fixed either hard labels or soft labels as the supervision information to guide the inner optimization, in which the discrepancy between the prediction of the inner result and the fixed supervision target is expected to be maximized. Hence, the inner results generated by them are more likely to be highly adversarial. Meanwhile, there is an inevitable demand of large model capacity for fitting highly adversarial inner results with the pre-given targets [38]. However, student models obviously do not meet the requirement. Intuitively, the larger the search radius ϵ in the inner optimization is, the stronger the adversarial nature of the produced inner results would be. Therefore, with large ϵ in the inner optimization, the performance of PGD-AT, RSLAD, and cited other AD methods will sharply drop.

In contrast, the inner optimization of AdaAD is to search for the upper bound of the prediction discrepancy between student and teacher models, implying the generated inner results are not necessarily adversarial. This inherent property allows AdaAD to significantly expand the searching region of the inner results, i.e., larger ϵ in the inner optimization. Increasing ϵ in the inner optimization means that the feasible searching region is expanded, which is beneficial for student models to inherit prediction accuracy and adversarial robustness from the robust teacher model. The experimental results can be referred to Sec. 4.2.

3.5. Reconciling Accuracy and Robustness

The robust error defined in the original formulation of PGD-AT is equivalent to

$$\mathbf{R}_{\text{Madry}}(\theta) = \mathbb{E}_{p_d(x)}[\max_{\|\delta\|_p \leq \epsilon} \text{KL}\left(p_d(y|x) \| f_\theta(x+\delta)\right)], \tag{12}$$

by replacing the Cross-Entropy loss with KL divergence [27, 30], where $p_d(y|x)$ denotes the ground-truth distribution. The desired goal of Eq. (12) is to minimize the difference between the distribution of predictions of adversarial examples and the ground-truth distribution. However, many empirical observations have shown that despite model robustness being enhanced, clean accuracy largely drops by optimizing Eq. (12). This adversarial trade-off has received extensive attention. In particular, [30] revealed that optimizing the robust error $\mathbf{R}_{\text{Madry}}(\theta)$ to find the optimal θ^* will impose an inductive bias towards local invariance: for $\forall\delta \in \{\delta \mid \|\delta\|_p \leq \epsilon\}$, $f_\theta(x+\delta)$ is encouraged to be equal to $p_d(y|x)$ that represents the hard labels in the dataset. Similarly, replacing $p_d(y|x)$ by fixed $T(x)$ in RSLAD still inevitably imposes the local invariance. As a result of the locally-invariant bias, the trained model has a propensity to be over-smoothed,i.e., an overcorrection towards model smoothness, as shown in the earlier publication [12, 17]. Consequently, when minimizing $\mathbf{R}_{\text{Madry}}$ w.r.t θ during AT, $f_{\theta^*}(x)$ generally does not converge to $p_d(y|x)$ as demonstrated in [30]. This inconsistency between $f_{\theta^*}(x)$ and $p_d(y|x)$ can explain why there exists a distinct trade-off between accuracy and robustness by AT.

The proposed AdaAD can address the inconsistency to some extent, which enables the trained model improve the adversarial trade-off. Specifically, different from PGD-AT, the robust error defined in AdaAD is

Table 1. The performance of teacher models for two datasets. WRN-34-10 and WRN-34-20 are abbreviations of teacher models WideResNet-34-10 and WideResNet-34-20, respectively.

Dataset	Teacher	Clean	PGD	CW	AA
CIFAR-10	WRN-34-10 [31]	87.20	55.90	77.80	51.79
CIFAR-10	WRN-34-20 [9]	86.03	63.33	82.60	57.71
CIFAR-100	WRN-34-10 [9]	64.07	36.61	56.22	30.57

$$\mathbf{R}_{\text{AdaAD}}(\theta_S) = \mathbb{E}_{p_d(x)}[\max_{\|\delta\|_p \leq \epsilon} \text{KL}\left(S(x+\delta) \| T(x+\delta)\right)]. \tag{13}$$

