Adversarial Finetuning with Latent Representation Constraint to Mitigate Accuracy-Robustness Tradeoff

Satoshi Suzuki [1] Shin'ya Yamaguchi [1,2] Shoichiro Takeda [3] Sekitoshi Kanai [1]
Naoki Makishima [1] Atsushi Ando [1,3] Ryo Masumura [1]
[1]NTT Computer and Data Science Laboratories [2]Kyoto University
[3]NTT Human Informatics Laboratories

{satoshixv.suzuki, shinya.yamaguchi, shoichiro.takeda, sekitoshi.kanai,
naoki.makishima, atsushi.ando, ryo.masumura}@ntt.com

Abstract

This paper addresses the tradeoff between standard accuracy on clean examples and robustness against adversarial examples in deep neural networks (DNNs). Although adversarial training (AT) improves robustness, it degrades the standard accuracy, thus yielding the tradeoff. To mitigate this tradeoff, we propose a novel AT method called ARREST, which comprises three components: (i) adversarial finetuning (AFT), (ii) representation-guided knowledge distillation (RGKD), and (iii) noisy replay (NR). AFT trains a DNN on adversarial examples by initializing its parameters with a DNN that is standardly pretrained on clean examples. RGKD and NR respectively entail a regularization term and an algorithm to preserve latent representations of clean examples during AFT. RGKD penalizes the distance between the representations of the standardly pretrained and AFT DNNs. NR switches input adversarial examples to nonadversarial ones when the representation changes significantly during AFT. By combining these components, ARREST achieves both high standard accuracy and robustness. Experimental results demonstrate that ARREST mitigates the tradeoff more effectively than previous AT-based methods do.

1. Introduction

Deep neural networks (DNNs) have demonstrated impressive performance for various computer vision tasks [18, 28, 32, 45, 51, 55]. However, standardly trained DNNs can easily be deceived by adversarial examples [15,56], causing incorrect predictions. Such adversarial examples are images with maliciously designed, human-imperceptible perturbations to deceive a DNN. As DNNs penetrate almost every corner of our daily life (*e.g*., autonomous driving), defense techniques against adversarial examples are becoming increasingly important.

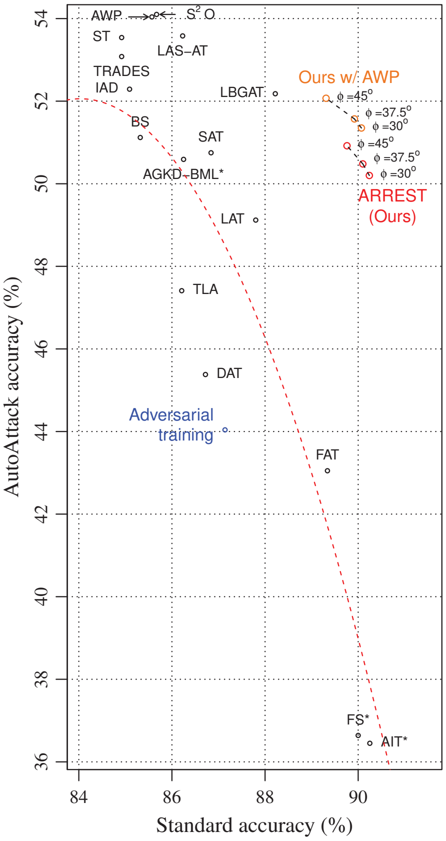


Figure 1. Relationship between the standard and AutoAttack accuracies of various existing methods (see Appendix A) and our proposed method (ARREST) on CIFAR-10. * indicates a result obtained with WideResNet-28-10 [66]; the other results were obtained with WideResNet-34-10. We also evaluated our method by integrating it with the state-of-the-art AWP method [63], as denoted by orange points. The red dashed line is an approximated curve of the accuracy-robustness tradeoff.

The many defense techniques include feature squeezing [65], input denoising [17,49], adversarial detection [30, 33], gradient regularization [41], and adversarial training [15, 37]. Among these, adversarial training (AT) has attracted much attention as a promising defense method. AT attempts to build a robust DNN through training on adversarial examples that are generated online to maximally deceive the on-training DNN [37]. Since AT's effectiveness was demonstrated by Mądry *et al*., a remarkable number of improvements have been proposed [5, 8, 12, 23, 24, 26, 29, 31, 34, 40, 52, 58, 59, 61–63, 68–72].

Although AT is the *de facto* standard method to build DNNs that are robust against adversarial examples, it has the disadvantage of degrading the classification accuracy on clean examples (*i.e*., the standard accuracy). This implies a tradeoff between the adversarial robustness and standard accuracy [20, 57], which we refer to as the *accuracy-robustness tradeoff*. Figure 1 shows the accuracy-robustness tradeoff on CIFAR-10 [27] for various existing methods evaluated by AutoAttack [11]. One particular state-of-the-art method, AWP [63], achieves high robustness, but its standard accuracy is 85.57 %, which is degraded from 95.37 % with a standardly trained DNN. This tradeoff limits the practical applications of AT, as many real-world DNN applications require high standard accuracy and cannot afford much degradation.

Several studies [12, 52, 70, 71] have attempted to mitigate the tradeoff; however, the standard accuracy is still degraded from the original accuracy of a standardly trained DNN. One possible reason for this degradation is the distribution mismatch [54, 64], which indicates that clean and adversarial examples have different underlying distributions [54, 64]. This mismatch suggests that if we train a robust DNN from scratch, like in the above studies, the latent representation will significantly diverge from that of a standardly trained DNN on clean examples (see Table 3). Hence, there is room for improvement in terms of obtaining suitable latent representations of both clean and adversarial examples.

In this paper, we propose a novel method to mitigate the accuracy-robustness tradeoff in AT, called AdversaRial finetuning with REpresentation conSTraint (**ARREST**). The idea behind our method is to obtain suitable representations of adversarial examples while preserving suitable representations of clean examples from standardly trained DNNs. To this end, ARREST comprises three key components: (i) *adversarial finetuning* (AFT), (ii) *representation-guided knowledge distillation* (RGKD), and (iii) *noisy replay* (NR). ARREST uses a two-step training process for robust DNNs, with standard pretraining of DNNs on clean examples followed by finetuning on adversarial examples to increase robustness. We especially refer to the second step as AFT. AFT encourages a DNN to obtain suitable representations of both clean and adversarial examples through finetuning with a standardly pretrained DNN, in contrast to previous studies that trained the DNN from scratch [12, 52, 70, 71]. We also propose RGKD and NR to preserve representations of clean examples from the pretrained DNN by alleviating the distribution mismatch issue [54, 64] during AFT. Inspired by knowledge distillation [19, 48], RGKD penalizes the distance between the on-training DNN's representation and that of the pretrained DNN. While RGKD modifies the objective function of training, NR modifies the perturbation of inputs in AFT. When the on-training DNN's representation of a certain clean example significantly diverges from that of the pretrained DNN, NR switches the input from an adversarial example to a noisy one, which is a clean example with added uniform random noise. NR thus serves to "remind" the DNN of the standard pretraining and encourage representations of clean examples to be close to the pretrained DNN's original representations.

