Understanding Robust Overfitting of Adversarial Training and Beyond

Chaojian Yu [1 †] Bo Han [2] Li Shen [3] Jun Yu [4] Chen Gong [5] Mingming Gong [6] Tongliang Liu [1]

Abstract

Robust overfitting widely exists in adversarial training of deep networks. The exact underlying reasons for this are still not completely understood. Here, we explore the causes of robust overfitting by comparing the data distribution of *non-overfit* (weak adversary) and *overfitted* (strong adversary) adversarial training, and observe that the distribution of the adversarial data generated by weak adversary mainly contain small-loss data. However, the adversarial data generated by strong adversary is more diversely distributed on the large-loss data and the small-loss data. Given these observations, we further designed data ablation adversarial training and identify that some small-loss data which are not worthy of the adversary strength cause robust overfitting in the strong adversary mode. To relieve this issue, we propose *minimum loss constrained adversarial training* (MLCAT): in a minibatch, we learn large-loss data as usual, and adopt additional measures to increase the loss of the small-loss data. Technically, MLCAT hinders data fitting when they become easy to learn to prevent robust overfitting; philosophically, MLCAT reflects the spirit of turning waste into treasure and making the best use of each adversarial data; algorithmically, we designed two realizations of MLCAT, and extensive experiments demonstrate that MLCAT can eliminate robust overfitting and further boost adversarial robustness.

1. Introduction

Adversarial examples easily mislead deep neural networks (DNNs) to produce incorrect outputs, which raises security concerns in various real-world applications since adversarial noise is usually small and human-imperceptible (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. Across existing defenses, adversarial training (AT) (Goodfellow et al., 2014; Madry et al., 2017) is one of the strongest empirical defenses (Athalye et al., 2018). AT incorporates adversarial examples into the training process and can be viewed as solving a min-max optimization problem (Madry et al., 2017).

Unfortunately, robust overfitting seems inevitable in adversarial training of deep networks: after a certain point in AT, i.e., shortly after the first learning rate decay, the robust performance on test data will continue to degrade with further training (Rice et al., 2020). More pessimistically, conventional remedies for overfitting in deep learning, including explicit regularizations, data augmentation, etc., cannot gain improvements upon early stopping (Rice et al., 2020). Nevertheless, early stopping might not be our desideratum since the expectation of double descent phenomena occurred in adversarial training (Nakkiran et al., 2019).

Robust overfitting widely exists in adversarial training of deep networks, and the exact underlying reasons for this are still not completely understood. To explore the causes of robust overfitting, we compare the data distribution of *non-overfit* adversarial training (weak adversary) and *overfitted* adversarial training (strong adversary). We observed that the training data of *non-overfit* adversarial training mainly contains small-loss data; while the distribution of training data of *overfitted* adversarial training is more divergent, usually mixed with a considerable proportion of small-loss data and large-loss data. Given these observations, we conduct a range of data ablation adversarial training experiments. By removing training data from various loss ranges, we study the impact of small-loss data and large-loss data on robust overfitting. Our results show that robust overfitting is actually caused by some small-loss data in *overfitted* adversarial training. Adversarial data that are not worthy of the adversary strength make adversarial training worse, which might be explained by the fact the network becomes more

[†] This work is done during an internship at JD Explore Academy.

[1]TML Lab, Sydney AI Centre, The University of Sydney, Sydney, Australia [2]Department of Computer Science, Hong Kong Baptist University, Hong Kong, China [3]JD Explore Academy, Beijing, China [4]Department of Automation, University of Science and Technology of China, Hefei, China [5]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China [6]School of Mathematics and Statistics, The University of Melbourne, Melbourne, Australia. Correspondence to: Tongliang Liu <tongliang.liu@sydney.edu.au>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

robust as the adversarial training progresses, making some generated adversarial data relatively less aggressive, and when their loss drops to a certain level, these adversarial data eventually lead to robust overfitting.