In AdaAD, the teacher model $T(\cdot)$ is a well-trained robust model and can be viewed as a well-estimated probability generator for each data point and its ϵ-neighborhood region. Hence, the teacher model $T(\cdot)$ can be served as a better substitute than $p_d(y|x)$. In this way, $p_d(y|(x+\delta))$ that is inaccessible and is not accurately labeled in the dataset can be approximated by $T(x+\delta)$ for any $\delta \in \{\delta \mid \|\delta\|_p \leq \epsilon\}$. The used teacher model $T(\cdot)$ generally has a non-negligible property of local variance: for $\forall\delta_1, \delta_2$, satisfying $\delta_1 \neq \delta_2$ and $\delta_1, \delta_2 \in \{\delta \mid \|\delta\|_p \leq \epsilon\}$, $T(x+\delta_1)$ is generally not exactly equal to $T(x+\delta_2)$. More importantly, even for data points of the same class in supervision classification, there are non-negligible numerical differences. Meanwhile, the inner results produced by AdaAD from different initial data points will essentially yield various convergence points. The alignment between $T(x+\delta^*)$ and $S(x+\delta^*)$ in the outer optimization in AdaAD allows $S(x+\delta)$ to point-wisely and maximally match $T(x+\delta)$ for any $\delta \in \{\delta \mid \|\delta\|_p \leq \epsilon\}$. These natures mean that local invariance can be largely eliminated during the training. Hence, AdaAD can generally reconcile accuracy and robustness.

4. Experimental Evaluations

Experimental Setup. We evaluate the effectiveness of AdaAD in two benchmark image datasets, namely CIFAR-10 and CIFAR-100 [25]. In both two datasets, the pixel range of images is normalized to be in the interval [0,1]. We compare AdaAD and AdaIAD with two commonly used AT methods (PGD-AT and TRADES) and some representative AD approaches, namely ARD, IAD, RSLAD, and AKD. Following the standard setting in AD [20, 49, 50], we also consider two widely used student models, including ResNet-18 [22] and MobileNetV2 [37], and teacher models including WideResNet-34-10 for both two datasets and WideResNet-34-20 for CIFAR-10 [9, 31]. The performance of teacher models is shown in Tab. 1

Implementation Details. We train the models using

Table 2. Model robustness by recognition accuracy (%) on various attacks over CIFAR-10 dataset. RN-18 and MN-V2 are abbreviations of student models ResNet-18 and MobileNetV2, respectively. The best results are **boldfaced**.

Teacher Model		WRN-34-20 [9]					WRN-34-10 [31]				
Model	Method	Clean	FGSM	PGD	CW_2	AA	Clean	FGSM	PGD	CW_2	AA
RN-18	PGD-AT	82.95	57.16	52.87	77.56	47.69	82.95	57.16	52.87	77.56	47.69
	TRADES	83.00	58.42	53.18	76.92	49.21	83.00	58.42	53.18	76.92	49.21
	ARD	84.03	58.16	53.11	79.13	48.07	84.04	58.26	52.67	74.95	48.62
	IAD	84.71	61.28	54.92	79.44	49.85	83.19	57.76	53.17	76.77	48.82
	RSLAD	83.52	58.36	53.46	78.36	48.66	83.60	57.45	52.60	76.85	48.45
	AKD	83.22	58.63	54.16	78.44	49.26	84.69	58.97	53.28	77.25	48.37
	AdaAD	**85.58**	60.85	56.40	80.83	51.37	86.75	60.37	54.13	78.18	50.06
	AdaIAD	85.04	**62.62**	**58.34**	**81.15**	**52.96**	**87.08**	**61.47**	**55.01**	**78.77**	**50.74**
MN-V2	PGD-AT	77.54	53.58	49.90	72.54	44.56	77.54	53.58	49.90	72.54	44.56
	TRADES	79.80	54.84	50.51	75.30	45.67	79.80	54.84	50.51	75.30	45.67
	ARD	79.56	53.17	49.06	74.51	44.04	84.63	58.00	50.82	72.93	46.48
	IAD	83.31	58.29	52.98	78.03	47.11	82.11	55.27	50.20	75.41	45.66
	RSLAD	81.11	56.39	51.66	76.20	46.75	83.24	56.69	51.57	76.52	47.18
	AKD	83.41	57.71	52.35	77.97	46.82	82.64	56.17	50.49	75.31	45.67
	AdaAD	83.79	57.29	53.04	79.24	47.66	**86.80**	**58.56**	**52.00**	**78.27**	**47.97**
	AdaIAD	**84.63**	**59.79**	**54.97**	**80.21**	**49.29**	85.69	56.55	50.11	77.55	46.03

