ROBUST WEIGHT PERTURBATION FOR ADVERSARIAL TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Overfitting widely exists in adversarial robust training of deep networks. An effective and promising remedy is adversarial weight perturbation, which injects the worst-case weight perturbation during network training by maximizing the classification loss on adversarial examples. Adversarial weight perturbation helps reduce the robust generalization gap; however, it also undermines the robustness enhancement. A criterion that regulates the weight perturbation is therefore crucial for adversarial training. In this paper, we propose such a criterion, namely Loss Stationary Condition (LSC) for constrained perturbation. With LSC, we find that deep network first overfits the adversarial examples with small loss, and then gradually develops to overfit all adversarial examples in the later stage of training. Following this, we find that it is essential to conduct weight perturbation on adversarial data with small classification loss to eliminate overfitting in adversarial training. Weight perturbation on adversarial data with large classification loss is not necessary and may even lead to poor robustness. Based on these observations, we propose a robust perturbation strategy to constrain the extent of weight perturbation. The perturbation strategy prevents deep networks from overfitting while avoiding the side effect of excessive weight perturbation, significantly improving the robustness of adversarial training. Extensive experiments demonstrate the superiority of the proposed method over the state-of-the-art adversarial training methods.

1 INTRODUCTION

Although deep neural networks (DNNs) have led to impressive breakthroughs in a number of fields such as computer vision (He et al., 2016), speech recognition (Wang et al., 2017), and natural language processing (Devlin et al., 2018), they are extremely vulnerable to adversarial examples that are crafted by adding small and human-imperceptible perturbation to normal examples (Szegedy et al., 2013; Goodfellow et al., 2014).

The vulnerability of DNNs has attracted extensive attention and led to a large number of defense techniques against adversarial examples. Across existing defenses, adversarial training (AT) is one of the strongest empirical defenses. AT directly incorporates adversarial examples into the training process to solve a min-max optimization problem (Madry et al., 2017), which can obtain models with moderate adversarial robustness and has not been comprehensively attacked (Athalye et al., 2018). However, different from the standard training scenario, overfitting is a dominant phenomenon in adversarial robust training of deep networks (Rice et al., 2020). After a certain point in AT, the robust performance on test data will continue to degrade with further training. This phenomenon, termed as *robust overfitting*, breaches the common practice in deep learning that using over-parameterized networks and training for as long as possible (Neyshabur et al., 2017; Belkin et al., 2019). Such anomaly in AT causes detrimental effects on the robust generalization performance and subsequent algorithm assessment (Rice et al., 2020; Chen et al., 2020b). Relief techniques that mitigate robust overfitting have thus become crucial for stable adversarial training.

An effective and promising remedy for robust overfitting is Adversarial Weight Perturbation (AWP) (Wu et al., 2020), which forms a double-perturbation mechanism in the adversarial train-

ing framework that adversarially perturbs both inputs and weights:

$$\min_{\boldsymbol{w}} \max_{\boldsymbol{v}\in\mathcal{V}} \frac{1}{n}\sum_{i=1}^{n} \max_{||x_i'-x_i||_p\leq\epsilon} \ell(f_{\boldsymbol{w}+\boldsymbol{v}}(x_i'), y_i), \tag{1}$$

where n is the number of training examples, x_i' is the adversarial example of x_i, $f_{\boldsymbol{w}}$ is the DNN with weight $\boldsymbol{w}$, $\ell(\cdot)$ is the loss function, ϵ is the maximum perturbation constraint for inputs (*i.e.*, $||x_i' - x_i||_p \leq \epsilon$), and $\mathcal{V}$ is the feasible perturbation region for weights (*i.e.*, $\{\boldsymbol{v} \in \mathcal{V} : ||\boldsymbol{v}||_2 \leq \gamma||\boldsymbol{w}||_2\}$, where γ is the constraint on weight perturbation size). The inner maximization is to find adversarial examples x_i' within the ϵ-ball centered at normal examples x_i that maximizes the classification loss ℓ. On the other hand, the outer maximization is to find weight perturbation $\boldsymbol{v}$ that maximizes the loss ℓ on adversarial examples to flatten the weight loss landscape and reduce robust generalization gap. This is the problem of training a weight-perturbed robust classifier on adversarial examples. Therefore, how well the weight perturbation is found directly affects the performance of the outer minimization, *i.e.*, the robustness of the classifier.

Several attack methods have been used to solve the inner maximization problem in Eq.(1), such as Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014) and Projected Gradient Descent (PGD) (Madry et al., 2017). For the outer maximization problem, AWP (Wu et al., 2020) injects the worst-case weight perturbation to reduce robust generalization gap. However, the extent to which the weights should be perturbed has not been explored. Without an appropriate criterion to regulate the weight perturbation, the adversarial training procedure is difficult to unleash its full power. In this paper, we propose such a criterion, namely Loss Stationary Condition (LSC) for constrained perturbation, which sheds light on the nitty-gritty of robust overfitting in adversarial training, and this in turn motivates us to propose an improved weight perturbation strategy for better robustness. Our main contributions are follows:

- We propose a principled criterion LSC to monitor the training status of different adversarial examples during network optimization. It provides a better understanding of robust overfitting in adversarial training, and it is also a good indicator for efficient weight perturbation.
- With LSC, we find that deep network first overfits adversarial data with small classification loss and then gradually develops to overfit all adversarial data. Following this, we find that better perturbation of model weights is associated with perturbing on adversarial data with small classification loss. For adversarial data with large classification loss, weight perturbation is not necessary and can even be harmful.
- We propose a robust perturbation strategy to constrain the extent of weight perturbation. Experiments show that the robust strategy significantly improves the robustness of adversarial training.