We experimentally demonstrate that ARREST achieves an impressive performance. For example, Fig. 1 shows its qualitative effectiveness in mitigating the accuracy-robustness tradeoff, as the results for ARREST are clustered on the upper-right side. Furthermore, we quantitatively evaluate the degree of tradeoff mitigation with a new metric inspired by the BD-Rate [3, 53] utilized in the field of video compression research. Specifically, our metric calculates the distance from the tradeoff by approximating a curve to represent it (red dashed line in Fig. 1). In terms of this metric, ARREST achieves a state-of-the-art performance, thus confirming both its qualitative and quantitative effectiveness.

Our main contributions are threefold:

1. We propose a novel adversarial training method, ARREST, to mitigate the accuracy-robustness tradeoff. ARREST comprises three components that work complementarily to obtain suitable representations of both clean and adversarial examples.

2. We conduct a wide range of experiments to demonstrate ARREST's effectiveness. Overall, the experimental results provide insights into the strengths of ARREST and the properties of its components.

3. We propose a novel quantitative evaluation metric inspired by the BD-Rate, and we show that ARREST achieves state-of-the-art performance in terms of this metric.

2. Related Work

Adversarial Attacks. Because of the documented vulnerability of DNNs [56], many works have proposed novel adversarial attack techniques [6, 15, 37, 38]. For example, Mądry *et al*. [37] proposed a projected gradient descent (PGD) method, which is a multistep version of the fast gradient sign method (FGSM) [15]. Recently, Croce and Hein [11] proposed two improved versions of the PGD attack, namely APGD-CE and APGD-DLR, which do not require selecting a step size or alternating a loss function, unlike the original PGD. Then, they combined those two methods with two other complementary adversarial attacks (FAB [10] and Square [1]) to evaluate robustness through an approach called AutoAttack. Recent studies have widely used AutoAttack to evaluate robustness, because it provides more reliable evaluation than the traditional PGD-based evaluation. Croce and Hein also applied AutoAttack on tens of previous AT-based methods and provided a comprehensive leaderboard [9]. In this paper, we mainly apply AutoAttack to evaluate the adversarial robustness, given that it is common and reliable.

Adversarial Training. Many defense methods have been proposed to improve model robustness against adversarial attacks. Among them, adversarial training (AT) [15, 37] has attracted much attention. AT attempts to build robust DNNs through training with online-generated adversarial examples that try to maximally deceive the on-training DNN. Goodfellow *et al*. [14] used FGSM to generate the adversarial examples; more recently, Mądry *et al*. [37] used the PGD method. Because of its high robustness, AT with the PGD method is currently the *de facto* standard method to build robust DNNs against adversarial examples.

Mitigation of Accuracy-Robustness Tradeoff. Several studies have attempted to mitigate the accuracy-robustness tradeoff. Zhang *et al*. [70] proposed a defense method called TRADES, which adjusts the tradeoff with a hyperparameter. TRADES is based on adversarial logit pairing (ALP) [25], which increases robustness by encouraging the outputs from clean examples and adversarial examples to be similar to each other. Cui *et al*. [12] proposed a method that guides a DNN's output to be the same as that of a standardly trained DNN, and they demonstrated that this method mitigates the tradeoff more than TRADES or ALP. Zhang *et al*. [71] and Sitawarin *et al*. [52] attempted to mitigate the tradeoff by using a curriculum learning strategy [2]. Although these are important methods that address the accuracy-robustness tradeoff in AT, they still degrade the standard accuracy from the original accuracy of a standardly trained DNN. We argue that this degradation is due to the distribution mismatch. In ARREST, we use three complementary components to address this issue. As another methodology to address the distribution mismatch, Xie *et al*. [64] proposed using different batch normalization layers [21] for clean and adversarial examples. However, their approach requires knowing at test time whether an input example is clean or adversarial, which may not be practical.

In another line of research, a methodology has been proposed to mitigate the accuracy-robustness tradeoff by using additional real or synthetic examples for training [16, 42, 43]. In general, AT requires more training data to generalize a DNN than standard training does [50], and DNNs often suffer from overfitting during AT [46]. This methodology can alleviate the overfitting and thus mitigate the accuracy-robustness tradeoff. However, the use of additional examples often leads to prohibitive increases in the training time cost. Thus, we focus primarily on methods that do not require additional examples. Note that we will also demonstrate that using the additional examples in [43] can provide benefits for ARREST (see Table 5).

Adversarial Finetuning. AFT has been used in several recent studies [22, 39]. Unlike ARREST, however, the aim of those studies was not mitigation of the accuracy-robustness tradeoff. Jeddi *et al*. [22] used AFT to make AT faster by reducing the number of training epochs. Moosavi-Dezfooli *et al*. [39] analyzed the effect of AT by comparing a DNN's decision boundary before and after AFT was applied. In contrast, ARREST aims to apply AFT to mitigate the accuracy-robustness tradeoff by incorporating RGKD and NR.

3. Preliminaries

We first describe the conventional AT method [37]. In general, AT directly incorporates adversarial examples into the training process to solve the following min-max problem:

$$\min_{\theta_\mathrm{r}} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathbb{D}} \left[\max_{||\boldsymbol{\delta}||_p \le \varepsilon} \mathcal{L}_\mathrm{CE}(f(\boldsymbol{x}+\boldsymbol{\delta};\theta_\mathrm{r}), y) \right]. \quad (1)$$

Here, θ_r represents the parameters of the DNN $f(\cdot)$, and $\mathcal{L}_\mathrm{CE}(\cdot)$ is the cross-entropy loss, which is commonly used for classification tasks. $\boldsymbol{x}$ and y are a clean training example and its ground-truth label, respectively, which are sampled from an underlying data distribution $\mathbb{D}$. In AT, the cross-entropy loss is calculated with an adversarial example, $\boldsymbol{x}+\boldsymbol{\delta}$. The L_p-norm of $\boldsymbol{\delta}$, $||\boldsymbol{\delta}||_p$, is bounded by a perturbation budget ε. As the inner maximization problem in Eq. (1) cannot be solved in closed form, the PGD method [37] is commonly used to solve it heuristically.

In the conventional AT method [37], a robust DNN is trained from scratch, *i.e*., the parameters θ_r of $f(\cdot)$ are randomly initialized. This training style has been followed by almost all AT improvements [5, 8, 12, 23, 24, 26, 31, 34, 40, 52, 58, 59, 61–63, 68–72]. We found that a robust DNN trained from scratch on adversarial examples obtains significantly different representations from a standardly trained DNN because of the distribution mismatch issue [54, 64] (see Table 3). Therefore, such robust DNNs have difficulty obtaining suitable representations of clean examples.