Table 3. Model robustness by recognition accuracy (%) on various attacks over CIFAR-100 dataset.

Teacher Model		WRN-34-10 [9]				
	Method	Clean	FGSM	PGD	CW_2	AA
RN-18	PGD-AT	56.27	32.08	29.84	49.05	24.99
	TRADES	57.82	32.52	30.38	51.30	25.02
	ARD	60.94	35.31	32.72	53.67	26.04
	IAD	60.43	35.75	32.80	52.71	26.84
	RSLAD	59.55	35.68	33.35	52.89	27.77
	AKD	57.84	34.32	31.98	51.06	26.06
	AdaAD	62.19	35.33	32.52	54.67	26.74
	AdaIAD	**62.49**	**36.31**	**33.76**	**55.18**	**27.98**
MN-V2	PGD-AT	51.55	29.34	27.26	45.73	22.07
	TRADES	53.05	29.07	27.44	47.62	21.82
	ARD	57.18	33.13	30.91	51.50	24.20
	IAD	56.33	32.88	30.18	49.00	24.07
	RSLAD	56.04	32.76	30.29	50.14	24.56
	AKD	56.75	33.11	30.50	49.53	24.65
	AdaAD	**61.44**	34.75	31.97	54.21	25.91
	AdaIAD	61.24	**34.82**	**32.68**	**54.47**	**26.43**

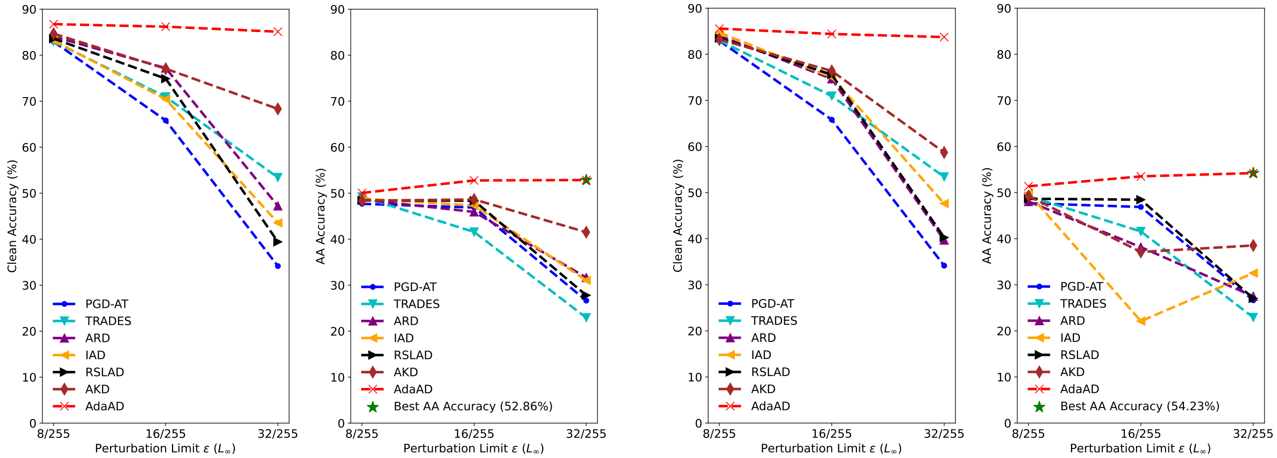
an SGD momentum optimizer with an initial learning rate 0.1, momentum 0.9, and weight decay 5e-4. For PGD-AT, we adopt 110 training epochs with early stopping strategy [31, 36], while for TRADES and other AD methods, we use 200 training epochs and the learning rate is divided by 10 at the 100th and 150th epochs. Unless otherwise specified, the number of iterations during the inner optimization is set to 10 with step size 2/255, and the total perturbation bound is 8/255 under L_∞ constrain. We set the hyper-parameter $\alpha = 1.0$ in ARD, IAD, RSLAD, and AdaAD as recommended in [20]. For each AD method, we use the recommended distillation temperature τ as reported in [20, 49, 50]. We adopt random cropping and flipping for data augmentation during the whole training process. Our implementation is based on Pytorch framework [34] and advertorch library [19]. Code is available at https://github.com/boyellow/AdaAD.

