Improving Adversarial Robustness by Mitigating Instability through Relearning

Yi Zeng[1], Ling Zhou[1], Ruilong Yu[1], Qihe Liu[⋆1], and Shijie Zhou[1]

University of Electronic Science and Technology of China
{yizeng312}@std.uestc.edu.cn,{lling.zhou,yrl666}@outlook.com,
{qiheliu,sjzhou}@uestc.edu.cn

Abstract. Adversarial training is the most effective defense against adversarial attacks, but it often suffers from robust overfitting, whose causes remain unclear. In this paper, we observe a counterintuitive phenomenon: the model robust stability consistently deteriorates during training. To quantify this behavior, we introduce the Robust Stability Rate (RSR) and define robust instability. Our analysis demonstrates that improving the Robust Stability Rate can effectively reduce robust instability, thereby mitigating robust overfitting. Building on this insight, we propose a novel framework called Relearning Adversarial Training (RAT). Our min-max optimization establishes a new adversarial dynamic where generated adversarial examples induce robust instability, while model training enforces robust stability, aiming to enhance the model Robust Stability Rate of the current learning model by leveraging knowledge from historical model. Experimental results demonstrate that RAT can be integrated with various methods to significantly enhance their performance. It effectively reduces robust instability and mitigates robust overfitting.

Keywords: Adversarial defense · Adversarial training · Overfitting

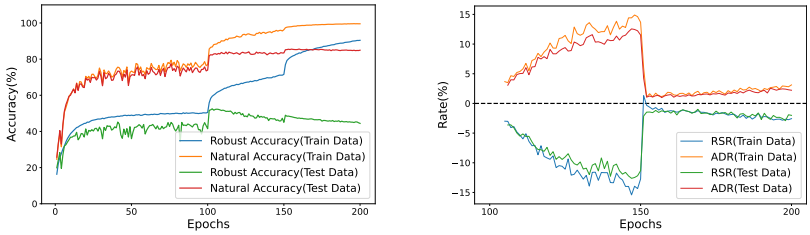
1 Introduction

Although deep neural networks are widely applied in various fields [2,17,28], [15] demonstrated that DNNs are highly vulnerable to adversarial attacks. To tackle this problem, [15] proposed adversarial training methods, while [25] regarded adversarial training as a min-max optimization problem. At present, adversarial training [15, 25] is one of the most effective approach to defend the threat of adversarial attacks [3,9], as it improves model robustness during training without requiring additional networks.

However, research has shown that adversarial training is prone to overfitting [31]. After the first learning rate decay, the robust accuracy of the model on the test data degrades as the number of training epochs increases (shown in Fig. 1a). The specific numerical meaning of robust overfitting can be expressed as:

$$H_{overfitting} = \mathrm{ACC}(f_{\theta_{t_{final}}}(\widehat{X}_{t_{final}}), Y) - \mathrm{ACC}(f_{\theta_{t_{best}}}(\widehat{X}_{t_{best}}), Y), \quad (1)$$

⋆ Corresponding author



(a) Robust Overfitting **(b)** Robust Instability

Fig. 1: Robust Overfitting(a) and Robust Instability(b) phenomenon on CIFAR-10 using ResNet-18. RSR (Robust Stability Rate): measure the robust accuracy of models on the same adversarial data across adjacent epochs. ADR (Attack Discrepancy Rate): measure the robust accuracy performance of the same model on adversarial data across adjacent epochs. Robust Overfitting is reflected in a continuous decline in robust accuracy during training, while Robust Instability is manifested as a negative RSR.

where $\widehat{X}_i$ denotes the adversarial examples generated based on the original data X after the i-th training epoch, θ_i represents the model weight after the i-th training epoch, t_{final} represents the final epoch, t_{best} represents the epoch when the highest robust accuracy is achieved, and ACC represents the model accuracy. The smaller the value of $H_{overfitting}$ is, the more severe the overfitting phenomenon has occurred.

Some traditional methods for mitigating robust overfitting, such as weight decay, data augmentation, explicit regularization and semi-supervised learning methods, either fail to prevent robust overfitting or perform worse than early stopping [31]. Currently, the most advanced adversarial training uses the additional data generated by the generative model to mitigate robust overfitting [4, 6, 16, 27, 30, 38], which further improves robust accuracy by increasing the quality and quantity of generated examples [33].

In this paper, we discover a counterintuitive phenomenon in adversarial training. It is typically assumed that the performance of model may improve after a full training round. However, we observe that after one round of learning, the model actually performs worse than in the previous round, even when using the same adversarial examples. This phenomenon represents a distinct challenge in adversarial training besides robust overfitting, which we define as **Robust Instability**:

$$H^{(i)}_{instability} = \mathrm{ACC}(f_{\theta_i}(\widehat{X}_i), Y) - \mathrm{ACC}(f_{\theta_{i-1}}(\widehat{X}_i), Y). \tag{2}$$

As shown in Fig. 1b, robust instability is specifically manifested as a low value of RSR. Further analysis suggests that robust instability reveals harmful decision boundary distortion and may be a contributing factor to overfitting. This implies that improving RSR could help mitigate robust overfitting.

Building on this insight, we propose a novel framework called Relearning Adversarial Training (RAT) which aims to obtain knowledge from historical models

to improve RSR. Experimental results show that RAT effectively alleviates robust instability and robust overfitting. Our main contributions are as follows:

- **Robust instability.** We identify and define a previously overlooked optimization issue in adversarial training, which progressively degrades model performance.
- **Novel framework.** We propose Relearning Adversarial Training (RAT), a novel framework that enhances the min-max optimization paradigm. RAT explicitly leverages knowledge from historical models to improve the Robust Stability Rate (RSR) of the current model, establishing a new dynamic between robust instability and stability.
- **Significant and plug-and-play improvement.** Our method effectively mitigates robust instability and robust overfitting, achieving state-of-the-art performance on benchmark datasets. Furthermore, the relearning strategy exhibits excellent plug-and-play capability, consistently and significantly boosting the robustness of various existing adversarial training methods.