To relieve this issue of robust overfitting due to the small-loss data, we propose *minimum loss constrained adversarial training* (MLCAT). Specifically, MLCAT works in each mini-batch: it learns large-loss data as usual, and if there are any small-loss data, it implements additional measures on these data to increase their loss. It is a general adversarial training prototype, where small-loss data and large-loss data can be separated by a threshold. The implementation of additional measures to increase data loss can be versatile. For instance, we designed two representative methods for the realizations of MLCAT: *loss scaling* and *weight perturbation*. They adopt different strategies to improve data loss, e.g., manipulating the learning rate or model parameters, respectively. Extensive experiments show that they not only eliminate robust overfitting, but also further boost adversarial robustness.

MLCAT can be justified as follows. It is of vital importance to distinguish small-loss data and large-loss data in the context of adversarial training. The inner maximization in min-max optimization is to generate worst-case adversarial example that maximizes the classification loss (Madry et al., 2017; Wang et al., 2019a). However, the adversarial data generated by the adversary are not always qualified since the model become more robust during training. In strong adversary mode, there are some data, even if they are attacked by adversary, they are still easy to be fitted to the network, which not only fails to enhance the adversarial robustness, but also leads to robust overfitting. Therefore, if the network can be trained on the data with a minimum loss constrained, then robust overfitting may not occur. Technically, MLCAT is a specially designed adversarial training prototype to hinder data fitting when they become easy to learn, which provides a novel viewpoint on the adversary in adversarial training.

Furthermore, it is known that the sample complexity of robust generalization in adversarial training is significantly larger than that of standard generalization in natural training (Schmidt et al., 2018). A substantially larger dataset is required to achieve the double descent phenomena in AT (Carmon et al., 2019; Uesato et al., 2019; Zhai et al., 2019). However, the cost of collecting and training additional data should not be neglected. Unlike them, we delve into the causes of robust overfitting and aim to eliminate robust overfitting without exploiting additional training data. Nevertheless, simply removing small-loss data which causes robust overfitting might not be a good choice, due to the benefit of sample size (Schmidt et al., 2018). Therefore, we adopt additional measures to increase their loss to take full advantage of each adversarial data. In this sense, philosophically, MLCAT reflects the spirit of turning waste into treasure and making the best use of each adversarial data.

2. Related Work

This section briefly reviews relevant adversarial learning methods from two perspectives: adversarial training and robust overfitting.

2.1. Adversarial Training

Adversarial training (AT) has been demonstrated to be the most effective method for defending against adversarial attacks (Athalye et al., 2018; Croce et al., 2022). Let $\mathcal{X}$ and $\mathcal{Y}$ be the input and output domains. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ be the training dataset with n samples, and $\mathcal{B}_\epsilon^p(x_i) = \{x_i' \in \mathcal{X} : ||x_i' - x_i||_p \leq \epsilon\}$ be their adversarial regions, where ϵ is the maximum perturbation constraint. In AT, the training data are all sampled from adversarial regions. Madry et al. (2017) formulated AT as a min-max optimization problem:

$$\min_w \sum_i \max_{x_i' \in \mathcal{B}_\epsilon^p(x_i)} \ell(f_w(x_i'), y_i), \tag{1}$$

where f_w is the DNN classifier with weight w, and $\ell(\cdot)$ is the loss function. The inner maximization pass is to find adversarial example x_i' that maximizes the loss. The outer minimization pass is to optimize network parameters w that minimize the loss on adversarial examples. The commonly used technique to solve the inner maximization problem is Projected Gradient Descent (PGD) (Madry et al., 2017), which perturbs normal example x_i for multiple step K with step size α:

$$x_i^k = \Pi_{\mathcal{B}_\epsilon^p(x_i)}(x_i^{k-1} + \alpha \cdot \text{sign}(\nabla_{x_i^{k-1}} \ell(f_w(x_i^{k-1}), y_i))), \tag{2}$$

where x_i^k denotes the adversarial example at step k, and $\Pi_{\mathcal{B}_\epsilon^p(x_i)}$ is the projection operator.

Another typical AT variant is TRADES (Zhang et al., 2019), which optimizes a regularized surrogate loss that is a tradeoff between the natural accuracy and adversarial robustness:

$$\begin{aligned} \min_w \sum_i \{&\text{CE}(f_w(x_i), y_i) \\ &+ \beta \cdot \max_{x_i' \in \mathcal{B}_\epsilon^p(x_i)} \text{KL}(f_w(x_i) || f_w(x_i'))\}, \end{aligned} \tag{3}$$

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 tradeoff between natural accuracy and adversarial robustness.