Evaluation Metrics. We use natural/clean accuracy on natural test samples and robust accuracy on adversarial test samples to demonstrate model performance. We consider 4 representative adversarial attacks including FGSM, PGD, CW_2 (constrained by l_2 norm), and AutoAttack (AA). For FGSM, PGD, and AA, the maximum perturbation size is set to 8/255, while PGD adopts 10 steps with step size 2/255. The balance constant in CW is set to 0.1. Unless otherwise specified, We report the results on the checkpoint with the best PGD-10 accuracy.

4.1. Model Adversarial Robustness

Tab. 2 and Tab. 3 report the recognition accuracy of the proposed AdaAD and AdaIAD models, as well as those of other state-of-the-art methods, under various adversarial attacks. Since PGD-AT and TRADES training is not facilitated by teacher models, the performance of models trained by PGD-AT and TRADES remains unchanged with different teacher models. From the two tables, we can observe three important findings.

First, AD methods are substantially ahead of AT ones for both clean accuracy and adversarial robustness, indicating AD is more effective and competitive than AT in improving the robustness of small-scaled models. Second, AdaAD and AdaIAD significantly outperform the cited state-of-the-art



(a) Using the teacher WideResNet-34-10 [31]

(b) Using the teacher WideResNet-34-20 [9]

Figure 2. Clean and AA accuracy (%) of the studnet model ResNet-18 trained with increasing ϵ in the inner optimization on CIFAR-10.

methods in most scenarios. In particular, compared to the best results achieved by the state-of-the-arts, for the robust accuracy on the most powerful attack AA (abbreviated by AA accuracy), AdaAD and AdaIAD achieve significant improvements of up to 1.52% and 3.11%, respectively, moreover, for the natural accuracy, AdaAD and AdaIAD achieve significant gains of up to 4.26% and 4.06%, respectively. It reveals AdaAD and AdaIAD can explicitly improve the adversarial trade-off. Third, the phenomenon of *robust saturation* is observed for ARD and RSLAD. That is, the robustness of distilled student models does not increase with the used teacher model becoming more robust. In contrast, AdaAD and AdaIAD bring greater improvement in robustness when using a more robust teacher model. It should also be noted that since AA includes two kinds of query-based attacks, the effectiveness of AdaAD and AdaIAD in improving AA accuracy demonstrates that AdaAD and AdaIAD are reliable in resisting query-based attacks.