In contrast with the above studies, in this paper, we propose ARREST, which finetunes a standardly pretrained

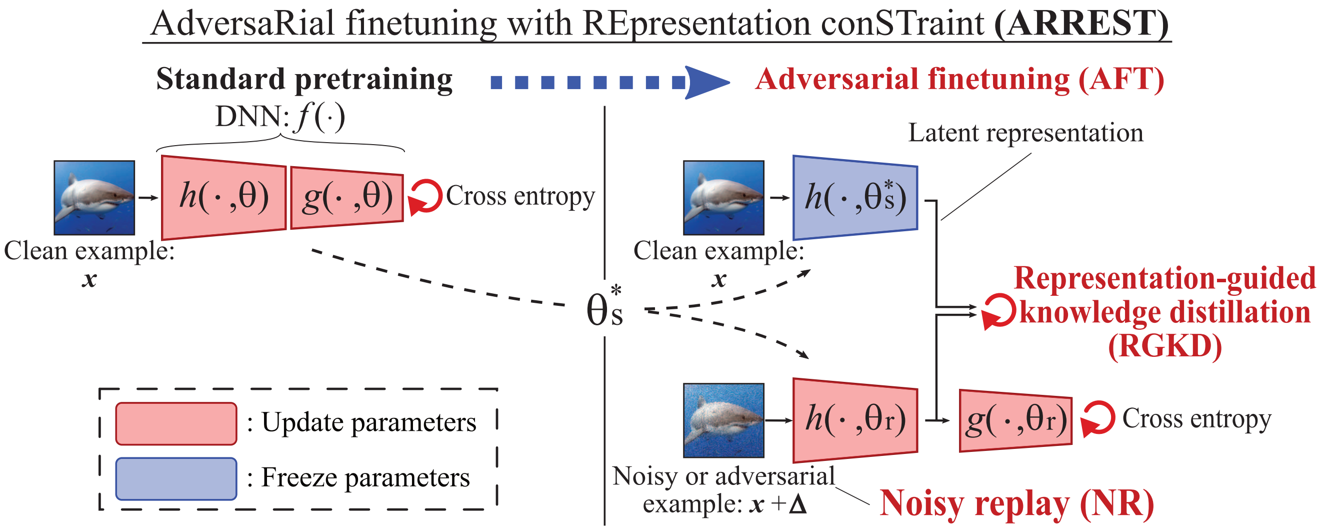


Figure 2. Overview of the proposed method, ARREST.

DNN with adversarial examples to increase its robustness, while introducing a constraint on the latent representation. As a result, ARREST generates a DNN that is robust against adversarial examples but achieves high standard accuracy. To formulate the constraint, we divide the DNN $f(\cdot)$ into $g \circ h(\cdot)$, where $h(\cdot)$ maps an input example to its corresponding latent representation, and $g(\cdot)$ is a classifier in $f(\cdot)$. The dividing point between $h(\cdot)$ and $g(\cdot)$ depends on the DNN architecture, though the penultimate layer is usually used [34, 58].

4. Proposed Method

In this paper, we propose ARREST to mitigate the accuracy-robustness tradeoff. Our main idea is to obtain suitable representations of adversarial examples while preserving suitable representations of clean examples from standardly trained DNNs. To this end, ARREST comprises three key components: (i) *adversarial finetuning* (AFT), (ii) *representation-guided knowledge distillation* (RGKD), and (iii) *noisy replay* (NR). Figure 2 and Algorithm 1 provide an overview and the detailed procedure of ARREST, respectively. In this section, we explain each component in detail.

4.1. Adversarial Finetuning (AFT)

We first explain AFT. In ARREST, we use a two-step training process to obtain robust DNNs, where the steps are standard pretraining of DNNs on clean examples and finetuning of the pretrained DNNs on adversarial examples via RGKD and NR. We especially refer to the second step as AFT.

Before AFT, we standardly train on clean examples to obtain the following DNN:

$$\theta_{\mathrm{s}}^* = \operatorname*{argmin}_{\theta} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathbb{D}} \left[\mathcal{L}_{\mathrm{CE}}(f(\boldsymbol{x};\theta),y)\right]. \qquad (2)$$

In AFT, we finetune the pretrained DNN via the min-max problem in Eq. (1). Specifically, as given in Algorithm 1, we initialize the parameters of a DNN θ_{r} with those of θ_{s}^* (line 1) and optimize θ_{r} iteratively (line 9).

Through the use of θ_{s}^* for the initial parameters, AFT helps the DNN obtain suitable representations of clean examples. However, AFT does not explicitly impose constraints on the DNN's representations. As a result, the distribution mismatch issue [54, 64] causes the DNN's representations of clean examples to gradually diverge from the original representations by the standardly pretrained DNN during AFT. To address this issue, we propose the application of RGKD and NR to constrain the DNN's representations.

4.2. Representation-Guided Knowledge Distillation (RGKD)

In RGKD, we penalize the distance between the representations of the DNN θ_{r} and the standardly pretrained DNN θ_{s}^* during AFT. RGKD was inspired by the knowledge distillation methodology [19, 48] in model compression. Knowledge distillation was originally proposed to guide a small DNN using knowledge (*i.e.*, an output or representation) from a large DNN to reduce the computation cost. Here, we apply this concept to mitigate the accuracy-robustness tradeoff by guiding the DNN with a representation from the standardly pretrained DNN.

We define the loss of RGKD as follows:

$$\mathcal{L}_{\mathrm{RGKD}}(\boldsymbol{x},\boldsymbol{\delta},\theta_{\mathrm{r}}) = d\left(h(\boldsymbol{x}+\boldsymbol{\delta};\theta_{\mathrm{r}}), h(\boldsymbol{x};\theta_{\mathrm{s}}^*)\right), \qquad (3)$$

Algorithm 1 AdversaRial finetuning with REpresentation conSTraint (ARREST).

```
Input: parameters of standardly trained DNN $\theta_{\mathrm{s}}^{*}$, perturbation budget $\varepsilon$, distance threshold $\tau$, training dataset $\mathbb{D}$, learning rate $\eta$, batch size $m$
Output: parameters of DNN $\theta_{\mathrm{r}}$
 1: Initialize parameters $\theta_{\mathrm{r}} \leftarrow \theta_{\mathrm{s}}^{*}$                // AFT
 2: while until convergence do
 3:     Sample mini-batch $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$ from $\mathbb{D}$
 4:     for $i = 1, \cdots, m$ do
 5:         Calculate $a = d\left(h(\boldsymbol{x}_i; \theta_{\mathrm{r}}), h(\boldsymbol{x}_i; \theta_{\mathrm{s}}^{*})\right)$
 6:         Obtain $\boldsymbol{\Delta}_i$                            // NR
                where $\begin{cases} \boldsymbol{\Delta}_i = \boldsymbol{\delta}_i \text{ obtained by PGD} & (a \leq \tau) \\ \boldsymbol{\Delta}_i \sim \mathcal{U}(-\varepsilon, \varepsilon) & (a > \tau) \end{cases}$
 7:         Calculate $\mathcal{L}(\boldsymbol{x}_i, \boldsymbol{\Delta}_i, y_i, \theta_{\mathrm{r}})$ in Eq. (4)   // RGKD
 8:     end for
 9:     $\theta_{\mathrm{r}} \leftarrow \theta_{\mathrm{r}} - \eta \frac{1}{m} \sum_{i=1}^{m} \nabla_{\theta_{\mathrm{r}}} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{\Delta}_i, y_i, \theta_{\mathrm{r}})$
10: end while
```