Based on AT, subsequent works have been developed to further improve its performance, such as adversarial regularization (Kannan et al., 2018; Wang et al., 2019b), curriculum learning (Cai et al., 2018; Wang et al., 2019a; Zhang et al., 2020a), input denoising (Wu et al., 2021), modeling adversarial noise (Zhou et al., 2021; Xia et al., 2019; 2020), feature alignments (Yan et al., 2021; Bai et al., 2021) and AT tricks (Pang et al., 2020; Gowal et al., 2020).

2.2. Robust Overfitting

Robust overfitting widely exists in AT, but the cause of robust overfitting remains unclear (Rice et al., 2020). Schmidt et al. (2018) theoretically proposes that a substantially large dataset is required to achieve robust generalization, which is supported by empirical results in derivative works, such as AT 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). Different from these works, we eliminate robust overfitting without exploiting more training data. Separate works have also attempt to mitigate robust overfitting by sample reweighting (Wang et al., 2019b; Zhang et al., 2020b), weight perturbation (Wu et al., 2020; Yu et al., 2022) and weight smoothing (Chen et al., 2020). Although robust overfitting has been widely investigated, there still lacks an explanation of why it occurs. Dong et al. (2022) explores the memorization effect in AT and thinks large-loss data induces robust overfitting. In contrast, we delve into the causes of robust overfitting by investigating the data distribution of AT and identify that some small-loss data caused robust overfitting in the strong adversary mode. Then, the proposed MLCAT prototype explicitly adjusts the training data based on their loss to avoid robust overfitting. Finally, experiments on two specific realizations of MLCAT demonstrate the effectiveness on eliminating robust overfitting and robustness improvement.

3. Understanding Robust Overfitting in AT

In this section, we first study the robust overfitting by comparing the data distribution of non-overfit (weak adversary) and overfitted (strong adversary) adversarial training (Section 3.1). Based on our observations, we further propose data ablation adversarial training to identify the specific causes of robust overfitting (Section 3.2). Finally, we develop a novel adversarial training prototype, MLCAT, to eliminate robust overfitting (Section 3.3).

3.1. Non-overfit AT vs. Overfitted AT: Data Distribution Perspective

We compare the data distribution of non-overfit (weak adversary) and overfitted (strong adversary) adversarial training. Specifically, we change the strength of the adversary by adjusting the maximum perturbation size ϵ. We train PreAct ResNet-18 on CIFAR10 under L_∞ threat model using various ϵ from 0, 1, 2, 4, 6, 8 to 10. In each setting, we evaluate the accuracy of trained model on CIFAR10 test data which are attacked with the same ϵ. The test robustness of adversarial training with different adversary is shown in Figure 1(a). We can observe that there is no robust overfitting in the case of weak adversary (ϵ is small). However, in the case of strong adversary (ϵ is large), robust overfitting is a dominant phenomenon. For each case, we then visualize the distribution of training data in different loss ranges, which are shown in Figure 1(b) and Figure 1(c).