4.2. Effects of Increasing The Search Radius ϵ

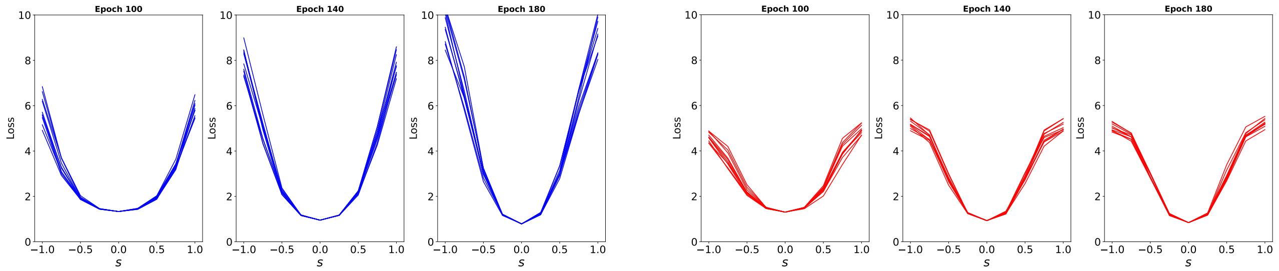
Fig. 2 evaluates the clean accuracy and AA accuracy for the student model ResNet-18 trained with increasing perturbation limit ϵ in inner optimization on CIFAR-10. Overall, the proposed AdaAD achieves significantly higher accuracy and robustness than other methods in all cases of ϵ. Specifically, when increasing ϵ from 8/255 to 32/255, the clean accuracy on AdaAD will keep almost unchanged, and the AA accuracy on AdaAD will consistently increase, while the clean accuracy on all the compared methods will drop sharply below 60% and their AA accuracy will decrease below 40%, with two different teacher models being used. In particular, the AA accuracy on the proposed AdaAD will increase from 50.06% to 52.86%, surprisingly surpassing that of the teacher WideResNet-34-10 [31] (51.79%). More importantly, as shown in Fig. 2b, even compared with the best robust accuracy (52.48%) for ResNet-18 exhibited on RobustBench [14] under AA attack in the scenario that no additional data augmentation is used, our proposed AdaAD method can achieve a 1.75% improvement (54.23%) when ϵ is set as 32/255 and a more robust model, i.e., WideResNet-34-20 [9], is employed as the teacher model. The results verify that the adaptive nature allows AdaAD to search for the upper bound of the prediction discrepancy between the student and teacher model in a larger local region, which benefits the student model to better inherit from the teacher model without clean accuracy degradation.

Table 4. Recognition accuracy (%) on transfer-based attacks for ResNet-18 models trained by various methods over CIFAR-10 dataset.

Surrogate	ResNet-34			VGG-16		
Method	FGSM	PGD	JSMA	FGSM	PGD	JSMA
PGD-AT	63.05	60.58	84.90	64.06	62.78	85.77
TRADES	65.57	63.93	84.71	66.88	66.00	85.36
ARD	65.26	63.20	86.06	66.64	65.43	87.03
IAD	67.49	65.57	86.72	69.09	68.27	87.42
RSLAD	65.06	62.77	85.43	65.91	64.83	86.26
AKD	64.34	62.23	85.22	65.24	64.30	86.24
AdaAD	66.81	64.57	**88.00**	68.74	67.89	**88.39**
AdaIAD	**67.66**	**65.81**	87.31	**69.44**	**68.62**	88.01

4.3. Evaluation on Transfer-based Attacks

We examine whether the proposed AdaAD and AdaIAD effectively resist transfer-based attacks. Specifically, we train two surrogate models with different architectures, ResNet-34 [22] and VGG-16 [42], using the PGD-AT method with the early stopping strategy. Then we gen-



(a) PGD-AT

(b) AdaAD

Figure 3. The visualization of weight loss landscape in the training process between 100 and 180 epochs for the ResNet-18 model along 10 different random directions, where Loss denotes the adversarial loss and s is the magnitude defined in [46].

Table 5. Recognition accuracy (%) evaluated on self adversarial distillation over CIFAR-10 dataset.

	Method	Clean	FGSM	PGD	CW_2	AA
RN-18	PGD-AT	82.95	57.16	52.87	77.56	47.69
	ARD	80.66	55.68	50.90	74.87	46.61
	IAD	81.32	57.54	52.91	75.69	48.20
	RSLAD	81.92	57.94	53.29	76.26	49.06
	AKD	**83.74**	**58.87**	54.17	**77.97**	48.84
	AdaAD	83.13	57.54	53.30	77.62	49.61
	AdaIAD	82.88	58.45	**54.29**	77.62	**50.19**

erate adversarial attacks on the two surrogate models to evaluate the effectiveness of the model ResNet-18 trained by the proposed AdaAD and AdaIAD, and the cited other AD approaches. Tab. 4 reports the evaluation results. It demonstrates that the adversarial examples generated on ResNet-34 model have stronger transferability than VGG-16, which may be attributed to their similar architecture. Moreover, AdaAD and AdaIAD consistently outperform other approaches in prediction accuracy, which indicates their effectiveness in mitigating transfer-based attacks.