2 Related Work

2.1 Adversarial Attacks

Given a normal example (x, y), a DNN $f_{\boldsymbol{w}}$, and maximum perturbation constraint ϵ. Let $\mathcal{X}$ denote the input feature space and $\mathcal{B}_\epsilon^p(x) = \{x' \in \mathcal{X} : ||x' - x||_p \leq \epsilon\}$ be the ℓ_p-norm ball of radius ϵ centered at x in $\mathcal{X}$. The goal of adversarial attack is to find an adversarial example $x' \in \mathcal{B}_\epsilon^p(x)$ that can fool the DNN to produce an incorrect output ($f_w(x') \neq y$). Here we selectively introduce several commonly used adversarial attack methods.

Fast Gradient Sign Method (FGSM). FGSM (Goodfellow et al., 2014) perturbs natural example x for one step with step size ϵ along the gradient direction:

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x \ell(f_{\boldsymbol{w}}(x), y)). \tag{2}$$

Projected Gradient Descent (PGD). PGD (Madry et al., 2017) is a stronger iterative variant of FGSM, which perturbs normal example x for multiple steps K with a smaller step size α:

$$x^0 \sim \mathcal{U}(\mathcal{B}_\epsilon^p(x)), \tag{3}$$

$$x^k = \Pi_{\mathcal{B}_\epsilon^p(x)}(x^{k-1} + \alpha \cdot \text{sign}(\nabla_{x^{k-1}} \ell(f_{\boldsymbol{w}}(x^{k-1}), y))), \tag{4}$$

where $\mathcal{U}$ denotes the uniform distribution, x^0 denotes the normal example disturbed by a small uniform random noise, x^k denotes the adversarial example at step k, and $\Pi_{\mathcal{B}_\epsilon^p(x)}$ denotes the projection function that projects the adversarial example back into the set $\mathcal{B}_\epsilon^p(x)$ if necessary.

AutoAttack (AA). AA (Croce & Hein, 2020b) is an ensemble of complementary attacks, which consists of three white-box attacks (APGD-CE (Croce & Hein, 2020b), APGD-DLR (Croce & Hein, 2020b), and FAB (Croce & Hein, 2020a)) and a black-box attack (Square Attack (Andriushchenko et al., 2020)). AA regards models to be robust only if the models correctly classify all types of adversarial examples, which is among the most reliable evaluation of adversarial robustness to date.

There are also other types of attacking methods, *e.g.*, the CW attack (Carlini & Wagner, 2017), deformation attack (Engstrom et al., 2017; Xiao et al., 2018; Engstrom et al., 2019), Hamming distance based attack (Shamir et al., 2019), Frank-Wolfe based attack (Chen et al., 2020a) and adaptive attack (Tramer et al., 2020).

2.2 Adversarial Defense

Since the discovery of adversarial examples, a large number of works have emerged for defending against adversarial attacks, such as input denoising (Guo et al., 2018; Liao et al., 2018; Wu et al., 2021), defensive distillation (Papernot et al., 2016; Carlini & Wagner, 2017), adversarial detection (Metzen et al., 2017; Tao et al., 2018), gradient regularization (Tramèr et al., 2018; Ross & Doshi-Velez, 2018) and adversarial training (Goodfellow et al., 2014; Madry et al., 2017). Among them, adversarial training has been demonstrated to be the most effective method (Athalye et al., 2018). Based on adversarial training, a wide range of subsequent works are then proposed to further improve the model robustness (Xie et al., 2019; Mosbach et al., 2018; Kannan et al., 2018; Zhang et al., 2019; Cai et al., 2018; Wang et al., 2019a; Zhang et al., 2020a; Dong et al., 2018; Yang et al., 2019; Wang et al., 2019b; Song et al., 2020; Carmon et al., 2019; Zhai et al., 2019; Uesato et al., 2019; Hendrycks et al., 2019; Yan et al., 2021; Du et al., 2021). Here, we introduce two currently state-of-the-art adversarial training frameworks.

TRADES. TRADES (Zhang et al., 2019) optimizes a regularized surrogate loss that is a trade-off between the natural accuracy and adversarial robustness:

$$\ell^{\mathrm{TRADES}}(\boldsymbol{w}; x, y) = \frac{1}{n}\sum_{i=1}^{n}\left\{\mathrm{CE}(f_{\boldsymbol{w}}(x_i), y_i) + \beta \cdot \max_{x' \in \mathcal{B}_\epsilon^p(x)} \mathrm{KL}(f_{\boldsymbol{w}}(x_i)||f_{\boldsymbol{w}}(x_i'))\right\}, \tag{5}$$

where CE is the cross-entropy loss that encourages the network to maximize the natural accuracy, KL is the Kullback-Leibler divergence that encourages to improve the robust accuracy, and β is the hyperparameter to control the trade-off between natural accuracy and adversarial robustness.