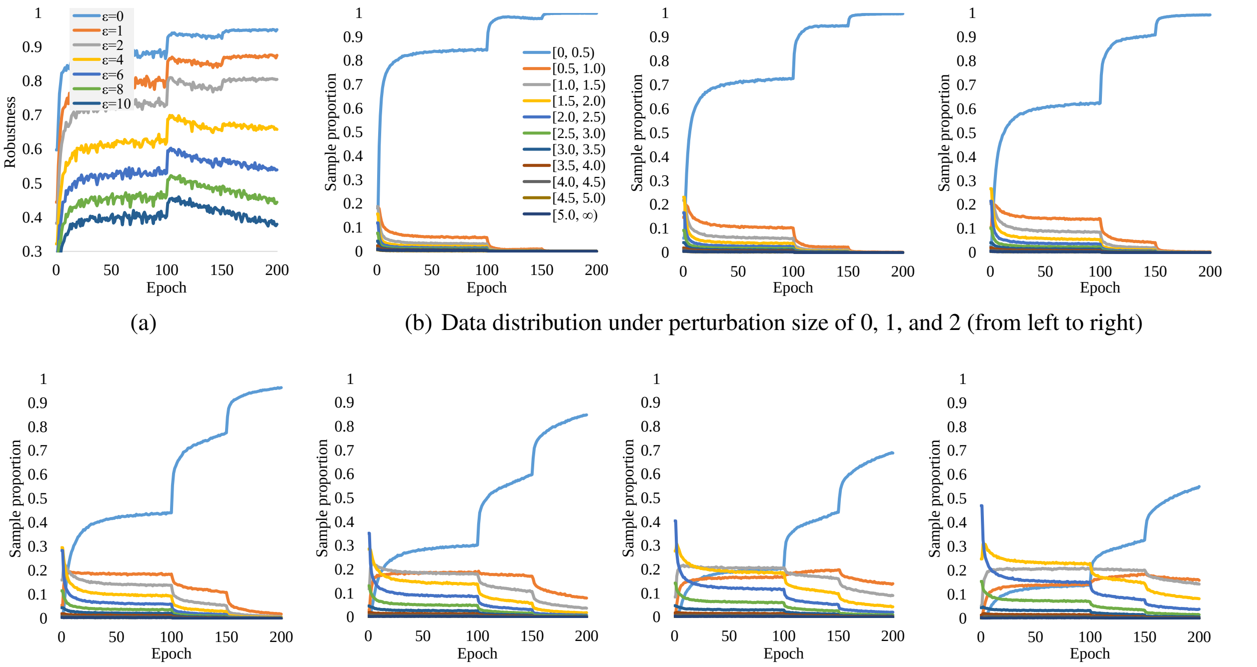
From Figure 1(b) and Figure 1(c), we can observe that the data distribution of overfitted adversarial training is obviously mismatched with that of the non-overfit adversarial training: the training data of non-overfit adversarial training mainly contains small-loss data. In contrast, the data distribution of overfitted adversarial training is more divergent, usually containing a considerable proportion of small-loss data and large-loss data. Given this observation, we wonder (Q1): *if we suppress the large-loss data in overfitted adversarial training to align the data distribution of non-overfit adversarial training, will it eliminate robust overfitting?*

On the other hand, robust overfitting occurs when the adversary becomes stronger. For strong adversary, the large-loss data is expected. In other words, for adversarial training with strong adversary, these large-loss data are good data, while these small-loss data may be bad data. From this perspective, we may ask another question (Q2): *if we suppress the small-loss data in overfitted adversarial training that does not match the strength of adversary, will it eliminate robust overfitting?*

It is worth noting that the robust overfitting behaviors and their data distribution can commonly be observed across a variety of datasets, model architectures, and threat models (shown in Appendix A), indicating that it is a general phenomenon in adversarial training. Considering the two questions above, we conduct further analysis in the next subsection to identify the specific causes of robust overfitting.

3.2. Causes of Robust Overfitting

To answer the two questions in Section 3.1, we conduct adversarial training with fixed perturbation size ($\epsilon = 8$) in a data ablation manner. Specifically, we train PreAct ResNet-18 on CIFAR10 under L_∞ threat model by removing training data from various loss ranges. To answer Q1, we remove the large-loss data within the specified loss range before robust overfitting occurs (for example, at 100th epoch), which is for the stability of optimization. To answer Q2, we remove small-loss data within the specified loss range from the beginning of training. As shown in Figure 2(a) and Figure 2(b), we can observe that adversarial training without large-loss data still has a significant robust overfitting phenomenon, which indicates the strategy of aligning with the data distribution of non-overfit adversarial training is invalid. In



(a)

(b) Data distribution under perturbation size of 0, 1, and 2 (from left to right)

(c) Data distribution under perturbation size of 4, 6, 8, and 10 (from left to right)

Figure 1. (a): The test robustness of adversarial training with various perturbation size ϵ; (b) and (c): The distribution of training data in different loss ranges under various perturbation size ϵ.

contrast, adversarial training without small-loss data can eliminate robust overfitting, which indicates that adversarial data which is not match the strength of adversary make adversarial training worse. Notice that similar experimental results can be obtained across a variety of datasets, model architectures, and threat models (shown in Appendix B). These empirical results clearly verified that the small-loss data causes robust overfitting in strong adversary mode.