2 Related Work

2.1 Adversarial Training

Adversarial training(AT) [15, 25, 44] is currently the most effective method for defending against adversarial attacks. The core idea of AT is to train a model f with weight θ, using adversarial examples generated from the original training data. [25] formalizes adversarial training as a min-max optimization problem:

$$\min_{\theta} \sum_{i} \max_{\widehat{x}_i} \mathcal{L}(f_\theta(\widehat{x}_i), y_i), \tag{3}$$

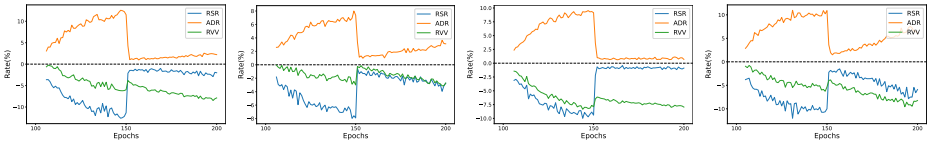
where $\widehat{x}_i = x_i + \Delta$ is defined as the adversarial example, Δ represents the adversarial perturbation constrained by the perturbation size ϵ, y_i is the ground-truth label corresponding to the original data x_i, $\mathcal{L}$ denotes loss function. The commonly employed technique to solve the inner maximization problem is Projected Gradient Descent (PGD):

$$\widehat{x}_i^{k+1} = x_i + clip(\widehat{x}_i^k + \alpha \cdot sign(\nabla_{\widehat{x}_i^k} \mathcal{L}(f_\theta(\widehat{x}_i^k), y_i)) - x_i, -\epsilon, \epsilon), \tag{4}$$

where $\widehat{x}_i^k$ is the adversarial example at iteration k, α is the step size of the perturbation, $sign$ function determines the direction of the generated perturbation, and $clip$ function constrains the perturbation magnitude within ϵ.

2.2 Robust Overfitting

Robust overfitting is a common issue in adversarial training. Although it has been thoroughly studied, there is still a lack of explanation for why it occurs [31]. Some approachs attempt to mitigate robust overfitting through methods such



(a) PGD cifar10 RN **(b)** TRADES cifar10 RN **(c)** PGD cifar100 RN **(d)** TRADES cifar10 WRN

Fig. 2: The results of RSR, ADR and RVV on test data under multiple scenes.

as knowledge distillation [14, 46, 47], weight perturbation [39, 42] and weight smoothing [7]. However, some of these methods perform even worse than early stopping [31]. Some studies have shown that adversarial training is prone to overfitting in the case of insufficient data [33]. Based on this, techniques such as data augmentation [19, 45] and diffusion model [35, 38] can solve robust overfitting. In addition, some studies have revealed that the cause of robust overfitting lies in the memorization of one-hot labels [12]. Moreover, research has shown that active forgetting and retraining strategies can mitigate overfitting [29].

3 Robust Instability

In this work, we identify a counterintuitive phenomenon in adversarial training, which we term Robust Instability. Through comprehensive evaluations in diverse datasets, model architectures, and adversarial training algorithms, we demonstrate the prevalence of this issue. In addition, we investigate the underlying causes of robust instability and explore potential strategies to mitigate it.

3.1 Definition of Robust Instability

We conduct adversarial training using the same experimental settings as [25]. To ensure reliable results, we employ multiple algorithms, datasets, and models, specifically (1) PGD-AT and TRADES, (2) CIFAR-10 and CIFAR-100, and (3) ResNet-18(RN) and WideResNet-28-10(WRN). We save the models at the end of each training epoch for our experiments. We introduce the Robust Stability Rate (RSR), a metric defined as follows to quantify robust instability by measuring the change in robust accuracy on the same adversarial data across adjacent epochs:

$$\mathrm{RSR}(i) = H^{(i)}_{instability} = \mathrm{ACC}(f_{\theta_i}(\widehat{X}_i), Y) - \mathrm{ACC}(f_{\theta_{i-1}}(\widehat{X}_i), Y). \tag{5}$$

To establish a correlation between robust instability and robust overfitting in subsequent analysis, we have defined an expression with a similar form to RSR, defining it as the adversarial Attack Discrepancy Rate(ADR):

$$\mathrm{ADR}(i) = \mathrm{ACC}(f_{\theta_{i-1}}(\widehat{X}_i), Y) - \mathrm{ACC}(f_{\theta_{i-1}}(\widehat{X}_{i-1}), Y). \tag{6}$$

Its specific meaning is to measure the robust accuracy performance of the same model on adversarial data across adjacent epochs.

Besides, We define the numerical representation of robust overfitting as the Robust Variation Value(RVV), specifically formulated as:

$$\mathrm{RVV}(i) = \mathrm{ACC}(f_{\theta_i}(\widehat{X}_i), Y) - \mathrm{ACC}(f_{\theta_{t_{best}}}(\widehat{X}_{t_{best}}), Y). \tag{7}$$

The lower RVV, the more serious robust overfitting occurs. When $i = t_{best}$, the value of RVV is zero, which explains why early stopping is consistently effective and completely avoids the occurrence of robust overfitting.

Robust overfitting typically emerges shortly after the first learning rate decay. To analyze this phenomenon, we plot the trajectories of RSR, ADR, and RVV from epoch 105 onward, as shown in Fig. 2. Our observations reveal three key patterns: (1) RSR exhibits a persistent downward trend, although it experiences a transient increase immediately following the second learning rate decay. This contradicts the conventional expectation of improving accuracy and instead indicates persistent model instability in adversarial training; (2) ADR maintains positive values with an upward trend after the first learning rate decay, followed by a brief decrease after the second decay. This sustained positivity is expected because $\widehat{X}_{i-1}$ is specifically optimized against θ_{i-1} and is thus more adversarial to it than $\widehat{X}_i$; and (3) RVV shows a consistent downward trend, reflecting progressively worsening robust overfitting throughout training.

3.2 Analysis of Robust Instability

Our analysis of robust instability reveals a harmful distortion of the decision boundary, enhancing robustness for some samples at the expense of others. Empirically, we observe a correlation between robust instability and overfitting: the trajectory of RVV closely mirrors the persistent decline of RSR (Fig. 2). This synchronized degradation leads us to hypothesize that the continuous accumulation of robust instability may be a potential driving mechanism behind overfitting. This insight motivates improving RSR could potentially alleviate overfitting.
Proposition 1: Instability Reveals Harmful Decision Boundary Distortion.
Let $f_\theta : \mathcal{X} \to \mathbb{R}^C$ be a classifier with loss function $\mathcal{L}(f_\theta(x), y)$ that is continuously differentiable with respect to the parameters θ, which are optimized using Stochastic Gradient Descent (SGD). Consider the i-th epoch of standard PGD-based adversarial training with the following definitions:
ϵ-perturbation ball: For a clean input $x \in \mathcal{X}$, define:

$$\mathcal{B}_\epsilon(x) := \{x' \in \mathcal{X} \mid \|x' - x\|_p \leq \epsilon\}, \tag{8}$$

where $\epsilon > 0$ is the perturbation magnitude and $p \in \{2, \infty\}$.
Adversarial example at the epoch $i-1$ and i:

$$\begin{aligned} \widehat{x}_{i-1} &= \arg\max_{x' \in \mathcal{B}_\epsilon(x)} \mathcal{L}(f_{\theta_{i-1}}(x'), y), \\ \widehat{x}_i &= \arg\max_{x' \in \mathcal{B}_\epsilon(x)} \mathcal{L}(f_{\theta_i}(x'), y). \end{aligned} \tag{9}$$

Parameter update rule:

$$\theta_i = \theta_{i-1} - \lambda \nabla_\theta \mathcal{L}(f_{\theta_{i-1}}(\widehat{x}_{i-1}), y), \tag{10}$$

where λ is the learning rate, usually a small positive value.