4.4. Self Adversarial Distillation

Empirical studies have shown that self adversarial distillation is a promising technique for enhancing the performance of AT [20, 49, 50]. In this section, we examine whether the proposed methods are also effective in the context of self-adversarial distillation. We first train a robust model using AT or its variations, and then directly use it as the teacher model to distillate a student model with the same network architecture. We conduct the experiment on ResNet-18 over CIFAR-10 and present the results in Tab. 5. From Tab. 2 and Tab. 5, we observe that self adversarial distillation by AdaAD and AdaIAD can improve the robustness performance of PGD-AT and TRADES. In particular, our proposed AdaIAD can significantly increase the AA accuracy from 47.69% to 50.19%, indicating its promising performance in self adversarial distillation.

4.5. The Effect of Alleviating Robust Overfitting

The prior work [46] studied the relationship between robust overfitting and weight loss landscape for AT methods, indicating the flatter the weight loss landscape is, the less likely robust overfitting occurs during the AT training process. On the other hand, AD methods can achieve better performance than AT in most scenarios, as shown in Tab. 2 and Tab. 3. To further explore the effectiveness of AD methods, we plot the weight loss landscape of AD methods and PGD-AT. Fig. 3 demonstrates the weight loss landscape of PGD-AT gradually becomes sharper while the landscape of AdaAD almost keeps steady during the training process between 100 and 180 epochs. A similar trend could also be observed for other AD methods. The results reveal that AD methods can result in a flatter weight loss landscape to avoid robust overfitting. Our results can coincide with and extend the analysis in [46] for AD methods.

5. Conclusion

In this paper, we propose to engage a robust teacher model in the inner optimization process to perform adaptive adversarial distillation (AdaAD) for building robust lightweight models. AdaAD adaptively searches for match points representing the upper bound of the prediction discrepancy between student and teacher models for alignment. We analytically show that AdaAD partially resolves the overcorrection problem towards model smoothness commonly faced by most of existing AT and AD methods, thereby reconciling the adversarial trade-off between accuracy and robustness. Moreover, AdaAD allows for a larger search radius ϵ in the inner optimization, which benefits the distillation process. Extensive experimental results demonstrate AdaAD and AdaIAD outperform existing AD and AT methods in most scenarios.

Acknowledgement

The work of Yi Wang was supported in part by Natural Science Foundation of China (grant no. 61876038).

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*, pages 484–501, 2020. 2

[2] Muhammad Awais, Fengwei Zhou, Chuanlong Xie, Jiawei Li, Sung-Ho Bae, and Zhenguo Li. Mixacm: Mixup-based robustness transfer via distillation of activated channel maps. In *NeurIPS*, pages 4555–4569, 2021. 2

[3] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *ECML PKDD*, pages 387–402, 2013. 1

[4] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *SIGKDD*, pages 535–541, 2006. 2

[5] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57, 2017. 2

[6] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C. Duchi, and Percy Liang. Unlabeled data improves adversarial robustness. In *NeurIPS*, pages 11190–11201, 2019. 2

[7] Alvin Chan, Yi Tay, and Yew-Soon Ong. What it thinks is important is important: Robustness transfers through input gradients. In *CVPR*, pages 329–338, 2020. 2

[8] Dian Chen, Hongxin Hu, Qian Wang, Yinli Li, Cong Wang, Chao Shen, and Qi Li. CARTL: cooperative adversarially-robust transfer learning. In *ICML*, pages 1640–1650, 2021. 2

[9] Erh-Chung Chen and Che-Rung Lee. LTD: low temperature distillation for robust adversarial training. *CoRR*, abs/2111.02331, 2021. 5, 6, 7

[10] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *IEEE Symposium on Security and Privacy*, pages 1277–1294, 2020. 2