where $d(\cdot)$ is a distance function, such as the angular distance [60], to measure the similarity between two representations. From the definition of $h(\cdot)$ in Section 3, the arguments of $d(\cdot)$ are the representation by the DNN θ_{r} of an adversarial example $\boldsymbol{x}+\boldsymbol{\delta}$ and that by the standardly pretrained DNN θ_{s}^{*} of a clean example $\boldsymbol{x}$. Note that the parameters of θ_{s}^{*} are not updated (frozen), as shown in Fig. 2. By minimizing this loss, we can penalize the DNN's representation if it diverges from the original representation $h(\boldsymbol{x}; \theta_{\mathrm{s}}^{*})$.

In ARREST, the loss for optimizing DNNs is the summation of $\mathcal{L}_{\mathrm{CE}}$ and $\mathcal{L}_{\mathrm{RGKD}}$:

$$\begin{aligned} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\delta}, y, \theta_{\mathrm{r}}) = \mathcal{L}_{\mathrm{CE}}&(f(\boldsymbol{x}+\boldsymbol{\delta}; \theta_{\mathrm{r}}), y) \\ &+ \lambda\, \mathcal{L}_{\mathrm{RGKD}}(\boldsymbol{x}, \boldsymbol{\delta}, \theta_{\mathrm{r}}), \end{aligned} \tag{4}$$

where λ is a hyperparameter for determining the effect of RGKD on optimization. As seen in lines 7 and 9 of Algorithm 1, this loss is calculated across all examples in a mini-batch and used for optimization of θ_{r}.

Several AT methods [12, 58] have also used the knowledge distillation methodology and guided a DNN by using a logit (output) [12] or a representation transferred to an attention map [58]. We found experimentally that RGKD is the best of those methods for mitigating the tradeoff (see Table 4).

4.3. Noisy Replay (NR)

While RGKD modifies the objective function for training, NR addresses the distribution mismatch issue by modifying the perturbation of inputs in AFT. Specifically, it monitors the distance between $h(\boldsymbol{x}; \theta_{\mathrm{r}})$ and $h(\boldsymbol{x}; \theta_{\mathrm{s}}^{*})$, *i.e.*, $d\left(h(\boldsymbol{x}; \theta_{\mathrm{r}}), h(\boldsymbol{x}; \theta_{\mathrm{s}}^{*})\right)$. When this distance exceeds a predefined threshold, NR attempts to avoid increasing the distance further by utilizing the replay technique [4, 13, 47]. This technique was originally developed to address catastrophic forgetting during continual learning [35]. It retrains a DNN with data from a previous task during current task training. Recent research [44] found that, during a current task, the replay technique preserves a suitable latent representation obtained by the previous task. In our case, the previous and current tasks correspond to standard pretraining on clean examples and AFT on adversarial examples, respectively. Via this analogy, NR switches input examples by using $d\left(h(\boldsymbol{x}; \theta_{\mathrm{r}}), h(\boldsymbol{x}; \theta_{\mathrm{s}}^{*})\right)$:

$$\boldsymbol{x} + \boldsymbol{\Delta}, \text{ where } \begin{cases} \boldsymbol{\Delta} \sim \mathcal{U}(-\varepsilon, \varepsilon) & \text{if } d\left(h(\boldsymbol{x}; \theta_{\mathrm{r}}), h(\boldsymbol{x}; \theta_{\mathrm{s}}^{*})\right) > \tau \\ \boldsymbol{\Delta} = \boldsymbol{\delta} & \text{otherwise,} \end{cases} \tag{5}$$

where $\mathcal{U}(-\varepsilon, \varepsilon)$ denotes a uniform distribution bounded by the absolute value of ε. Empirically, we found that adjusting the threshold value τ plays a role in balancing the tradeoff. Specifically, the robustness increases as τ increases, while the accuracy increases as τ decreases. Therefore, in the experimental section, we use several τ values for achieving various tradeoffs (see Figs. 1 and 3).

With Eq. (5), NR inputs random noisy examples to the DNN when the distance exceeds τ. We could also naively consider inputting a clean example ($\boldsymbol{\Delta} = \boldsymbol{0}$) when $d\left(h(\boldsymbol{x}; \theta_{\mathrm{r}}), h(\boldsymbol{x}; \theta_{\mathrm{s}}^{*})\right) > \tau$. However, we observed that this approach does not work well, and we discuss the reason in Section 6. The NR processes are in lines 5 and 6 of Algorithm 1. Finally, the loss in Eq. (4) is calculated with $\boldsymbol{\Delta}$, rather than $\boldsymbol{\delta}$; that is, $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\Delta}, y, \theta_{\mathrm{r}})$ is used for optimization in line 7.

5. Experiments

5.1. Quantitative Evaluation Metrics

Before describing our experiments, we introduce a new metric, accuracy robustness distance (ARDist), to quantitatively evaluate mitigation of the accuracy-robustness tradeoff. ARDist was inspired by the BD-Rate metric that is commonly used in the field of video compression [3,53] and has a similar purpose. Specifically, the BD-Rate quantitatively evaluates the *tradeoff between a codec's bitrate and distortion* by approximating a curve representing the tradeoff. Similarly, ARDist uses existing methods to approximate a curve representing the accuracy-robustness tradeoff. This approximation is easily implemented by polynomial regression with a cubic function on all data points of the existing methods (listed in Appendix A), including ARREST[1]. In this paper, we made this approximation for

[1] We used the result with $\phi = 30°$ for approximation.

Table 1. Quantitative evaluation of ARREST and four existing state-of-the-art methods via the Sum and ARDist metrics for CIFAR-10 and CIFAR-100 datasets. The adversarial robustness was calculated utilizing AutoAttack. Bold type indicates the highest value for each metric.