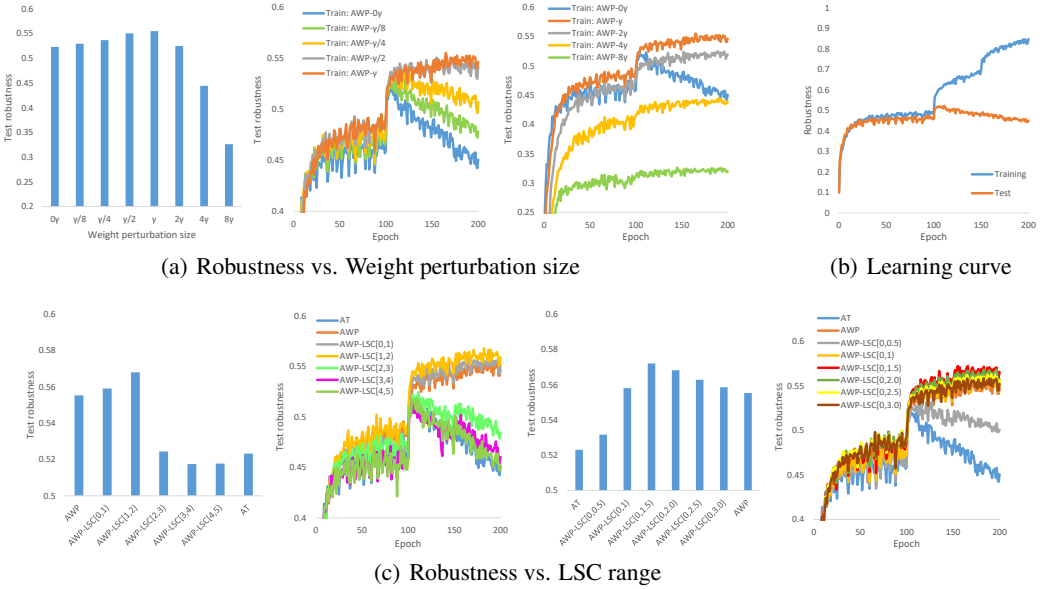
Robust Self-Training (RST). RST (Carmon et al., 2019) utilize additional 500K unlabeled data extracted from the 80 Million Tiny Images dataset (Torralba et al., 2008). RST first leverages the surrogate natural model to generate pseudo-labels for these unlabeled data, and then adversarially trains the network with both additional pseudo-labeled unlabeled data $(\tilde{x}, \tilde{y})$ and original labeled data (x, y) in a supervised setting:

$$\ell^{\mathrm{RST}}(\boldsymbol{w}; x, y, \tilde{x}, \tilde{y}) = \ell^{\mathrm{TRADES}}(\boldsymbol{w}; x, y) + \lambda \cdot \ell^{\mathrm{TRADES}}(\boldsymbol{w}; \tilde{x}, \tilde{y}), \tag{6}$$

where λ is the weight on unlabeled data.

2.3 Robust Overfitting

Nowadays, there are effective countermeasures to alleviate the overfitting in standard training. But in adversarial training, robust overfitting widely exists and those common countermeasures used in standard training help little (Rice et al., 2020). Schmidt et al. (2018) explains robust overfitting partially from the perspective of sample complexity, and is supported by empirical results in derivative works, such as adversarial training with semi-supervised learning (Carmon et al., 2019; Uesato et al., 2019; Zhai et al., 2019), robust local feature (Song et al., 2020) and data interpolation (Zhang & Xu, 2019; Lee et al., 2020; Chen et al., 2021). Separate works have also attempt to mitigate robust overfitting by the unequal treatment of data (Zhang et al., 2020b) and weight smoothing (Chen et al., 2020b). Recent study (Wu et al., 2020) reveals the connection between the flatness of weight loss



(a) Robustness vs. Weight perturbation size

(b) Learning curve

(c) Robustness vs. LSC range

Figure 1: (a): Test robustness of AWP with varying weight perturbation size; (b): The learning curve of vanilla AT; (c): Test robustness of AWP with varying LSC range.

landscape and robust generalization gap, and proposes to incorporate adversarial weight perturbation mechanism in the adversarial training framework. Despite the efficacy of adversarial weight perturbation in suppressing the robust overfitting in adversarial training, a deeper understanding of the cause of robust overfitting and a clear direction for valid weight perturbation is largely missing. The outer maximization in Eq.(1) lacks an effective criterion to regulate and constrain the extent of weight perturbation, which in turn influences the optimization of the outer minimization problem. In this paper, we propose such a criterion and provide new understanding of the robust overfitting in adversarial training. Following this, we design a robust weight perturbation strategy that significantly improves the robustness of adversarial training.

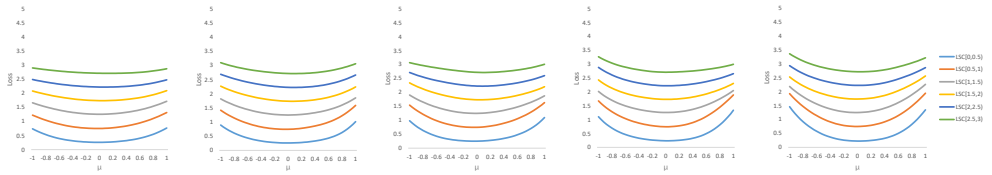
3 Loss Stationary Condition

In this section, we first empirically investigate the relationship between weight perturbation robustness and adversarial robustness, and then propose a new criterion to monitor the training status of different adversarial examples in the learning process of adversarial training, which leads to a new perspective of robust overfitting. To this end, some discussions about robust overfitting and adversarial weight perturbation are provided.

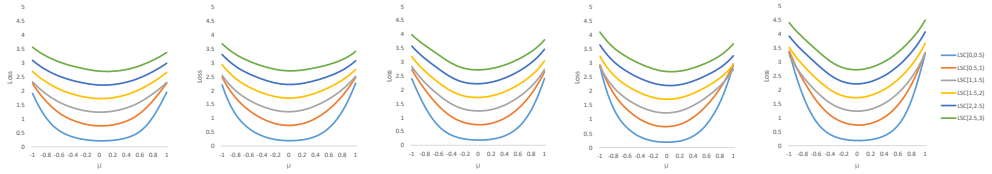
Does Weight Perturbation Robustness Lead to Better Adversarial Robustness? First, we investigate whether the robustness against weight perturbation is beneficial to the adversarial robustness. In particular, we train PreAct ResNet-18 with AWP on CIFAR-10 using varying weight perturbation size from 0γ, $\gamma/8$, $\gamma/4$, $\gamma/2$, γ, 2γ, 4γ to 8γ. In each setting, we evaluate the robustness of the model against 20-step PGD (PGD-20) attacks on CIFAR-10 test images. As shown in Figure 1(a), when varying weight perturbation size, the best adversarial robustness has a certain improvement in the early stage. When weight perturbation size is large, the best adversarial robustness begins to decrease significantly as the size of the perturbation increases. It might be explained by the fact that the network has to sacrifice adversarial robustness to allocate more capacity to defend against weight perturbation, which implies that weight perturbation robustness and adversarial robustness are not actually mutually beneficial. The performance gain of AWP is mainly due to suppressing robust overfitting.