The results in Fig. 2 indicate that the model performs worse on the current adversarial example than the previous model, except in epochs adjacent to the second learning rate decay:

$$\mathrm{ACC}(f_{\theta_i}(\widehat{x}_i), y) - \mathrm{ACC}(f_{\theta_{i-1}}(\widehat{x}_i), y) < 0. \tag{11}$$

Under such conditions, the decision boundary of f_θ undergoes harmful distortion over $\mathcal{B}_\epsilon(x)$: there exist two perturbation directions $\widehat{x}_{i-1}, \widehat{x}_i \in \mathcal{B}_\epsilon(x)$ such that during the update $\theta_{i-1} \to \theta_i$, robustness improves along $\widehat{x}_{i-1}$ but degrades along $\widehat{x}_i$.

Proof. We analyze the loss change on two fixed inputs: $\widehat{x}_{i-1}$ and $\widehat{x}_i$.

Step 1 (theoretical derivation): Loss decreases on $\widehat{x}_{i-1}$.

Define $h(\theta) := \mathcal{L}(f_\theta(\widehat{x}_{i-1}), Y)$. Since $\mathcal{L}$ is continuously differentiable in θ, the first-order Taylor expansion of h around θ_{i-1} gives:

$$h(\theta_i) = h(\theta_{i-1}) + \nabla_\theta h(\theta_{i-1})^\top (\theta_i - \theta_{i-1}) + o(\|\theta_i - \theta_{i-1}\|). \tag{12}$$

Substituting the PGD update θ by Eq.(10), we obtain:

$$\mathcal{L}(f_{\theta_i}(\widehat{x}_{i-1}), y) = \mathcal{L}(f_{\theta_{i-1}}(\widehat{x}_{i-1}), y) - \lambda \left\| \nabla_\theta \mathcal{L}(f_{\theta_{i-1}}(\widehat{x}_{i-1}), y) \right\|^2 + o(\eta). \tag{13}$$

Since the squared gradient norm is non-negative and strictly positive at non-stationary points, there exists $\lambda_0 > 0$ such that for all $0 < \lambda < \lambda_0$:

$$\mathcal{L}(f_{\theta_i}(\widehat{x}_{i-1}), y) < \mathcal{L}(f_{\theta_{i-1}}(\widehat{x}_{i-1}), y). \tag{14}$$

Step 2 (empirical observation): Loss increases on $\widehat{x}_i$.

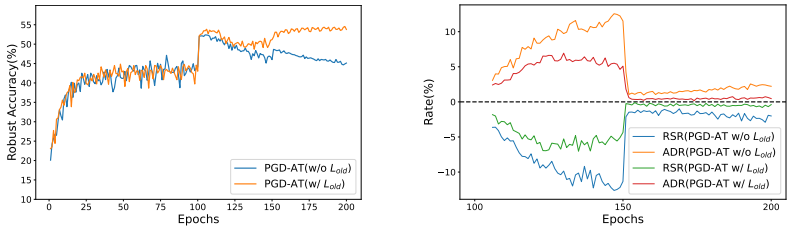
By Eq. (11) and the monotonic inverse relationship between accuracy and loss, we obtain the following:

$$\mathcal{L}(f_{\theta_i}(\widehat{x}_i), y) > \mathcal{L}(f_{\theta_{i-1}}(\widehat{x}_i), y). \tag{15}$$

The experiments in the Appendix C show that the tension described by Eq. (14) and Eq. (15) arises in almost all training epoch.

Since both $\widehat{x}_{i-1}$ and $\widehat{x}_i$ lie within the same ϵ-ball $\mathcal{B}_\epsilon(x)$, Eq. (14) and Eq. (15) indicate that the model's robustness varies non-uniformly over $\mathcal{B}_\epsilon(x)$. Specifically, it improves in one direction while deteriorating in another, leading to a harmful distortion of the decision boundary. This observation aligns with the conclusion of [41], which highlights the conflicting dynamics involved in the movement of the decision boundary. Therefore, a smooth change rather than a catastrophic reconstruction of the decision boundary should be considered.

Hypothesis 1: Robust Instability May Be a Contributing Factor to Overfitting.



(a) Robust Overfitting **(b)** Robust Instability

Fig. 3: (a) The robust accuracy of PGD-AT and PGD-AT (w/ $\mathcal{L}_{old}$) on test data of CIFAR-10. (b) The curves of RSR(%) and ADR(%) in PGD-AT and PGD-AT (w/ $\mathcal{L}_{old}$) on CIFAR-10 after the first learning rate decay on test data.

We further subdivide RVV according to training epochs to study Robust Variation Values at a micro level. After the division, it is defined as follows:

$$\mathrm{RVV}(i)\downarrow = \sum_{j=t_{best}+1}^{i} \mathrm{RSR}(j)\downarrow + \sum_{j=t_{best}+1}^{i} \mathrm{ADR}(j)\uparrow . \tag{16}$$

Observations in Section 3.1 indicate that $\mathrm{RSR}(i)$ is almost consistently negative and exhibits a persistent downward trend, whereas $\mathrm{ADR}(i)$ remains consistently positive with a continuous upward trend. Meanwhile, $\mathrm{RVV}(i)$ shows an overall declining tendency. When considered alongside Eq. (16), this suggests a potential correlation between the cumulative effect of RSR and the observed decline in RVV. We analyzed the sum of RSR accumulation and RVV in the Appendix C and found that the two trajectories exhibit a consistent trend. This empirical observation leads us to hypothesize that the persistent decline in RVV is closely associated with the continuous deterioration of RSR, which implies that the accumulation of robust instability may be a contributing factor to overfitting. This insight motivates improving RSR could potentially alleviate overfitting.

3.3 Handling of Robust Instability

Based on the analysis of robust instability and to verify the hypothesis, we incorporate Eq. (5) into the PGD-based adversarial training minimization objective to facilitate training smoothness and enhance RSR:

$$\mathcal{L}_{old}(\theta_i;\theta_{i-1};\widehat{X}) = \mathcal{L}_{mse}(f_{\theta_i}(\widehat{X}), f_{\theta_{i-1}}(\widehat{X})). \tag{17}$$

As demonstrated in the Appendix D, this approach is theoretically effective: by penalizing large shifts in the model's outputs, it implicitly limits the magnitude of weight updates across training epochs, thereby enhancing RSR. The experimental results are shown in Fig. 3. Specifically, Fig. 3a demonstrates that this strategy helps alleviate the problem of robust overfitting, while Fig. 3b shows that leveraging the knowledge of historical models can effectively increase RSR and mitigate robust instability.