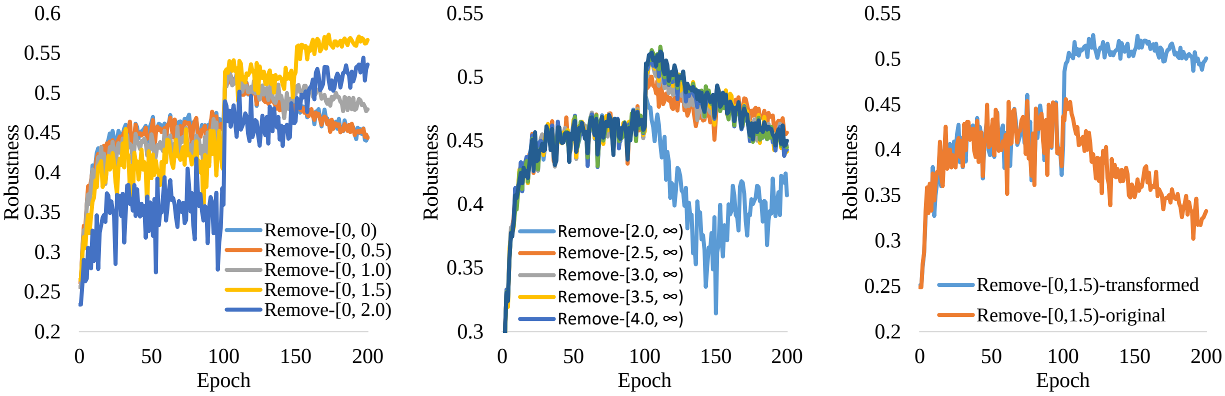
It is worth noting that the small-loss data contains two sources. As shown in Figure 1(b) and Figure 1(c), one is the original data before the learning rate decay (before 100th epoch), and the other is transformed from other loss ranges (after 100th epoch). We further investigate the respective effects on robust overfitting of the two parts of small-loss data, as shown in Figure 2(c). It is observed that robust overfitting is mainly due to these transformed small-loss data, which might be explained by the fact that as adversarial training progresses, the network becomes more robust, so some generated adversarial data are relatively less aggressive, and when they weaken to a certain extent, these adversarial data eventually lead to robust overfitting. In this paper, we take measures to eliminate robust overfitting based on general data loss. Thus we make no distinction between them and refer to them together as small-loss data. Given existing results, it does not seem to be a wise choice to remove the small-loss data directly in order to eliminate robust overfitting, because this will reduce the training sample size. In the next subsection, we introduce a novel adversarial training prototype to address this issue.

3.3. A Prototype of MLCAT

As mentioned in Section 3.2, the small-loss data causes robust overfitting in adversarial training with strong adversary. To relieve this issue, we propose to train network on adversarial data under a minimum loss constraint, dubbed as *minimum loss constrained adversarial training* (MLCAT). We adopt additional measures to increase the loss of small-loss data, so as to ensure that there is neither robust overfitting nor sample size decline.

For now, let us omit the technical details, and assume that we have a base AT method that is implemented as an algorithm with a *inner maximization pass* and a *outer minimization pass*. The inner maximization pass generates adversarial data that maximizes the loss, and then the outer minimization pass returns the gradient by backward propagating the average loss. With this algorithmic abstraction, we can present MLCAT at a high level.