Loss Stationary Condition. In order to further understand the robust overfitting, we propose a criterion that divides the training adversarial examples into different groups according to their classification loss:

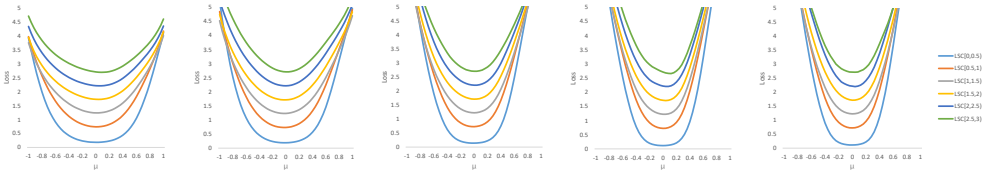
$$\mathrm{LSC}[p, q] = \{x' \in \mathcal{X} \mid p \leq \ell(f_{\boldsymbol{w}}(x'), y) \leq q\}, \tag{7}$$



(a) Epoch 20 (b) Epoch 40 (c) Epoch 60 (d) Epoch 80 (e) Epoch 100



(f) Epoch 101 (g) Epoch 104 (h) Epoch 108 (i) Epoch 112 (j) Epoch 116



(k) Epoch 120 (l) Epoch 140 (m) Epoch 160 (n) Epoch 180 (o) Epoch 200

Figure 2: The weight loss landscape of different LSC groups on different checkpoints.

where $p \leq q$. The adversarial examples in the group all satisfy their classification loss within a certain range, which is termed Loss Stationary Condition (LSC). The proposed criterion LSC allows the analysis of training status of different adversarial examples independently, and provides more insights into the robust overfitting.

LSC View of Robust Overfitting. To provide details of the robust overfitting in adversarial training, we train a PreAct ResNet-18 for 200 epochs on CIFAR-10 using PGD-10 with step size $\epsilon/4$, maximum perturbation $\epsilon = 8/255$, following the standard setting in Madry et al. (2017). The learning curve is shown in Figure 1(b). For each intermediate model, we then apply the same PGD-10 attack on CIFAR-10 training images to craft adversarial examples, and divide the crafted adversarial examples into 6 consecutive LSC groups ranging from 0.0 to 3.0. Then, we use the weight loss landscape to characterize the training status of the adversarial examples in each LSC group, which plots the classification loss change when perturbing the model weight $\boldsymbol{w}$ by a random noise $\boldsymbol{d}$ with magnitude μ:

$$g_j(\mu) = \frac{1}{n_j} \sum_{i=1}^{n_j} \ell(f_{\boldsymbol{w}+\mu\boldsymbol{d}}(x'_{ij}), y_{ij}),\ x'_{ij} \in \text{LSC}[p_j, q_j], \tag{8}$$

where j is the number of groups, n_j is the number of adversarial examples in j-th LSC group, and $\boldsymbol{d}$ is filter normalized by $\boldsymbol{d} \leftarrow \frac{\boldsymbol{d}}{||\boldsymbol{d}||}||\boldsymbol{w}||$ following Li et al. (2017). It is worth noting that weight loss landscape has been widely used to characterize the generalization gap (Neyshabur et al., 2017; Foret et al., 2020; Wu et al., 2020). Here, we use it to characterize the training status of different adversarial examples. For training adversarial examples, the higher the degree of overfitting by the model, the more sensitive its loss is to model weight perturbations, thus making the weight loss landscape sharper. Here the weight loss curve sharpness is served as a comparable measurement of overfitting strength. Besides, another key difference to previous works lies on the LSC criterion used for visualization, which provides more insights into the robust overfitting.

We show the weight loss curve of each LSC group on different checkpoints in Figure 2. In the early stage of training (between 100 and 120 epoch), it can be seen that the weight loss curve of the LSC group with small loss is obviously sharper than that of the LSC group with large loss, which indicates that the adversarial examples with small classification loss were first overfitted. As the training progresses, the weight loss curves of all LSC groups become very sharp, which shows that the network overfits all adversarial examples. These observations suggest that robust overfitting

Algorithm 1 Robust Weight Perturbation

Input: Network $f_{\boldsymbol{w}}$, training data S, mini-batch $\mathcal{B}$, batch size n, learning rate η, PGD step size α, PGD steps K_1, PGD constraint ϵ, RWP steps K_2, RWP constraint γ, minimum LSC value c_{min}.
Output: Adversarially robust model $f_{\boldsymbol{w}}$.
repeat
  Read mini-batch $\boldsymbol{x}_{\mathcal{B}}$ from training set S.
  $x'_{\mathcal{B}} \leftarrow x_{\mathcal{B}} + \delta$, where $\delta \sim \text{Uniform}(-\epsilon, \epsilon)$
  for $k = 1$ **to** K_1 **do**
    $x'_{\mathcal{B}} \leftarrow \Pi_{\epsilon}(x'_{\mathcal{B}} + \alpha \cdot \text{sign}(\nabla_{x'_{\mathcal{B}}} \ell(f_{\boldsymbol{w}}(x'_{\mathcal{B}}), y)))$
  end for
  Initialize $\boldsymbol{v} = \boldsymbol{0}$
  for $k = 1$ **to** K_2 **do**
    $V = \mathbb{1}_{\mathcal{B}}(\ell(f_{\boldsymbol{w}+\boldsymbol{v}}(x'_{\mathcal{B}}), y) \leq c_{min})$
    if $\sum V = 0$ **then**
      break
    else
      $\boldsymbol{v} \leftarrow \boldsymbol{v} + \nabla_{\boldsymbol{v}}(V \cdot \ell(f_{\boldsymbol{w}+\boldsymbol{v}}(x'_{\mathcal{B}}), y))$
      $\boldsymbol{v} \leftarrow \gamma \frac{\boldsymbol{v}}{||\boldsymbol{v}||}||\boldsymbol{w}||$
    end if
  end for
  $\boldsymbol{w} \leftarrow (\boldsymbol{w} + \boldsymbol{v}) - \eta \nabla_{\boldsymbol{w}+\boldsymbol{v}} \frac{1}{n} \sum_{i=1}^{n} \ell(f_{\boldsymbol{w}+\boldsymbol{v}}(x'^{(i)}_{\mathcal{B}}), y^{(i)}) - \boldsymbol{v}$
until training converged