4 Relearning Adversarial Training

Based on the verification experimental results in Section 3.3, we propose an adversarial training framework called Relearning Adversarial Training (RAT), which enables the current learning model to acquire knowledge from historical models to enhance RSR. The algorithm framework is shown in Fig. 4. The Relearning method implemented based on PGD-AT is shown in Algorithm 1. More implementation details can be found in the Appendix G.

Relearning Labels. Although $\mathcal{L}_{old}$ is effective, it is only based on a single model. If a model performs poorly in one epoch, subsequent models may inherit this weakness, leading to a lower performance of the final model. The experimental results in the Appendix E indicate that the current model can learn from asynchronous historical models to increase RSR. This suggests that leveraging predictions from multiple historical models may provide a more stable teaching signal, mitigating the limitations of single-model supervision. However, storing all historical models and involving them in gradient computation is impractical.

To address this, we adopt a Temporal Ensembling strategy [22] that approximates multi-model integration at negligible cost. Concretely, we maintain a relearning label p_i for each sample x_i and updated recursively as:

$$p_i \leftarrow \eta \cdot p_i + (1-\eta) \cdot q_i, \tag{18}$$

where η is the momentum term and q_i is calculated using the output of the current model f_θ. After t training epochs, unrolling the recursive update yields:

$$p_i = (1-\eta) \sum_{j=1}^{t} \eta^{t-j} q_i^j + \eta^t \tilde{y}_i, \tag{19}$$

where q_i^j denotes the value of q_i computed from the output of the model at the j-th epoch, and $\tilde{y}_i$ is the one-hot encoding of the ground-truth label y_i, which serves as the initialization of the recursion. This expansion reveals that p_i aggregates all previous historical outputs q_i^j with exponentially decaying weights, effectively forming an ensemble of historical models. In the Appendix F, we prove that learning from p_i enhances RSR.

The auxiliary term q_i combines the outputs of the current model f_θ on clean and adversarial examples:

$$q_i = \gamma \cdot f_\theta(x_i) + (1-\gamma) \cdot f_\theta(\widehat{x}_i), \tag{20}$$

where γ is the balance factor. As training progresses, the model's predictions on clean examples tend to become one-hot vectors. A large γ causes p_i to collapse towards a one-hot hard label; a small γ leads to overfitting to adversarial examples and deviation from the clean distribution.

Relearning Loss. This loss combines the cross-entropy loss and MSE loss using a relearning weight ω, which is used to learn from data $\widehat{x}_i$ and obtain knowledge from the relearning label p_i, defined as:

$$\mathcal{L}_{re}(\theta; \widehat{x}_i; y_i; p_i; \omega) = \mathcal{L}_{ce}(f_\theta(\widehat{x}_i), y_i) + \omega \cdot \mathcal{L}_{mse}(f_\theta(\widehat{x}_i), p_i). \tag{21}$$

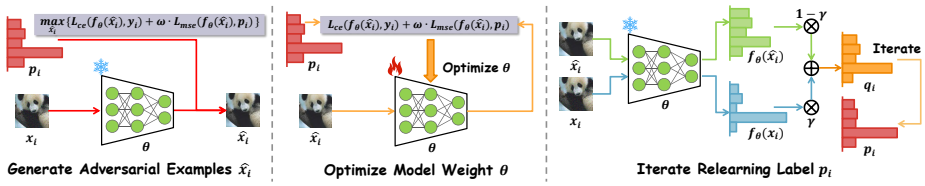


Fig. 4: Schematics of the proposed Relearning Adversarial Training prototype. **Step 1**: Generate Adversarial Examples. An additional MSE loss is incorporated during the generation of adversarial examples. **Step 2**: Optimize Model Weight. The optimization strategy for adversarial training is reconstructed based on the Relearning loss. **Step 3**: Iterate Relearning Label. After optimizing model weight, Relearning labels are iterated based on the output results of the current model. For the next training epoch, the knowledge contained in p_i comes from historical models.

Algorithm 1 Realizing of PGD-AT+Relearning

Require: optimizer $\mathcal{O}$, model f with model weight θ, relearning loss $\mathcal{L}_{re}$, training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, total epochs N, relearning label p_i, attack steps K, perturbation size ϵ, step size α, ensembling momentum η, relearning weight ω.

Initialize the relearning label as the one-hot encoding of the ground-truth label
for $t = 1$ **to** N **do**
  for each mini-batch (x_i, y_i) **in** $\mathcal{D}$ **do**
    $\widehat{x}_i^0 = x_i + \epsilon^{rdm}, \epsilon^{rdm} \sim Uniform(-\epsilon, \epsilon)$
    for $k = 0$ **to** $K-1$ **do**
      $\widehat{x}_i^{k+1} = x_i + clip(\widehat{x}_i^k + \alpha \cdot sign(\nabla_{\widehat{x}_i^k}\{\mathcal{L}_{ce}(f_\theta(\widehat{x}_i^k), y_i) + \omega \cdot \mathcal{L}_{mse}(f_\theta(\widehat{x}_i^k), p_i)\}) - x_i, -\epsilon, \epsilon)$
    end for
    $\mathcal{L}_{re}(\theta; \widehat{x}_i; y_i; p_i; \omega) = \mathcal{L}_{ce}(f_\theta(\widehat{x}_i), y_i) + \omega \cdot \mathcal{L}_{mse}(f_\theta(\widehat{x}_i), p_i)$
    Update θ according to $\mathcal{L}_{re}$ by $\mathcal{O}$
    $p_i \leftarrow \eta \cdot p_i + (1-\eta) \cdot \{\gamma \cdot f_\theta(x_i) + (1-\gamma) \cdot f_\theta(\widehat{x}_i)\}$
  end for
end for
return trained model weight θ

Generation of Adversarial Examples. Our goal is to generate adversarial examples that not only cause misclassification but also undermine the stability of the model. To achieve this, we augment the standard adversarial loss with an MSE term that induces a deliberate divergence between the model's output and the relearning labels, resulting in the following modified generation function:

$$\widehat{x}_i^{k+1} = x_i + clip(\widehat{x}_i^k + \alpha \cdot sign(\nabla_{\widehat{x}_i^k}\mathcal{L}_{re}(\theta; \widehat{x}_i^k; y_i; p_i; \omega)) - x_i, -\epsilon, \epsilon). \tag{22}$$

New min-max Optimization. Our min-max optimization establishes an adversarial dynamic where generated adversarial examples induce robust instability, while model training enforces robust stability. The relearning training strategy will be applied shortly before the learning rate drops. Prior to that, the original adversarial training method will be used to ensure that the model has

certain basic capabilities. The specific redefined min-max optimization problem is as follows:

$$\min_{\theta} \sum_{i} \max_{\widehat{x}_i} \{\mathcal{L}_{ce}(f_\theta(\widehat{x}_i), y_i) + \omega \cdot \mathcal{L}_{mse}(f_\theta(\widehat{x}_i), p_i)\}. \tag{23}$$

5 Experiment

5.1 Experimental Settings

Datasets. We evaluate Relearning method on CIFAR-10 [21], CIFAR-100 [21] and SVHN [26]. Besides, we also use the more challenging datasets STL-10 [8] and Tiny-ImageNet-200 [11] in some experiments.