	CIFAR-10				CIFAR-100			
	Standard	AutoAttack	Sum	ARDist	Standard	AutoAttack	Sum	ARDist
AT [37]	87.14%	44.04%	131.18	-1.500	59.59%	22.86%	82.45	-3.268
LAS-AT [23]	86.23%	53.58%	139.81	2.236	61.80%	29.03%	90.83	3.189
AWP [63]	85.57%	54.04%	139.61	2.314	60.38%	28.86%	89.24	2.424
S^{2}O [24]	85.67%	54.10%	139.77	2.410	63.40%	27.60%	91.00	2.786
LBGAT [12]	88.22%	52.18%	140.40	2.706	70.25%	26.73%	96.98	6.639
ARREST (ours)	90.24%	50.20%	**140.44**	**3.521**	73.05%	24.32%	**97.37**	**7.165**

CIFAR-10 and CIFAR-100 datasets. The obtained equations of the approximated curves are

$$c_{10}(x) = (9.877 \cdot 10^{-5})\, x^3 - 0.3922x^2 + 63.82x - 2600,$$
$$c_{100}(x) = (5.615 \cdot 10^{-4})\, x^3 - 0.1582x^2 + 12.44x - 271.8,$$

where x indicates the standard accuracy, and $c_{10}(\cdot)$ and $c_{100}(\cdot)$ denote the approximated curves for CIFAR-10 and CIFAR-100, respectively. The red dashed line in Fig. 1 is a concrete example of an approximated curve for CIFAR-10, and it fits the accuracy-robustness tradeoffs of the existing methods. ARDist evaluates the mitigation by calculating the distance between the approximated curve and a point given by the method being evaluated. This calculation can be done numerically, and we provide Python source code in Appendix B. ARDist yields positive or negative values depending on whether a method's point is above or below the approximated curve.

The simple sum of the accuracy and robustness is sometimes used as another quantitative metric [52]. This is an important metric for evaluating a method's absolute performance in terms of its accuracy and robustness. In contrast, ARDist can evaluate the relative performance when comparing the tradeoffs of a new method and existing methods. In this paper, we use both metrics to quantitatively evaluate the tradeoff mitigation from multiple perspectives.

5.2. Experimental Settings

Datasets. We evaluated ARREST on two popular datasets: CIFAR-10 and CIFAR-100 [27]. CIFAR-10 dataset contains 60,000 color images having a size of 32×32 in 10 classes, with 50,000 training and 10,000 test images. CIFAR-100 dataset contains 50,000 training and 10,000 test images in 100 classes.

Optimization Details. We adopted the SGD optimizer with a momentum of 0.9 and weight decay of 5×10^{-4}. The batch size was set to 128. In the standard pretraining, we set the number of training epochs to 100. The learning rate started at 0.1 and then decayed by ×0.1 with transition epochs $\{75, 90\}$, following Zhang *et al.* [70]. In AFT, we set the number of training epochs to 20. The learning rate started at 0.025, decayed to 0.02 at 11 epochs, and then decayed by half every two epochs thereafter. We used NR only in the first 10 of 20 AFT epochs, which minimized the sacrificed robustness.

Implementation Details. We used WideResNet-34-10 [66] as the main DNN architecture, following many previous studies [12, 26, 29, 37, 62, 63, 70, 71]. On CIFAR-10, we also used ResNet-18 [18] to evaluate ARREST's flexibility with respect to the architecture. Following previous works [12, 37, 70], $\boldsymbol{\delta}$ was bounded by the L_∞-norm. We used a perturbation budget of $\varepsilon = 8/255$ for both training and evaluation. In AFT, $\boldsymbol{\delta}$ was obtained by the PGD method with a step size of $2/255$ and 10 iterative steps. The PGD objective function was $\mathcal{L}_{\text{CE}}$ alone, without the loss of RGKD. For ARREST, we used the output of the penultimate layer (just before global pooling) as the latent representation for both architectures, as a higher-dimensional penultimate layer tends to preserve more information [34]. The distance function was the angular distance [60]: $d(\boldsymbol{u}, \boldsymbol{v}) = 1 - \frac{|\boldsymbol{u} \cdot \boldsymbol{v}|}{||\boldsymbol{u}||_2 \cdot ||\boldsymbol{v}||_2}$. Note that RGKD also performs well with other distance functions, *e.g.*, the mean squared error, and we provide those results in Appendix C. We set the hyperparameter λ to 50. We adjusted τ using the form $1 - \cos\phi$. Since τ determines the balance of the tradeoff (as mentioned in Subsection 4.3), we set various values for ϕ: $\{30°, 37.5°, 45°\}$ for CIFAR-10 and $\{30°, 32.5°, 35°\}$ for CIFAR-100.

5.3. Comparison with Existing Methods

To benchmark ARREST's effectiveness, we conducted comparison experiments with existing AT methods on CIFAR-10 and CIFAR-100. Figures 1 and 3 show the standard accuracy and robustness calculated by AutoAttack [11] for the existing methods and ARREST on CIFAR-10 and CIFAR-100, respectively. The details of the existing methods are given in Appendix A. In both figures, ARREST appears in the upper-right relative to the existing methods. As this position indicates both high accuracy and high robustness, these results qualitatively demonstrate the effectiveness of ARREST for mitigating the accuracy-robustness tradeoff as compared to existing methods. It is worth men-

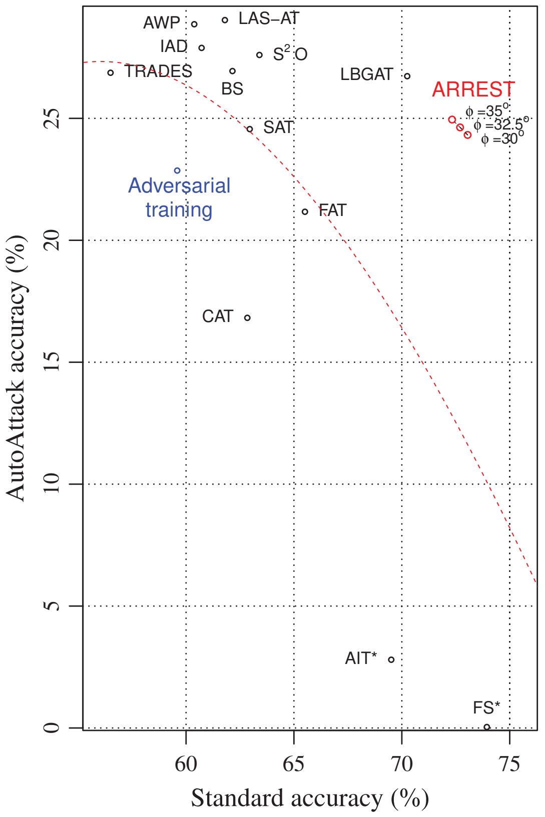


Figure 3. Relationship between the standard and AutoAttack accuracies of existing methods (see Appendix A) and ARREST on CIFAR-100. * indicates a result obtained with WideResNet-28-10; the other results were obtained with WideResNet-34-10. The red dashed line is the tradeoff's approximated curve.

tioning that ARREST integrates easily with other existing techniques for increasing the robustness, such as AWP [63]. In Fig. 1, the results of ARREST with AWP [63] are denoted by the orange points. We can observe that this combination achieves higher robustness with only a slight sacrifice to the standard accuracy. As a result, ARREST with AWP better mitigates the accuracy-robustness tradeoff compared with only using ARREST. In Appendix D, we further compared ARREST with the results of varying the hyperparameter β for TRADES and S^2O (integrated with TRADES). From these comparisons, we can see that the results of ARREST appear on the right side relative to those of TRADES and S^2O. This indicates that ARREST can achieve higher standard accuracy while achieving the same robustness as these methods.