exists a diffusion process: the model will first memorize some easy-to-learn adversarial examples, and then spread to the entire training dataset.

LSC view of Adversarial Weight Perturbation. To provide more insight into how AWP suppresses robust overfitting, we train PreAct ResNet-18 on CIFAR-10 by varying the LSC group that performs adversarial weight perturbation. In each setting, we evaluate the robustness of the model against PGD-20 attacks on CIFAR-10 test images. As shown in Figure 1(c), when varying the LSC range, we can observe that conducting adversarial weight perturbation on adversarial examples with small classification loss is sufficient to suppress robust overfitting. Recalling the diffusion process in robust overfitting, we can infer that to eliminate robust overfitting, it is essential to prevent the model from memorizing the easy-to-learn adversarial examples. Besides, it is observed that conducting adversarial weight perturbation on adversarial examples with large classification loss leads to worse adversarial robustness, which again verifies that the robustness against weight perturbation will not bring adversarial robustness gain, or even on the contrary, it undermines the adversarial robustness enhancement.

Do We Really Need the Worst-case Weight Perturbation? As aforementioned, the robustness against weight perturbation is detrimental to the adversarial robustness enhancement. Therefore, to purely prevent the network from memorizing the adversarial examples with small classification loss, conducting worst-case weight perturbation on these adversarial examples is not necessary, since it will also deteriorate the adversarial robustness. In the next section, we will propose a robust perturbation strategy to address this issue.

4 Robust Weight Perturbation

In this section, we introduce the proposed robust weight perturbation strategy and its algorithmic realization.

As mentioned in Section 3, conducting adversarial weight perturbation on adversarial examples with small classification loss is enough to prevent robust overfitting and leads to higher robustness. However, conducting adversarial weight perturbation on adversarial examples with large classification loss may not be helpful. Recalling the criterion LSC proposed in Section 3, we have seen that it is closely correlated with the tendency of adversarial example to be overfitted. Thus, it can be used to regulate the extent of weight perturbation at a fine-grained level. Therefore, we propose to train network with adversarial examples that are all above a minimum LSC value, so as to ensure that no

robust overfitting occurs while avoiding the side effect of excessive weight perturbation. Let c_{min} be the minimum LSC value. Instead of generating weight perturbation $\boldsymbol{v}$ via outer maximization in Eq.(1), we generate $\boldsymbol{v}$ as follows:

$$\begin{aligned} \boldsymbol{v}^{k+1} &= \boldsymbol{v}^{k} + \nabla_{\boldsymbol{v}^k} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(x_i', y_i) \ell(f_{\boldsymbol{w}+\boldsymbol{v}^k}(x_i'), y_i), \\ \text{where} \quad \mathbb{1}(x_i', y_i) &= \begin{cases} 0 & \text{if } \ell(f_{\boldsymbol{w}+\boldsymbol{v}^k}(x_i'), y_i) > c_{min} \\ 1 & \text{if } \ell(f_{\boldsymbol{w}+\boldsymbol{v}^k}(x_i'), y_i) \leq c_{min} \end{cases} \end{aligned} \tag{9}$$

The proposed Robust Weight Perturbation (RWP) algorithm is shown in Algorithm 1. We use PGD attack (Madry et al., 2017) to generate the training adversarial examples, which can be also extended to other variants such as TRADES (Zhang et al., 2019) and RST (Carmon et al., 2019). The mimimum LSC value c_{min} controls the minimum classification loss (minimum weight perturbation strength) of the adversarial examples during network training. In the early stages of training, the classification loss of adversarial example is generally larger than c_{min} corresponding to no weight perturbation process. The classification loss of adversarial examples then decreases as training progresses. At each optimization step, we monitor the classification loss of the adversarial example and conduct the weight perturbation process for adversarial examples whose classification loss is already smaller than c_{min}, enabled by an indicator control vector V. At each perturbation step, the weight perturbation $\boldsymbol{v}$ will be updated to increase the classification loss of the corresponding adversarial example. When the classification loss of training adversarial examples is all higher than c_{min} or the number of perturbation step reaches the defined value, we stop the weight perturbation process and inject the generated weight perturbation $\boldsymbol{v}$ for adversarial training.

5 Experiments

In this section, we conduct comprehensive experiments to evaluate the effectiveness of RWP including its experimental settings, robustness evaluation and ablation studies.