[11] Mingyang Chen, Junda Lu, Yi Wang, Jianbin Qin, and Wei Wang. DAIR: A query-efficient decision-based attack on image retrieval systems. In *SIGIR*, pages 1064–1073, 2021. 2

[12] Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *ICLR*, 2021. 1, 5

[13] Minhao Cheng, Simranjit Singh, Patrick H. Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. In *ICLR*, 2020. 2

[14] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *NeurIPS*, 2021. 7

[15] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*, pages 2196–2205, 2020. 2

[16] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, pages 2206–2216, 2020. 2

[17] dd Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *ICML*, pages 9155–9166, 2020. 1, 5

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019. 1

[19] Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. advertorch v0.1: An adversarial robustness toolbox based on pytorch. *CoRR*, abs/1902.07623, 2019. 6

[20] Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *AAAI*, pages 3996–4003, 2020. 1, 2, 3, 5, 6, 8

[21] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 1, 2, 4

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 5, 7

[23] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. 1, 2

[24] Xiaojun Jia, Yong Zhang, Baoyuan Wu, Ke Ma, Jue Wang, and Xiaochun Cao. LAS-AT: adversarial training with learnable attack strategy. In *CVPR*, pages 13388–13398, 2022. 1, 2

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1106–1114, 2012. 1

[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, 4, 5

[28] Javier Maroto, Guillermo Ortiz-Jiménez, and Pascal Frossard. On the benefits of knowledge distillation for adversarial robustness. *CoRR*, abs/2203.07159, 2022. 1, 2, 3

[29] Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-student curriculum learning. *IEEE Trans. Neural Networks Learn. Syst.*, 31(9):3732–3740, 2020. 1

[30] Tianyu Pang, Min Lin, Xiao Yang, Jun Zhu, and Shuicheng Yan. Robustness and accuracy could be reconcilable by (proper) definition. In *ICML*, pages 17258–17277, 2022. 5

[31] Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *ICLR*, 2021. 1, 5, 6, 7

[32] Tianyu Pang, Xiao Yang, Yinpeng Dong, Taufik Xu, Jun Zhu, and Hang Su. Boosting adversarial training with hypersphere embedding. In *NeurIPS*, pages 7779–7792, 2020. 1, 2

[33] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *IEEE European Symposium on Security and Privacy*, pages 372–387, 2016. 2

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019. 6

[35] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A. Mann. Data augmentation can improve robustness. In *NeurIPS*, pages 29935–29948, 2021. 2

[36] Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning. In *ICML*, pages 8093–8104, 2020. 1, 2, 6

[37] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 5

[38] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *NeurIPS*, pages 5019–5031, 2018. 2, 5

[39] Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? In *ICLR*, 2022. 2

[40] Ali Shafahi, Parsa Saadatpanah, Chen Zhu, Amin Ghiasi, Christoph Studer, David W. Jacobs, and Tom Goldstein. Adversarially robust transfer learning. In *ICLR*, 2020. 2

[41] Rulin Shao, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. How and when adversarial robustness transfers in knowledge distillation? *CoRR*, abs/2110.12072, 2021. 2

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 7

[43] 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

[44] Pratik Vaishnavi, Kevin Eykholt, and Amir Rahmati. Transferring adversarial robustness through robust representation matching. In Kevin R. B. Butler and Kurt Thomas, editors, *USENIX*, pages 2083–2098, 2022. 2

[45] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. KDGAN: knowledge distillation with generative adversarial networks. In *NeurIPS*, pages 783–794, 2018. 1

[46] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, pages 2958–2969, 2020. 8

[47] 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*, pages 7472–7482, 2019. 2

[48] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan S. Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *ICLR*, 2021. 1

[49] 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. In *ICLR*, 2022. 1, 2, 4, 5, 6, 8

[50] Bojia Zi, Shihao Zhao, Xingjun Ma, and Yu-Gang Jiang. Revisiting adversarial robustness distillation: Robust soft labels make student better. In *ICCV*, pages 16423–16432, 2021. 1, 2, 3, 5, 6, 8