In addition to this qualitative comparison, we conducted a quantitative comparison using the two metrics explained in Subsection 5.1, *i.e.*, ARDist and the sum of the standard accuracy and robustness (Sum). Here, we set ϕ in NR to 30° for both CIFAR-10 and CIFAR-100. Table 1 lists these results, which were obtained by comparing ARREST with AT [37] and four existing state-of-the-art methods (AWP [63], LAS-AT [23], S^2O [24], and LBGAT [12]). Note that these four existing methods had the best scores in terms of Sum and ARDist across all methods listed in Appendix A on both CIFAR-10 and CIFAR-100. As shown in the table, ARREST also achieved a state-of-the-art performance in terms of Sum and ARDist on both datasets. These results indicate that ARREST can obtain more suitable latent representations of both clean and adversarial examples compared to the existing methods.

Table 2. Results obtained with four variations of our method (only AFT, AFT with RGKD, AFT with NR, ARREST) and two baselines (standard training and AT).

	ResNet-18		WideResNet-34-10	
	Standard	AutoAttack	Standard	AutoAttack
ST	**94.53%**	0%	**95.37%**	0%
AT [37]	84.71%	44.19%	87.14%	44.04%
AFT	84.08%	45.36%	87.54%	48.74%
AFT + RGKD	85.11%	46.01%	88.52%	**50.20%**
AFT + NR	85.52%	45.29%	88.94%	48.63%
ARREST	86.63%	**46.14%**	90.24%	**50.20%**

5.4. Ablation Study

We analyzed the ablation effect of each component of ARREST on CIFAR-10 dataset, where ResNet-18 and WideResNet-34-10 were used as the architectures. Table 2 lists the results obtained with four variations of our method (only AFT, AFT with RGKD, AFT with NR, and all components, *i.e.*, ARREST) and two baselines (standard training and AT [37]). For AFT with NR only, we set $\phi = 45°$ because the constraining effect changes in the absence of RGKD, and $\phi = 30°$ does not yield an optimal performance. This ϕ value was searched from 15° to 75° in 15° increments. We used AutoAttack [11] for the evaluation. The results using other attacks are listed in Appendix E.

As seen in Table 2, we found that AFT alone could only provide standard accuracy similar to that of AT. This is because AFT does not explicitly impose constraints on the representation, and the DNN's representations of clean examples gradually diverge from the original representations of the standardly pretrained DNN during AFT. However, AFT with RGKD or NR increased the standard accuracy. These results demonstrate that these constraint techniques help the DNN to preserve the representation during AFT and effectively mitigate the accuracy-robustness tradeoff, as expected. Furthermore, AFT with both RGKD and NR achieved the highest standard accuracy and robustness. As described above, the three key components of ARREST, *i.e.*, AFT, RGKD, and NR, work complementarily to obtain suitable representations of both clean and adversarial examples. Therefore, the ablation study results indicate that the components' complementary roles actually serve to mit-

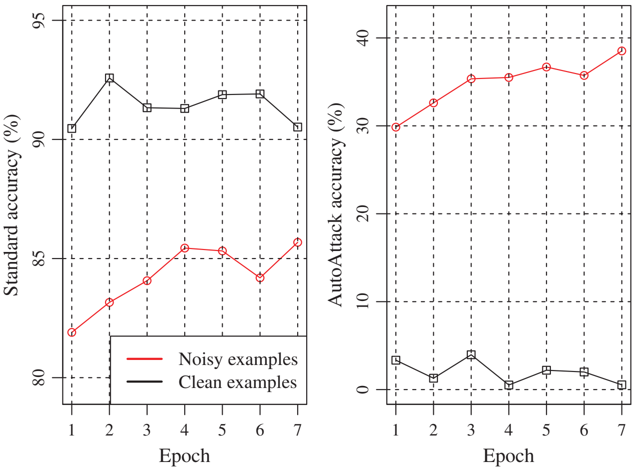


Figure 4. Left: Relationship between the number of epochs and the standard accuracy in AFT with the replay technique. Right: Relationship between the number of epochs and the adversarial robustness with the same DNN used on the left. The black and red lines show results obtained by inputting clean and adversarial examples, respectively.

igate the accuracy-robustness tradeoff. Finally, ARREST achieved a standard accuracy of 2.0 to 3.0 points higher than that of AT while also demonstrating higher adversarial robustness. Overall, these results suggest that ARREST successfully mitigates the accuracy-robustness tradeoff in AT.

6. Analysis of ARREST

In this section, we analyze ARREST from four viewpoints: noisy examples in NR, the effect of ARREST on preserving the representation, comparison of other types of knowledge distillation with RGKD, and the effect of additional examples on ARREST.

Effect of Inputting Noisy Examples on NR. First, we analyzed the effect of random noisy examples on the NR performance. As mentioned in Subsection 4.3, we could consider inputting a clean example in Eq. (5); however, we found that this did not improve the adversarial robustness during AFT at all, as shown on the right in Fig. 4. This is because partial input of clean examples during AFT makes optimization challenging with the distribution mismatch between clean and adversarial examples. As a result, the DNN only obtains suitable representations of clean examples. Actually, it maintains a high standard accuracy, as shown on the left in Fig. 4. To avoid this issue, we use noisy examples in NR. Figure 5 shows a visualization of the representations of clean, noisy, and adversarial examples with a standardly pretrained DNN. Because the uniform noise is nonadversarial, the distribution underlying noisy examples is similar to that underlying clean examples but shifted slightly toward that of adversarial examples. Hence, the use of noisy examples is expected to alleviate the distribution mismatch while enabling the DNN to benefit from the replay technique. In fact, replay with noisy examples, *i.e.*, our proposed NR, can improve the robustness in contrast with the use of clean examples, as shown by the red line on the right in Fig. 4.