5.1 Experimental Setup

Baselines and Implementation Details. Our implementation is based on PyTorch and the code as well as other related resources will be released for public use and verification. We conduct extensive experiments across three benchmark datasets (CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 (Krizhevsky et al., 2009) and SVHN (Netzer et al., 2011)) and two threat models (L_∞ and L_2). We use PreAct ResNet-18 (He et al., 2016) and Wide ResNet (WRN-28-10 and WRN-34-10) (Zagoruyko & Komodakis, 2016) as the network structure following Wu et al. (2020). We compare the performance of the proposed method on a number of baseline methods: 1) standard adversarial training without weight perturbation, including vanilla AT (Madry et al., 2017), TRADES (Zhang et al., 2019) and RST (Carmon et al., 2019); 2) adversarial training with adversarial weight perturbation (AWP) (Wu et al., 2020). For training, the network is trained for 200 epochs using SGD with momentum 0.9, weight decay 5×10^{-4}, and an initial learning rate of 0.1. The learning rate is divided by 10 at the 100-th and 150-th epoch. Standard data augmentation including random crops with 4 pixels of padding and random horizontal flips are applied. For testing, model robustness is evaluated by measuring the accuracy of the model under different adversarial attacks. For hyper-parameters in RWP, we set perturbation step $K_2 = 10$ for all datasets. The minimum LSC value $c_{min} = 1.7$ for CIFAR-10, $c_{min} = 2.2$ for SVHN and $c_{min} = 4.0$ for CIFAR-100. The weight perturbation budget of $\gamma = 0.01$ for AT-RWP, $\gamma = 0.005$ for TRADES-RWP and RST-RWP following literature (Wu et al., 2020). Other hyper-parameters of the baselines are configured as per their original papers.

Adversarial Setting. The training attack is 10-step PGD attack with random start. We follow the same settings in Rice et al. (2020) : for L_∞ threat model, $\epsilon = 8/255$, step size $\alpha = 1/255$ for SVHN, and $\alpha = 2/255$ for both CIFAR10 and CIFAR100; for L_2 threat model, $\epsilon = 128/255$, step size $\alpha = 15/255$ for all datasets. The test attacks used for robustness evaluation are generated from the original test set images by attacking the defense models using different attacking methods, including: FGSM, PGD-20, PGD-100, C&W$_\infty$ (L_∞ version of C&W optimized by PGD for 100 steps) and Auto Attack (AA).

5.2 ROBUSTNESS EVALUATION

Performance Evaluations. To validate the effectiveness of the proposed RWP, we conduct performance evaluation on vanilla AT, AT-AWP and AT-RWP across different benchmark datasets and threat models using PreAct ResNet-18. We report the accuracy on the test images under PGD-20 attack. The evaluation results are summarized in Table 1. "Best" denotes the highest robustness that ever achieved at different checkpoints and "last" denotes the robustness at the last epoch checkpoint. It is observed vanilla AT suffers from severe robust overfitting (the performance gap between "best" and "last" is very large). AT-AWP and AT-RWP method narrow the performance gap significantly over the vanilla AT model due to suppression of robust overfitting. Moreover, on CIFAR-10 dataset under the L_∞ attack, vanilla AT achieves 52.79% "best" test robustness. The AT-AWP approach boosts the performance to 55.39%. The proposed approach further outperforms both methods by a large margin, improving over vanilla AT by 5.76%, and is 3.16% better than AT-AWP, achieving 58.55% accuracy under the standard 20 steps PGD attack. Similar patten has been observed on other datasets and threat model. AT-RWP consistently improves the test robustness across a wide range of datasets and threat models, demonstrating the effectiveness of the proposed approach.

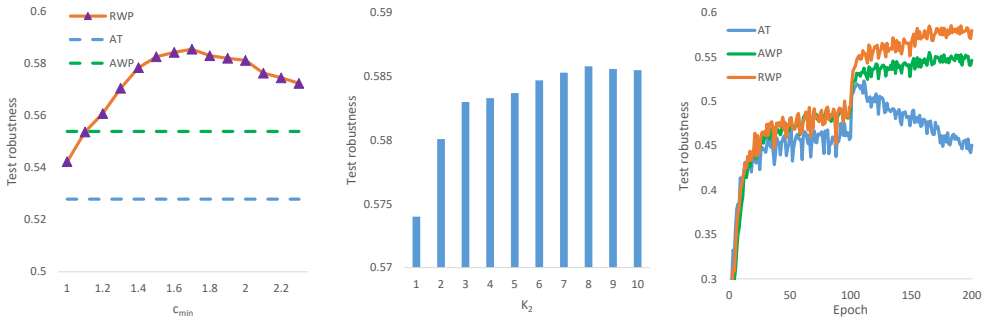
Table 1: Test robustness (%) of AT, AT-AWP and AT-RWP using PreAct ResNet-18.

Threat Model	Method	SVHN		CIFAR-10		CIFAR-100	
		Best	Last	Best	Last	Best	Last
L_∞	AT	53.36	44.49	52.79	44.44	27.22	20.82
	AT-AWP	59.12	55.87	55.39	54.73	30.71	30.28
	AT-RWP	**61.15**	**57.45**	**58.55**	**58.01**	**31.17**	**30.64**
L_2	AT	66.87	65.03	69.15	65.93	41.33	35.27
	AT-AWP	72.57	67.73	72.69	72.08	45.60	44.66
	AT-RWP	**73.35**	**69.48**	**74.47**	**73.84**	**45.71**	**45.05**

Benchmarking the state-of-the-art Robustness. To manifest the full power of our proposed perturbation strategy and also benchmark the state-of-the-art robustness on CIFAR-10 under L_∞ threat model, we conduct experiments on the large capacity network with different baseline methods. We train Wide ResNet-34-10 for AT and TRADES, and Wide ResNet-28-10 for RST following their original papers. We evaluate the adversarial robustness of trained model with various test attack and report the "best" test robustness, with the results shown in Table 2. "Natural" denotes the accuracy on natural test data. First, it is observed that the natural accuracy of RWP model consistently outperforms AWP by a large margin. It is due to the benefits that our RWP avoids the excessive weight perturbation. Moreover, RWP achieves the best adversarial robustness against all types of attack across a wide range of baseline methods, which verifies that RWP is effective in general and improves adversarial robustness reliably rather than improper tuning of hyper-parameters of attacks, gradient obfuscation or masking.

Table 2: Test robustness (%) on CIFAR-10 using Wide ResNet under L_∞ threat model.