Attack Methods. We primarily employ PGD-20 and Auto Attack [10] to evaluate the performance of adversarial training. We use standard version of AA, including APGD-CE [10], APGD-DLR [10], FAB [10], and Square Attack [1], as it is considered the most reliable method for robustness assessment. In some experiments, we use FGSM [15], Square Attack [1] and CW [5]. The experiments in the main text are all based on the L_∞ threat model, and the results of the L_2 threat model can be found in the Appendix H.

Architectures. We select ResNet-18 [18] and WideResNet-34-10 [43] as the network architectures and we use VIT [13], VGG-16 [34] and MobileNetV2 [32] for partial experiments. During training, the model is trained using the SGD optimizer with a momentum of 0.9, weight decay of 5×10^{-4}, an initial learning rate of 0.1 and a total training epoch of 200. The learning rate is divided by 10 at 100-th and 150-th epoch. The hyperparameters γ, η and ω are set to 0.5, 0.9 and 30, respectively. During training, a 10-step adversarial attack is employed with a step size of $2/255$ and a perturbation magnitude of $8/255$. More training details can be found in Appendix H.1.

Metrics. Alongside attack metrics for evaluating robustness, we also use two additional metrics: RSR and trade-off. The trade-off metric as defined by [29] assesses how effectively a method balances natural accuracy and robust accuracy. The calculation methods for two indicators can be found in the Appendix H.1.

5.2 Performance Evaluation and Analysis

We study the performance of Relearning method to mitigate model instability and robust overfitting. We select the baseline method PGD-AT and compare it with the TE method. We evaluate the performance on CIFAR-10 using PGD-20 and AA, with the experimental results presented in Tab. 1. Compared with TE, our method has better model performance, better ability to mitigate robust overfitting and enhance robust stability. More experimental results for different datasets(CIFAR-100, SVHN, STL-10, TinyImageNet-200), architectures(VGG-16, MobileNetV2) and metrics(FGSM, Square Attack) are in the Appendix H.2.

Table 1: Test accuracy (%) of PGD-AT, PGD-AT+TE, PGD-AT+Relearning on CIFAR-10 using ResNet-18 and VIT under the (L_∞, $\epsilon = 8/255$) threat model evaluated by PGD-20, Auto Attack and RSR. Our experimental results are averaged over five rounds.

Method	ResNet-18							VIT						
	PGD-20			Auto Attack			RSR	PGD-20			Auto Attack			RSR
	Best	Final	Diff.	Best	Final	Diff.		Best	Final	Diff.	Best	Final	Diff.	
PGD-AT	50.70	41.40	-9.30	47.62	40.74	-6.88	-88.61	47.34	38.78	-8.56	44.09	38.11	-5.98	-70.74
PGD-AT+TE	55.49	53.87	-1.62	50.32	49.45	-0.87	-29.48	47.22	42.76	-4.46	43.98	40.97	-3.01	-52.74
PGD-AT+RE	**56.02**	**55.40**	**-0.62**	**51.22**	**50.83**	**-0.39**	**-22.80**	**47.76**	**44.55**	**-3.21**	**44.22**	**41.75**	**-2.47**	**-44.71**

Table 2: Test accuracy (%) of baseline method and Relearning method on CIFAR-10 and CIFAR-100 using ResNet-18 under the (L_∞, $\epsilon = 8/255$) threat model. Our experimental results are averaged over five rounds.

Method	CIFAR-10						CIFAR-100					
	Natural	PGD	Square	CW	AA	RSR	Natural	PGD	Square	CW	AA	RSR
PGD-AT	82.40	41.40	59.97	40.98	40.74	-88.61	58.08	20.81	31.14	21.15	19.69	-44.17
PGD-AT+RE	**82.50**	**55.40**	**64.29**	**52.75**	**50.83**	**-22.80**	**58.36**	**31.39**	**35.45**	**27.99**	**26.00**	**-20.66**
TRADES	82.42	50.10	62.37	48.88	47.29	-100.90	56.35	27.19	33.02	24.61	23.73	-56.95
TRADES+RE	**82.58**	**54.66**	**63.61**	**51.29**	**49.76**	**-32.09**	**58.35**	**30.92**	**35.10**	**26.68**	**25.42**	**-44.53**
MART	82.18	48.45	60.13	45.81	43.61	-104.30	55.19	24.13	30.93	22.37	21.00	-56.33
MART+RE	**82.30**	**54.66**	**62.10**	**50.75**	**49.30**	**-29.42**	**55.65**	**31.05**	**34.12**	**26.76**	**25.65**	**-29.37**
AT-AWP	81.15	55.03	**63.93**	51.67	49.45	-45.36	53.89	30.21	**34.80**	27.25	25.30	-15.79
AT-AWP+RE	**81.18**	**57.23**	63.19	**52.74**	**51.23**	**-31.75**	**54.88**	**32.11**	34.61	**27.75**	**25.87**	**-12.40**

5.3 Ability to Combine with Other Methods

We evaluate Relearning by integrating it with four baseline methods: PGD-AT [25], TRADES [44], MART [37], and AT-AWP [39]. Performance on CIFAR-10 and CIFAR-100 using ResNet-18 of the Relearning strategy are presented in Tab. 2. We observe that Relearning not only improves almost both natural and robust accuracy against various attacks but also mitigate robust instability, confirming the overall effectiveness and powerful integration capability of our strategy. More implementation details and experimental results can be found in the Appendix H.3.

5.4 Comparison with Related Work

We evaluate Relearning against a comprehensive set of competitors, categorized as follows: (1) Standard baselines (PGD-AT [25], TRADES [44], and AWP [39]); (2) Knowledge distillation-inspired methods (PGD-AT+TE [12], KD-SWA [7], and FOMO [29]), which are most relevant to our soft-label approach; (3) Exponential moving average(EMA) method (SEAT [36]) and (4) State-of-the-art

Table 3: Test accuracy (%) of the proposed methods and other methods on CIFAR-10 using ResNet-18 and ResNet-34-10 under the (L_∞, $\epsilon = 8/255$) threat model evaluated by PGD-20. Our experimental results are averaged over five rounds.