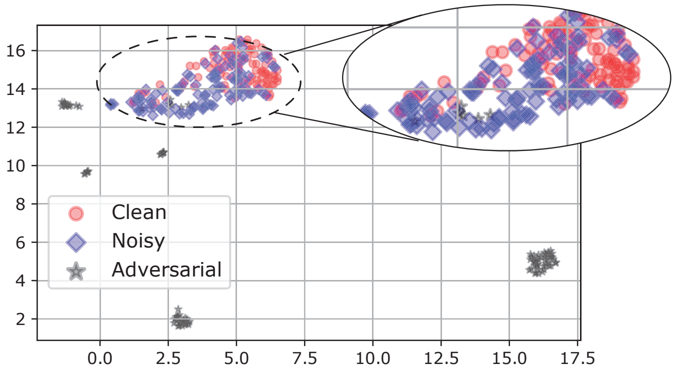
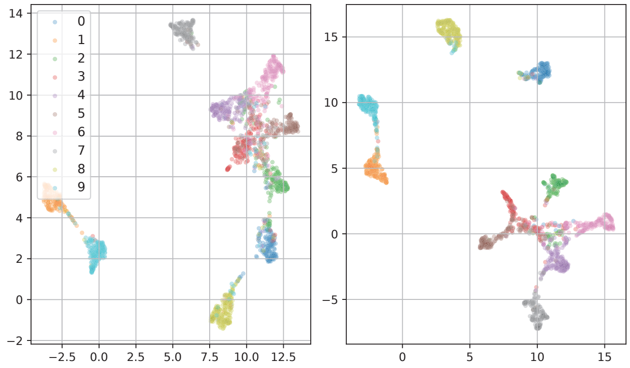


Figure 5. Visualization of representations of clean, noisy, and adversarial examples on a standardly pretrained DNN by using UMAP [36]. From CIFAR-10, 100 randomly selected test images labeled "dog" were used. The solid ellipse shows an enlargement of the dashed region.

Table 3. Comparison of the effectiveness of each ARREST component in preserving representation, with the cosine similarity as the metric.

	Cosine similarity
AT	0.255
AFT	0.753
AFT + RGKD	0.894
AFT + NR	0.764
ARREST	**0.901**

Effect of ARREST on Preserving Representation. Next, we analyzed the effectiveness of ARREST on preserving the representation from a standardly trained DNN. ARREST's three key components were designed to obtain suitable representations of clean and adversarial examples by preserving the representation of clean examples from a standardly trained DNN. To evaluate ARREST's proper operation, we compared the representations of robust and standardly pretrained DNNs. Table 3 lists the cosine similarity between $h(\boldsymbol{x};\theta_\mathrm{r})$ and $h(\boldsymbol{x};\theta^*_\mathrm{s})$ for evaluation on clean test examples from CIFAR-10. As seen in the table, AT obtained a significantly different representation from that of the standardly trained DNN. As explained above, this is due to the distribution mismatch issue [54, 64]. AFT significantly increased the cosine similarity as compared with AT. Moreover, AFT still had a gap between $h(\boldsymbol{x};\theta_\mathrm{r})$ and $h(\boldsymbol{x};\theta^*_\mathrm{s})$ because of the remaining distribution mismatch, but each of RGKD and NR further increased the similarity. Consequently, ARREST achieved the highest cosine



(a) Adversarial training (b) ARREST

Figure 6. Visualization of clean example representations for (a) a robust DNN with AT and (b) ARREST by using UMAP [36]. For visualization, 2,000 randomly selected test images from CIFAR-10 were used.

similarity of 0.901. Furthermore, as shown in Fig. 6, we visualized the representations of clean examples with AT and ARREST. We can see that ARREST (b) obtained more discriminative representations of clean examples than AT did (a). These results indicate that preservation of the representation from the standardly trained DNN leads to higher standard accuracy, as expected.

Comparison with Other Types of Knowledge Distillation. As described in Subsection 4.2, several AT methods [12,58] have applied knowledge distillation methodologies besides RGKD. Here, we compared RGKD with them by replacing the loss in Eq. (3) during AFT with two different methods. First, "logit" [12] guides the DNN with the logit (final output) of the pretrained DNN. Second, "attention map" [58] guides the DNN with a spatial attention map computed by summing the latent representations along the channel dimension [67]. Appendix F provides these methods' detailed formulation and optimization. Table 4 lists the comparison results, which show that RGKD achieved the best performance in terms of both accuracy and robustness. This may be because the comparison methods had a weak penalty effect by transferring the latent representation to a logit or attention map, in contrast with RGKD, which guides the DNN by using the representation as is. For example, guiding with a logit does not directly affect the DNN representation, thus limiting the penalty effect. This comparison indicates that RGKD is more suitable than the other methods for our aim of mitigating the accuracy-robustness tradeoff.

Effect of Additional Examples on ARREST. Finally, we investigated the effect of additional examples on ARREST, an approach that has often been used in previous works [16, 42, 43] to mitigate the tradeoff. Specifically, we used the additional examples from RST [43], which were originally from [7]. Appendix G explains the training setup for this experiment. The results are listed in Table 5. ARREST with additional examples achieved higher standard accuracy and adversarial robustness than RST, which adversarially trains the DNN in the usual manner. These results indicate that ARREST can benefit from additional examples, and the combination is promising for mitigating the accuracy-robustness tradeoff.

Table 4. Comparison of RGKD with other types of knowledge distillation.

	Standard	AutoAttack
Logit [12]	87.45%	49.62%
Attention map [58]	87.70%	49.05%
RGKD	**88.52%**	**50.20%**

Table 5. Results for ARREST with additional examples from RST [43].

	Standard	AutoAttack
RST [43]	91.62%	54.81%
ARREST w/ RST	**93.17%**	**55.73%**

7. Conclusion

We have proposed AdversaRial finetuning with REpresentation conSTraint (**ARREST**) to mitigate the accuracy-robustness tradeoff in adversarial training (AT). ARREST aims to obtain suitable representations of adversarial examples while preserving suitable representations of clean examples from standardly trained DNNs. To this end, ARREST comprises three key components: (i) *adversarial finetuning* (AFT), (ii) *representation-guided knowledge distillation* (RGKD), and (iii) *noisy replay* (NR). It uses a two-step training process to obtain robust DNNs, entailing standard pretraining of DNNs on clean examples and finetuning of the pretrained DNNs on adversarial examples with RGKD and NR. Further, we have proposed a new quantitative evaluation metric, accuracy robustness distance (ARDist), which was inspired by the BD-Rate [3, 53] metric used in video compression research. Using ARDist, we demonstrated the quantitative effectiveness of ARREST in mitigating the tradeoff on CIFAR-10 and CIFAR-100 datasets.

While ARREST efficiently mitigates the accuracy-robustness tradeoff in AT, it could not perfectly eliminate the tradeoff; that is, it could not achieve the same standard accuracy as a standardly trained DNN. A promising direction would be to combine ARREST with additional training examples, as was done to obtain the results in Table 5. This might enable us to maximize the tradeoff mitigation or completely eliminate the tradeoff. We plan to explore this direction in the future.