Defense	Natural	FGSM	PGD-20	PGD-100	C&W$_\infty$	AA
AT	86.07	61.76	56.10	55.79	54.19	52.60
AT-AWP	85.57	62.90	58.14	57.94	55.96	54.04
AT-RWP	**86.86**	**66.22**	**62.87**	**62.87**	**56.62**	**54.61**
TRADES	84.65	61.32	56.33	56.07	54.20	53.08
TRADES-AWP	85.36	63.49	59.27	59.12	57.07	56.17
TRADES-RWP	**86.14**	**64.70**	**60.45**	**60.30**	**58.07**	**57.20**
RST	**89.69**	69.60	62.60	62.22	60.47	59.53
RST-AWP	88.25	67.94	63.73	63.58	61.62	60.05
RST-RWP	88.87	**69.71**	**64.11**	**63.92**	**62.03**	**60.36**



(a) Importance of minimum LSC value c_{min} (b) Impacts of step number K_2 (c) Effect of RWP on adversarial robustness and robust overfitting

Figure 3: The ablation study experiments on CIFAR-10.

5.3 Ablation Studies

In this part, we investigate the impacts of algorithmic components using AT-RWP on PreAct ResNet-18 under L_∞ threat model with $\epsilon = 8/255$ and $\alpha = 2/255$ following the same setting in section 5.1. The training/test attacks are PGD-10/PGD-20 respectively.

The Importance of Minimum LSC Value. We empirically verify the effectiveness of minimum LSC value c_{min}, by comparing the performance of models trained using different weight perturbation schemes: 1) AT: standard adversarial training without weight perturbation (equivalent to $c_{min} = 0$); 2) AWP: weight perturbation generated via outer maximization in Eq.(1) (equivalent to $c_{min} = \infty$); 3) RWP: weight perturbation generated using the proposed robust strategy with different c_{min} values. All other hyper-parameters are kept exactly the same other than the perturbation scheme used. The results are summarized in Table 3(a). It is observed that the test robustness of RWP model first increases and then decreases as the minimum LSC value increases, and the best test robustness is obtained at $c_{min} = 1.7$. It is evident that RWP with a wide range of c_{min} outperforms both AT and AWP model, demonstrating its effectiveness. Furthermore, as it is the major component that is different from the AWP pipeline, this result suggests that LSC criterion constraints is the main contributor to the improved adversarial robustness.

The Impact of Step Number. We further investigate the effect of step number K_2, by comparing the performances of model trained using different perturbation steps. The step number K_2 for RWP varies from 1 to 10. The results are shown in Figure 3(b). As expected, when K_2 is small, increasing K_2 leads higher test robustness. When K_2 increases from 7 to 10, the performance is flat, which suggests that the generating weight perturbation is sufficient to comprehensively avoid robust overfitting. Note that extra iterations will not bring computational overhead when classification loss of adversarial examples in the batch exceeds minimum LSC value c_{min}, as shown in Algorithm 1. Therefore, we uniformly use $K_2 = 10$ in our implementation.

Effect on Adversarial Robustness and Robust Overfitting. We then visualize the learning curve of AT, AWP and RWP in Figure 3(c). We observe that the test robustness of RWP model continues to increase as the training progresses. In addition, RWP outperforms AWP with a clear margin in the later stage of training. Such observations exactly reflect the nature of our approach which aims to prevent robust overfitting as well as enhance adversarial robustness.

6 Conclusion

In this paper, we proposed a criterion, Loss Stationary Condition (LSC) for constrained perturbation, to monitor the training status of different adversarial examples during network optimization. The proposed criterion provides a new understanding of robust overfitting in adversarial training. Based on LSC, we found that elimination of robust overfitting and higher robustness of adversarial training can be achieved by weight perturbation on adversarial examples with small classification loss, rather than adversarial examples with large classification loss. Following this, we proposed a Robust Weight Perturbation (RWP) strategy to monitor and regulate the extent of weight perturbation. Comprehensive experiments show that RWP is generic and can improve the state-of-the-art adversarial robustness across different adversarial training approaches, network architectures, threat models and benchmark datasets.

REPRODUCIBILITY STATEMENT

For sake of reproducibility of our algorithm, we make the following efforts: (**i**) In Section 5.1, we clearly state the implementation details, including benchmark datasets, network structure, baselines, training and test parameter setting as well as training and test attack setting. (**ii**) In Section 5.3, we evaluate the sensitivity of the algorithm to hyperparameters and show the detailed hyperparameter tuning process. (**iii**) At last, we open-source the source code of RWP algorithm, available at supplementary material.

REFERENCES

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pp. 484–501. Springer, 2020.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Qi-Zhi Cai, Min Du, Chang Liu, and Dawn Song. Curriculum adversarial training. *arXiv preprint arXiv:1805.04807*, 2018.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C Duchi. Unlabeled data improves adversarial robustness. *arXiv preprint arXiv:1905.13736*, 2019.

Chen Chen, Jingfeng Zhang, Xilie Xu, Tianlei Hu, Gang Niu, Gang Chen, and Masashi Sugiyama. Guided interpolation for adversarial training. *arXiv preprint arXiv:2102.07327*, 2021.

Jinghui Chen, Dongruo Zhou, Jinfeng Yi, and Quanquan Gu. A frank-wolfe framework for efficient and effective adversarial attacks. In *AAAI*, pp. 3486–3494, 2020a.

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2020b.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pp. 2196–2205. PMLR, 2020a.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020b.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9185–9193, 2018.

Xuefeng Du, Jingfeng Zhang, Bo Han, Tongliang Liu, Yu Rong, Gang Niu, Junzhou Huang, and Masashi Sugiyama. Learning diverse-structured networks for adversarial robustness. *arXiv preprint arXiv:2102.01886*, 2021.