The MLCAT prototype is given in Algorithm 1. Since it is only a prototype, it serves as a versatile approach where the loss adjustment strategy $\mathcal{S}$ and minimum loss condition ℓ_{min} can be flexibly implemented depend on the base AT



(a) (b) (c)

Figure 2. The test robustness of adversarial training (a) without small-loss data in various loss ranges; (b) without large-loss data in various loss ranges, and (c) without original small-loss data and transformed small-loss data.

algorithm $\mathcal{A}$. Algorithm 1 runs as follows. Given the mini-batch $\mathcal{B}$, the inner maximization pass of $\mathcal{A}$ is called in Line 2. Then the loss values are manipulated in Line 3-13 where they are reduced to a scalar for backpropagation. Before the for loop, the loss accumulator $\ell_{\mathcal{B}'}$ is initialized in Line 4. Subsequently, in Line 7, ℓ_i is added to $\ell_{\mathcal{B}'}$ if x_i is regarded as large-loss data, which will result in normal gradient descent by $\mathfrak{O}$ in Line 15; in Line 9-10, $\ell_i^{\mathcal{S}}$ is adjusted by strategy $\mathcal{S}$ and then added to $\ell_{\mathcal{B}'}$ if x_i is regarded as small-loss data, which will lead to adjusted gradient descent by $\mathfrak{O}$ in Line 15. After the for loop, the accumulated loss $\ell_{\mathcal{B}'}$ is divide by the batch size m in Line 13 to obtain the average loss. Finally, the outer minimization pass of $\mathcal{A}$ is called in Line 14, and the optimizer $\mathfrak{O}$ updates the model f_w in Line 15.

Last but not least, notice that MLCAT is extremely general. MLCAT becomes standard AT if $\mathcal{S}$ is identical mapping; it becomes data ablation adversarial training if $\mathcal{S}$ always returns 0. In AT, the sample size plays an important role in robust generalization (Schmidt et al., 2018). Hence, we use $\mathcal{S}$ to convert the small-loss data into the large-loss data. This converting behavior makes the network learn all the training data without robust overfitting, which is the key of MLCAT and what we meant by turning waste into treasure in the *abstract*. Moreover, the hyperparameter ℓ_{min} controls the range of small-loss data: when $\ell_{min} \leq 0$, it becomes standard adversarial training again; when $\ell_{min} \to \infty$, all data will be adjusted by strategy $\mathcal{S}$. Nevertheless, forcibly adjusting the loss of all data will inevitably have negative impacts on network optimization. In MLCAT, ℓ_{min} is closely related to the perturbation size ϵ and specific tasks. For example, when ϵ is small, the degradation of adversarial attack is very limited, thus ℓ_{min} may not needed since robust overfitting does not occur. Consequently, ℓ_{min} should be carefully tuned in practice in consideration of the elimination of robust overfitting as well as robustness improvement.

Algorithm 1 MLCAT-prototype (in a mini-batch).

Require: base adversarial training algorithm $\mathcal{A}$, optimizer $\mathfrak{O}$, network f_w, training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, mini-batch $\mathcal{B}$, batch size m, minimum loss conditions ℓ_{min} for $\mathcal{A}$, loss adjustment strategy $\mathcal{S}$

```
 1: Sample a mini-batch $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^m$ from $\mathcal{D}$
 2: $\mathcal{B}' = \mathcal{A}$.inner_maximization($f_w, \mathcal{B}$)
 3: $\{\ell_i\}_{i=1}^m \leftarrow \ell(f_w, \mathcal{B}')$
 4: $\ell_{\mathcal{B}'} \leftarrow 0$                # initialize loss accumulator
 5: for $i = 1, ..., m$ do
 6:     if $\ell_i \geq \ell_{min}$ then
 7:         $\ell_{\mathcal{B}'} = \ell_{\mathcal{B}'} + \ell_i$
 8:     else
 9:         $\ell_i^{\mathcal{S}} \leftarrow \mathcal{S}(f_w, x_i', \ell_{min})$          # adjust loss
10:         $\ell_{\mathcal{B}'} = \ell_{\mathcal{B}'} + \ell_i^{\mathcal{S}}$      # accumulate adjusted loss
11:     end if
12: end for
13: $\ell_{\mathcal{B}'} \leftarrow \ell_{\mathcal{B}'}/m$          # average accumulated loss
14: $\nabla_w \leftarrow \mathcal{A}$.outer_minimization($f_w, \ell_{\mathcal{B}'}$)
15: $\mathfrak{O}$.step($\nabla_w$)
```

4. Two Realizations of MLCAT

In this section, we illustrate what $\mathcal{S}$ can be used in MLCAT. We realize MLCAT through *loss scaling* that belongs to the loss-correction approach and *weight perturbation* that belongs to the parameter-correction approach. Loss scaling and weight perturbation are two representative and more importantly orthogonal methods, which spotlights the great versatility of MLCAT.