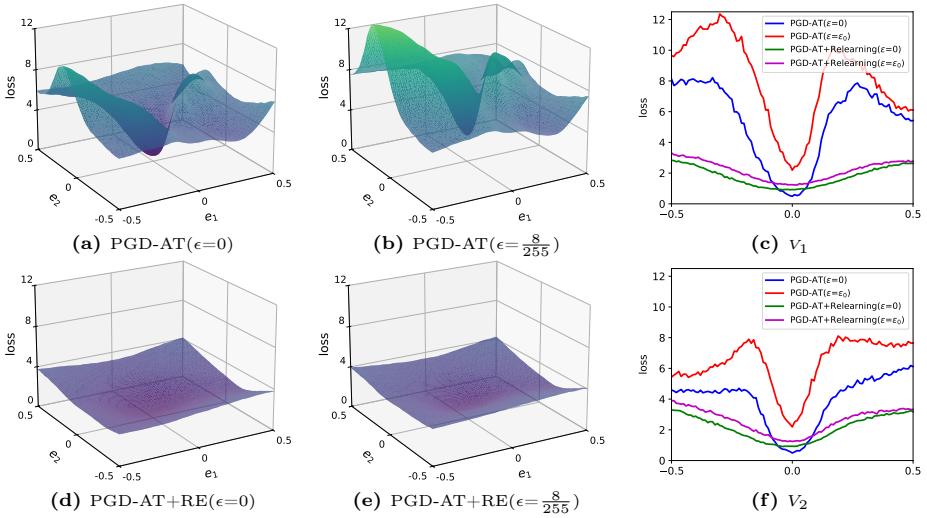
Method	ResNet-18							WideResNet-34-10						
	Natural Accuracy			Robust Accuracy			Trade.	Natural Accuracy			Robust Accuracy			Trade.
	Best	Final	Diff.	Best	Final	Diff.		Best	Final	Diff.	Best	Final	Diff.	
PGD-AT	80.70	82.40	1.70	50.70	41.40	-9.30	55.11	85.79	85.15	-0.64	54.83	46.42	-8.41	60.08
TRADES	81.12	82.42	1.30	52.87	50.10	-2.77	62.32	84.58	85.19	0.61	55.60	48.26	-7.34	61.62
PGD-AT+TE	82.44	83.06	0.62	55.49	53.87	-1.62	65.35	85.80	85.20	-0.60	56.79	53.51	-3.28	65.74
KD+SWA	**84.06**	84.48	0.42	53.70	53.68	**-0.02**	65.65	86.85	**88.03**	1.18	56.92	55.74	-1.18	68.26
AT-AWP	81.11	81.15	0.04	55.60	55.03	-0.57	65.59	85.21	85.65	0.44	58.53	57.81	-0.72	69.03
SEAT	82.48	**85.63**	**3.15**	55.14	47.44	-7.70	61.05	86.21	87.59	1.38	59.18	53.66	-5.52	66.55
ADR	82.41	85.08	2.67	54.98	51.38	-3.60	64.07	84.57	86.37	**1.80**	57.36	49.81	-7.55	63.18
HFAT	81.94	82.59	0.65	56.95	55.74	-1.21	66.56	85.56	87.22	1.66	59.33	54.74	-4.59	67.26
FOMO	81.84	82.51	0.67	56.68	56.46	-0.22	67.04	**87.31**	87.08	-0.23	59.69	59.23	-0.46	70.50
PGD-AT+RE	82.01	82.50	0.49	56.02	55.40	-0.62	66.29	85.01	85.30	0.29	56.63	54.37	-2.26	66.41
AT-AWP+RE	80.73	81.18	0.45	**57.31**	**57.23**	-0.08	**67.13**	85.62	86.17	0.45	**61.32**	**61.21**	**-0.11**	**71.58**

methods (PGD-AT+ADR [40] and HFAT [23]). In contrast to other knowledge distillation-inspired methods, we adopt historical models as teachers instead of pre-trained ones, and our min-max optimization establishes a novel adversarial dynamic whereby the generated adversarial examples induce robust instability, while model training enhances robust stability.

The results on CIFAR-10 using ResNet-18 and ResNet-34-10 evaluated by PGD-20 are shown in Tab. 3. Experiments under early-stopping(Best) and full-training(Final) scenarios reveal that while some advanced methods excel initially, they suffer from significant performance degradation later(lower Diff.), indicating robust overfitting. In contrast, our method achieves the highest robust accuracy in both scenarios with a negligible drop, effectively mitigating overfitting. Moreover, it improves the natural accuracy of the baseline method, attaining the best overall trade-off between clean and robust accuracy(higher Trade.). Further analysis and results are provided in the Appendix H.4.

5.5 Other Experiments using Relearning

Investigate Mechanism of Relearning. We investigate the mechanism of Relearning by analyzing the decision boundary with two novel metrics. Remain (accuracy on adversarial examples consistently correct across historical and current models) quantifies the stable inheritance of robust knowledge, while Forget (accuracy on adversarial examples forgotten by the current model) signals destructive boundary shifts. Results shown in Tab. 14(see Appendix H.5) demonstrate that Relearning increases Remain and reduces Forget, which supports Relearning imposes a mild constraint on the decision boundary, encouraging smooth evolution rather than disruptive reconstruction, thereby stabilizing training.