Logan Engstrom, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling cnns with simple transformations. *arXiv preprint arXiv:1712.02779*, 1(2):3, 2017.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pp. 1802–1811. PMLR, 2019.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

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

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pp. 2712–2721. PMLR, 2019.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

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

Saehyung Lee, Hyungyu Lee, and Sungroh Yoon. Adversarial vertex mixup: Toward better adversarially robust generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 272–281, 2020.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *arXiv preprint arXiv:1712.09913*, 2017.

Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1787, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. In *International Conference on Learning Representations*, 2017.

Marius Mosbach, Maksym Andriushchenko, Thomas Trost, Matthias Hein, and Dietrich Klakow. Logit pairing methods can fool gradient-based attacks. *arXiv preprint arXiv:1810.12042*, 2018.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597. IEEE, 2016.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pp. 8093–8104. PMLR, 2020.

Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *Advances in Neural Information Processing Systems*, 31:5014–5026, 2018.

Adi Shamir, Itay Safran, Eyal Ronen, and Orr Dunkelman. A simple explanation for the existence of adversarial examples with small hamming distance. *arXiv preprint arXiv:1901.10861*, 2019.

Chubiao Song, Kun He, Jiadong Lin, Liwei Wang, and John E Hopcroft. Robust local features for improving the generalization of adversarial training. In *International Conference on Learning Representations*, 2020.

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

Guanhong Tao, Shiqing Ma, Yingqi Liu, and Xiangyu Zhang. Attacks meet interpretability: Attribute-steered detection of adversarial samples. In *Advances in Neural Information Processing Systems*, pp. 7717–7728, 2018.

Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):1958–1970, 2008.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

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

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

Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, volume 1, pp. 2, 2019a.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019b.

Boxi Wu, Heng Pan, Li Shen, Jindong Gu, Shuai Zhao, Zhifeng Li, Deng Cai, Xiaofei He, and Wei Liu. Attacking adversarial attacks as a defense. *arXiv preprint arXiv:2106.04938*, 2021.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *arXiv preprint arXiv:2004.05884*, 2020.

Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *International Conference on Learning Representations*, 2018.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 501–509, 2019.

Hanshu Yan, Jingfeng Zhang, Gang Niu, Jiashi Feng, Vincent YF Tan, and Masashi Sugiyama. Cifs: Improving adversarial robustness of cnns via channel-wise importance-based feature selection. *arXiv preprint arXiv:2102.05311*, 2021.

Yuzhe Yang, Guo Zhang, Dina Katabi, and Zhi Xu. Me-net: Towards effective adversarial robustness with matrix estimation. In *International Conference on Machine Learning*, pp. 7025–7034. PMLR, 2019.

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

Runtian Zhai, Tianle Cai, Di He, Chen Dan, Kun He, John Hopcroft, and Liwei Wang. Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*, 2019.

Haichao Zhang and Wei Xu. Adversarial interpolation training: A simple approach for improving model robustness. 2019.

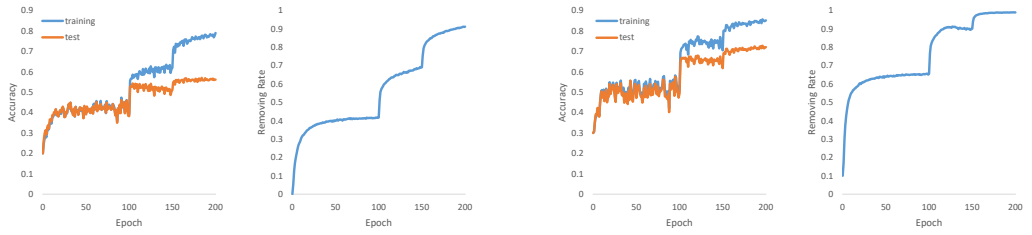
Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pp. 7472–7482. PMLR, 2019.

Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International Conference on Machine Learning*, pp. 11278–11287. PMLR, 2020a.

Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. *arXiv preprint arXiv:2010.01736*, 2020b.

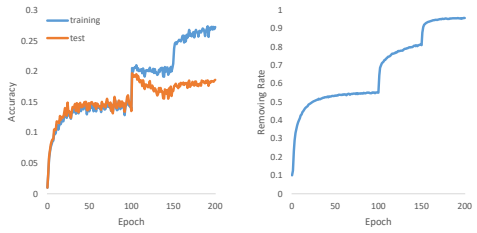
A Appendix

In this part, we verify the generalities of diffusion process in robust overfitting (the model will first memorize some easy-to-learn adversarial examples, and then spread to the entire training dataset) across different threat models, datasets and network architectures. Specifically, we remove the training examples whose loss value is lower than the LSC value during adversarial training. The learning curve and the rate of removal are shown in Figure 4. We can observe that if these easy-to-learn adversarial examples are not included in the training data, robust overfitting will not occur during adversarial training, which verified our conclusion.

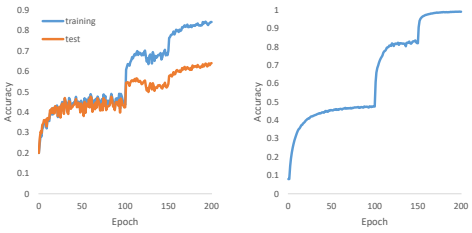


(a) Cifar10 - L_∞ - PreAct ResNet18

(b) Cifar10 - L_2 - PreAct ResNet18



(c) Cifar100 - L_∞ - PreAct ResNet18



(d) Cifar100 - L_∞ - Wide ResNet

Figure 4: The learning curve and removing rate of adversarial training under (a) Cifar10 - L_∞ - PreAct ResNet18; (b) Cifar10 - L_2 - PreAct ResNet18; (c) Cifar100 - L_∞ - PreAct ResNet18; (d) Cifar100 - L_∞ - Wide ResNet.