References

[1] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[2] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *International Conference on Machine Learning (ICML)*, 2009. 3

[3] G. Bjøntegaard. Calculation of average PSNR differences between RD-Curves. *ITU-T Video Coding Experts Group (VCEG)-M33*, 2001. 2, 5, 9

[4] M. Boschini, L. Bonicelli, A. Porrello, G. Bellitto, M. Pennisi, S. Palazzo, C. Spampinato, and S. Calderara. Transfer without forgetting. In *European Conference on Computer Vision (ECCV)*, 2022. 5

[5] Q. Cai, C. Liu, and D. Song. Curriculum adversarial training. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018. 2, 3, 13

[6] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, 2017. 3, 15, 16

[7] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 9, 16

[8] J. Chen, Y. Cheng, Z. Gan, Q. Gu, and J. Liu. Efficient robust training via backward smoothing. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 2, 3, 13

[9] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein. Robustbench: a standardized adversarial robustness benchmark. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3

[10] F. Croce and M. Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning (ICML)*, 2020. 3

[11] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning (ICML)*, 2020. 2, 3, 6, 7, 15

[12] J. Cui, S. Liu, L. Wang, and J. Jia. Learnable boundary guided adversarial training. In *International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 5, 6, 7, 9, 13, 16

[13] S. Farquhar and Y. Gal. Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*, 2018. 5

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 3

[15] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015. 1, 2, 3, 15, 16

[16] S. Gowal, S.-A. Rebuffi, O. Wiles, F. Stimberg, D. A. Calian, and T. Mann. Improving robustness using generated data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3, 9

[17] C. Guo, M. Rana, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 6, 16

[19] G. E. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. 2, 4

[20] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Mądry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[21] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015. 3

[22] A. Jeddi, M. J. Shafiee, and A. Wong. A simple fine-tuning is all you need: Towards robust deep learning via adversarial fine-tuning. *arXiv preprint, arXiv:2012.13628*, 2020. 3

[23] X. Jia, Y. Zhang, B. Wu, J. Wang K. Ma, and X. Cao. LAS-AT: Adversarial training with learnable attack strategy. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 6, 7, 13

[24] G. Jin, X. Yi, W. Huang, S. Schewe, and X. Huang. Enhancing adversarial training with second-order statistics of weights. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 6, 7, 13, 15

[25] H. Kannan, A. Kurakin, and I. Goodfellow. Adversarial logit pairing. *arXiv preprint, arXiv:1803.06373*, 2018. 3

[26] J. Kim and X. Wang. Sensible adversarial learning. In *OpenReview*, 2019. 2, 3, 6, 13

[27] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 2, 6

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012. 1, 16

[29] N. Kumari, M. Singh, A. Sinha, H. Machiraju, B. Krishnamurthy, and V. N Balasubramanian. Harnessing the vulnerability of latent layers in adversarially trained models. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019. 2, 6, 13

[30] K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 2

[31] Q. Li, Y. Guo, W. Zuo, and H. Chen. Squeeze training for adversarial robustness. In *International Conference on Learning Representations (ICLR)*, 2023. 2, 3, 13

[32] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[33] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, M. E. Houle, D. Song, and J. Bailey. Char-

acterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[34] C. Mao, Z. Zhong, J. Yang, C. Vondrick, and B. Ray. Metric learning for adversarial robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2, 3, 4, 6, 13

[35] M. Mccloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *The Psychology of Learning and Motivation*, 24:104–169, 1989. 5

[36] L. McInnes, J. Healy, N. Saul, and L. Grossberger. UMAP: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018. 8, 9

[37] A. Mądry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018. 2, 3, 6, 7, 13, 15, 16

[38] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[39] S. Moosavi-Dezfooli, A. Fawzi, J. Uesato, and P. Frossard. Robustness via curvature regularization, and vice versa. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[40] T. Pang, X. Yang, Y. Dong, H. Su, and J. Zhu. Bag of tricks for adversarial training. In *International Conference on Learning Representations (ICLR)*, 2021. 2, 3, 13

[41] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Asia Conference on Computer and Communications Security (CCS)*, 2017. 2

[42] R. Rade and S. Moosavi-Dezfooli. Helper-based adversarial training: Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *International Conference on Machine Learning (ICML) Workshop on Adversarial Machine Learning*, 2021. 3, 9

[43] A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, and P. Liang. Understanding and mitigating the tradeoff between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2020. 3, 9, 16

[44] V. V. Ramasesh, E. Dyer, and M. Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 5

[45] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[46] L. Rice, E. Wong, and Z. Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning (ICML)*, 2020. 3

[47] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 5

[48] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015. 2, 4

[49] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[50] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Mądry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 3

[51] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 1, 16

[52] C. Sitawarin, S. Chakraborty, and D. Wagner. Sat: Improving adversarial training via curriculum-based loss smoothing. In *ACM Workshop on Artificial Intelligence and Security (AISec)*, 2021. 2, 3, 6, 13

[53] J. Ström, K. Andersson, R. Sjöberg, F. Bossen, G. Sullivan, and J.-R. Ohm. Summary information on bd-rate experiment evaluation practices. *Joint Video Experts Team (JVET)-Q2016*, 2020. 2, 5, 9

[54] D. Stutz, M. Hein, and B. Schiele. Disentangling adversarial robustness and generalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 4, 8

[55] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 16

[56] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014. 1, 2

[57] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Mądry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*, 2019. 2

[58] H. Wang, Y. Deng, S. Yoo, H. Ling, and Y. Lin. AGKD-BML: defense against adversarial attack by attention guided knowledge distillation and bi-directional metric learning. In *International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 4, 5, 9, 13, 16

[59] J. Wang and H. Zhang. Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. In *International Conference on Computer Vision (ICCV)*, 2019. 2, 3, 13

[60] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin. Deep metric learning with angular loss. In *International Conference on Computer Vision (ICCV)*, 2017. 5, 6

[61] Y. Wang, X. Ma, J. Bailey, J. Yi, B. Zhou, and Q. Gu. On the convergence and robustness of adversarial training. In *International Conference on Machine Learning (ICML)*, 2019. 2, 3, 13

[62] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations (ICLR)*, 2020. 2, 3, 6, 13

[63] D. Wu, S. Xia, and Y. Wang. Adversarial weight perturbation helps robust generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2, 3, 6, 7, 13

[64] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le. Adversarial examples improve image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3, 4, 8

[65] W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017. 2

[66] S. Zagoruyko and N. Komodakis. Wide residual networks. In *British Machine Vision Conference (BMVC)*, 2016. 1, 6, 16

[67] S. Zagoruyko and N. Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations (ICLR)*, 2017. 9, 16

[68] H. Zhang and J. Wang. Defense against adversarial attacks using feature scattering-based adversarial training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2, 3, 13

[69] H. Zhang and W. Xu. Adversarial interpolation training: A simple approach for improving model robustness. In *OpenReview*, 2020. 2, 3, 13

[70] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019. 2, 3, 6, 13, 15

[71] J. Zhang, X. Xu, B. Han, G. Niu, L. Cui, M. Sugiyama, and M. Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International Conference on Machine Learning (ICML)*, 2020. 2, 3, 6, 13

[72] J. Zhu, J. Yao, B. Han, J. Zhang, T. Liu, G. Niu, J. Zhou, J. Xu, and H. Yang. Reliable adversarial distillation with unreliable teachers. In *International Conference on Learning Representations (ICLR)*, 2022. 2, 3, 13