(a) PGD-AT(ϵ=0) **(b)** PGD-AT($\epsilon=\frac{8}{255}$) **(c)** v_1

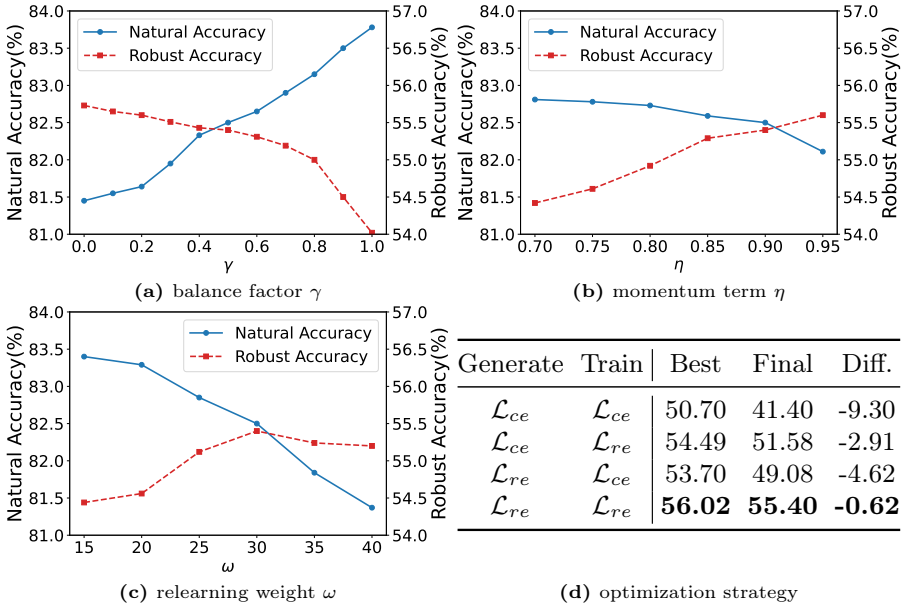
(d) PGD-AT+RE(ϵ=0) **(e)** PGD-AT+RE($\epsilon=\frac{8}{255}$) **(f)** v_2

Fig. 5: The loss landscape $\mathcal{L}_{ce}(\theta + v_1 a_1 + v_2 a_2)$ of PGD-AT and PGD-AT+Relearning using ResNet-18 on CIFAR-10. θ, v_1, v_2 are weight, the first and second unit eigenvectors of the Hessian matrix. (a)(b)(d)(e) depict the 3D loss landscapes under different scenarios, while (c)(f) provide the corresponding 2D cross-sections along two directions.

Effect of Addtional Data. Our Relearning method achieve SOTA performance on CIFAR-10 using WideResNet-28-10 under 1M additional data, with robust accuracy of 67.44% on PGD-40 and 64.70% on Auto Attack. The Relearning method achieves significant performance gains with only a small number of additional data. The complete experimental details and results can be found in the Appendix H.6.

Research on the Sustained Ability in Mitigating Robust Overfitting. We increase epoch to 300 to validate the sustained ability in mitigating robust overfitting (see Appendix H.7). Our Relearning methods show better performance in this regard. On CIFAR-10 and CIFAR-100, the robust accuracy of the TE method decreased by 3.62% and 2.66%, respectively, while the Relearning method only decreased by 2.07% and 0.92%.

Visualization Results. We evaluate the effectiveness of Relearning by visualizing the Loss Landscape, Centered Kernel Alignment [20], and Weight Drift using ResNet-18. The visualization results of the loss landscape are presented in Fig. 5, with full results available in the Appendix H.8. Following the setup in [24], we select the first and second unit eigenvectors of the Hessian matrix as the perturbation directions. Compared to PGD-AT, Relearning exhibits a flatter loss landscape both with and without input perturbations, which illustrates that Relearning provides better robustness.



(a) balance factor γ **(b)** momentum term η

(c) relearning weight ω

Generate	Train	Best	Final	Diff.
$\mathcal{L}_{ce}$	$\mathcal{L}_{ce}$	50.70	41.40	-9.30
$\mathcal{L}_{ce}$	$\mathcal{L}_{re}$	54.49	51.58	-2.91
$\mathcal{L}_{re}$	$\mathcal{L}_{ce}$	53.70	49.08	-4.62
$\mathcal{L}_{re}$	$\mathcal{L}_{re}$	**56.02**	**55.40**	**-0.62**

(d) optimization strategy

Fig. 6: The ablation results on CIFAR-10 using ResNet-18.

5.6 Ablation Studies

In this section, we conduct ablation experiments using ResNet-18 architecture under the L_∞ threat model on CIFAR-10. More experimental details and results are provided in the Appendix H.9.

Hyperparameters γ, η, ω. We set the baseline hyperparameters γ to 0.5, η to 0.9, and ω to 30 for the ablation experiment to study the impact of each hyperparameter on model performance. We vary γ from 0 to 1 and η from 0.7 to 0.95. The results in Fig. 6a and 6b indicate that higher values of γ and η improve natural accuracy but reduce robust accuracy. A large γ causes the relearning label to approach a one-hot label, whereas a small γ leads to overfitting to the distribution over adversarial examples. When η is small, the relearning label becomes unstable and overly susceptible to the poor performance of a single epoch's model, ultimately leading to degraded robust accuracy. Therefore, we choose $\gamma = 0.5$ and $\eta = 0.9$ as a compromise. Next, we vary ω from 15 to 40. The results in Fig. 6c show that as ω increases, natural accuracy decreases, while robust accuracy peaks at $\omega = 30$. If ω is too high, the model excessively relies on historical knowledge and fails to acquire new knowledge, leading to a decline in both natural and robust accuracy. Ultimately, we set ω to 30. More results are provided in the Appendix H.9.

Min-max Optimization Strategy. As demonstrated in Fig. 6d, employing relearning loss in either the adversarial example generation phase or the training phase consistently enhances both the best and final robust accuracy while effectively mitigating robust overfitting. The performance degradation between best

and final epochs is substantially reduced compared to conventional approaches using cross-entropy loss. Most notably, when relearning loss is simultaneously adopted in both generation and training phases, the model achieves optimal performance, attaining the highest robust accuracy with minimal overfitting. This comprehensive improvement validates the effectiveness of relearning in enhancing adversarial robustness and training stability. Additionally, we conducted experiments to verify that the MSE loss is more suitable than the KL divergence as a component of the relearning loss (see Appendix H.9).

Computation Efficiency. We compare proposed methods with baseline methods in terms of runtime and GPU memory usage. To ensure fair comparison, all methods are integrated into a universal training framework. The results (see Appendix H.9) indicate that our Relearning method incurs a small amount of additional computational overhead and negligible extra space consumption.

Stability Study. We conduct two experiments to validate the stability of our algorithm: one tests robust accuracy under varying perturbation levels, and the other assesses mean and standard deviation across different random seeds. The results (see Appendix H.9) show that our method consistently outperforms baseline adversarial techniques at all perturbation levels and maintains strong stability across different random seed conditions.

6 Conclusion

This work identifies and investigates the phenomenon of robust instability in adversarial training. Our analysis suggests that robust instability reveals harmful decision boundary distortion and may be a contributing factor to overfitting. This implies that improving RSR could help mitigate robust overfitting. Building on this insight, we introduce the Relearning Adversarial Training (RAT) framework, which mitigates robust instability and overfitting by leveraging knowledge from historical models without any pre-trained teacher model. This approach reformulates the min-max optimization process as a new adversarial dynamic, where generated examples induce robust instability, while model training enforces robust stability. Extensive experimental results confirm that our relearning strategy is compatible with existing methods and achieves superior performance, effectively mitigating robust instability and robust overfitting.

7 Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant (No.62272089), and Key Science and Technology Special Projects (No.2023YFG0373).

References

1. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search. In: European Conference on Computer Vision. pp. 484–501. Springer (2020)

2. Awais, M., Naseer, M., Khan, S., Anwer, R.M., Cholakkal, H., Shah, M., Yang, M.H., Khan, F.S.: Foundation models defining a new era in vision: a survey and outlook. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)
3. Bai, T., Luo, J., Zhao, J., Wen, B., Wang, Q.: Recent advances in adversarial training for adversarial robustness. arXiv preprint arXiv:2102.01356 (2021)
4. Bartoldson, B.R., Diffenderfer, J., Parasyris, K., Kailkhura, B.: Adversarial robustness limits via scaling-law and human-alignment studies. arXiv preprint arXiv:2404.09349 (2024)
5. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. Ieee (2017)
6. Chen, E.C., Lee, C.R.: Data filtering for efficient adversarial training. Pattern Recognition **151**, 110394 (2024)
7. Chen, T., Zhang, Z., Liu, S., Chang, S., Wang, Z.: Robust overfitting may be mitigated by properly learned smoothening. In: International Conference on Learning Representations (2020)
8. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 215–223. JMLR Workshop and Conference Proceedings (2011)
9. Croce, F., Gowal, S., Brunner, T., Shelhamer, E., Hein, M., Cemgil, T.: Evaluating the adversarial robustness of adaptive test-time defenses. In: International Conference on Machine Learning. pp. 4421–4435. PMLR (2022)
10. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: International conference on machine learning. pp. 2206–2216. PMLR (2020)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
12. Dong, Y., Xu, K., Yang, X., Pang, T., Deng, Z., Su, H., Zhu, J.: Exploring memorization in adversarial training. In: International Conference on Learning Representations (2022)
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
14. Goldblum, M., Fowl, L., Feizi, S., Goldstein, T.: Adversarially robust distillation. In: Proceedings of the AAAI conference on artificial intelligence. pp. 3996–4003 (2020)
15. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
16. Gowal, S., Rebuffi, S.A., Wiles, O., Stimberg, F., Calian, D.A., Mann, T.A.: Improving robustness using generated data. Advances in Neural Information Processing Systems **34**, 4218–4233 (2021)
17. He, C., Shen, Y., Fang, C., Xiao, F., Tang, L., Zhang, Y., Zuo, W., Guo, Z., Li, X.: Diffusion models in low-level vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)

19. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple data processing method to improve robustness and uncertainty. arXiv preprint arXiv:1912.02781 (2019)
20. Kornblith, S., Norouzi, M., Lee, H., Hinton, G.: Similarity of neural network representations revisited. In: International conference on machine learning. pp. 3519–3529. PMlR (2019)
21. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
22. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. In: International Conference on Learning Representations (2017)
23. Li, Q., Hu, Y., Dong, Y., Zhang, D., Chen, Y.: Focus on hiders: Exploring hidden threats for enhancing adversarial training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24442–24451 (2024)
24. Liu, C., Salzmann, M., Lin, T., Tomioka, R., Süsstrunk, S.: On the loss landscape of adversarial training: Identifying challenges and how to overcome them. Advances in Neural Information Processing Systems **33**, 21476–21487 (2020)
25. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
26. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y., et al.: Reading digits in natural images with unsupervised feature learning. In: NIPS workshop on Deep Learning and Unsupervised Feature Learning. vol. 2011, p. 4. Granada (2011)
27. Pang, T., Lin, M., Yang, X., Zhu, J., Yan, S.: Robustness and accuracy could be reconcilable by (proper) definition. In: International Conference on Machine Learning. pp. 17258–17277. PMLR (2022)
28. Parraga, O., More, M.D., Oliveira, C.M., Gavenski, N.S., Kupssinskü, L.S., Medronha, A., Moura, L.V., Simões, G.S., Barros, R.C.: Fairness in deep learning: A survey on vision and language research. ACM Computing Surveys **57**(6), 1–40 (2025)
29. Ramkumar, V.R.T., Zonooz, B., Arani, E.: The effectiveness of random forgetting for robust generalization. arXiv preprint arXiv:2402.11733 (2024)
30. Rebuffi, S.A., Gowal, S., Calian, D.A., Stimberg, F., Wiles, O., Mann, T.: Fixing data augmentation to improve adversarial robustness. arXiv preprint arXiv:2103.01946 (2021)
31. Rice, L., Wong, E., Kolter, Z.: Overfitting in adversarially robust deep learning. In: International Conference on Machine Learning. pp. 8093–8104. PMLR (2020)
32. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
33. Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., Madry, A.: Adversarially robust generalization requires more data. Advances in Neural Information Processing Systems **31** (2018)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
35. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International Conference on Machine Learning. pp. 2256–2265. PMLR (2015)
36. Wang, H., Wang, Y.: Self-ensemble adversarial training for improved robustness. arXiv preprint arXiv:2203.09678 (2022)

37. Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., Gu, Q.: Improving adversarial robustness requires revisiting misclassified examples. In: International conference on learning representations (2019)
38. Wang, Z., Pang, T., Du, C., Lin, M., Liu, W., Yan, S.: Better diffusion models further improve adversarial training. In: International Conference on Machine Learning. pp. 36246–36263. PMLR (2023)
39. Wu, D., Xia, S.T., Wang, Y.: Adversarial weight perturbation helps robust generalization. Advances in Neural Information Processing Systems **33**, 2958–2969 (2020)
40. Wu, Y.Y., Wang, H.J., Chen, S.T.: Annealing self-distillation rectification improves adversarial training. arXiv preprint arXiv:2305.12118 (2023)
41. Xu, Y., Sun, Y., Goldblum, M., Goldstein, T., Huang, F.: Exploring and exploiting decision boundary dynamics for adversarial robustness. In: The Eleventh International Conference on Learning Representations (2023)
42. Yu, C., Han, B., Gong, M., Shen, L., Ge, S., Du, B., Liu, T.: Robust weight perturbation for adversarial training. arXiv preprint arXiv:2205.14826 (2022)
43. Zagoruyko, S.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
44. Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., Jordan, M.: Theoretically principled trade-off between robustness and accuracy. In: International Conference on Machine Learning. pp. 7472–7482. PMLR (2019)
45. Zhang, H.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
46. Zhu, J., Yao, J., Han, B., Zhang, J., Liu, T., Niu, G., Zhou, J., Xu, J., Yang, H.: Reliable adversarial distillation with unreliable teachers. In: International Conference on Learning Representations (2021)
47. Zi, B., Zhao, S., Ma, X., Jiang, Y.G.: Revisiting adversarial robustness distillation: Robust soft labels make student better. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16443–16452 (2021)