MLCAT through loss scaling. The loss-correction approach creates a corrected loss from original loss ℓ_i and then trains the model f_w based on the corrected loss. Misclassification-aware (Wang et al., 2019b) is the most primitive method in this direction. It introduces a regularizer to enhance the effect of misclassified examples on the final robustness of adversarial training.

In order to increase the loss of small-loss data. We adopt a straightforward scaling technique to correct the loss of small-loss data. Since we aim to eliminate robust overfitting, we leverage the minimum loss conditions ℓ_{min} as a constraint and scale the loss ℓ_i as follows:

$$\ell_i^S = \frac{\ell_{min}}{\ell_i} \cdot \ell_i = \ell_{min}, \tag{4}$$

where $\frac{\ell_{min}}{\ell_i}$ is the scaling coefficient, which is always greater than 1. The smaller the original loss ℓ_i, the larger the scaling coefficient, and vice versa. Previous works (Zhang et al., 2020b; Hitaj et al., 2021) show that scaling loss makes the trained model more sensitive to the logit-scaling attack. It does not matter since our aim is to verify the cause of robust overfitting, and the realization of MLCAT based on loss scaling is just designed for this purpose. Note that there is no difference between scaling losses and scaling gradients, since scaling coefficient has the same effects in increasing losses and increasing the learning rate inside $\mathfrak{D}$. Therefore, loss scaling can be regarded as learning small-loss data with a larger learning rate to effectively prevent the network from fitting these data. We refer the implementation method based on loss scaling as MLCAT$_{\text{LS}}$.

MLCAT through weight perturbation. On the other hand, the parameter-correction approach generates perturbation to the model weights, and trains the network on the perturbative parameters. AWP (Wu et al., 2020) is the most primitive method in this direction. It develops a double-perturbation mechanism that adversarially perturbs both inputs and weights to reduce the robust generalization gap:

$$\min_w \max_{v \in V} \sum_i \max_{x_i' \in \mathcal{B}_\epsilon^p(x_i)} \ell(f_{w+v}(x_i'), y_i), \tag{5}$$

where v is the adversarial weight perturbation, which is generated by maximizing the classification loss:

$$v = \nabla_w \sum_i \ell_i. \tag{6}$$

In order to increase the loss of small-loss data. We adopt the weight perturbation technique to generate perturbation noise for the small-loss data in a targeted manner. Similarly, since we aim to eliminate robust overfitting, we leverage the minimum loss conditions ℓ_{min} as a constraint and generate the perturbation noise v as follows:

$$v = \nabla_w \sum_i \mathbb{1}(\ell_i \le \ell_{min})\, \ell_i, \tag{7}$$

where $\mathbb{1}(\ell_i \le \ell_{min})$ is an indicator function, which will output 1 if $\ell_i \le \ell_{min}$ and 0 if $\ell_i > \ell_{min}$. After obtaining the perturbation noise v, we scale the perturbation noise according to the norm of w to get the final weight perturbation $v = \gamma \frac{||w||}{||v||} v$, where γ is the weight perturbation size. Then, the adjusted loss of small-loss data can be expressed as:

$$\ell_i^S = \ell(f_{w+v}(x_i'), y_i). \tag{8}$$

Note that although weight perturbation technique can increase the loss of these small-loss data, it can not guarantee that they all satisfy the minimum loss condition ℓ_{min}, as the goal of Eq.(7) is to maximize the overall loss of the small-loss data in the entire mini-batch rather than instance level. Anyway, the perturbation noise generated by Eq.(7) will effectively prevent the network from fitting these small-loss data. Therefore, *weight perturbation* can be regarded as preventing robust overfitting by manipulating the model parameters, which is different from *loss scaling* and is orthogonal in implementation. We refer the implementation method based on weight perturbation as MLCAT$_{\text{WP}}$.

5. Experiment

In this section, we conduct extensive experiments to verify the effectiveness of MLCAT$_{\text{LS}}$ and MLCAT$_{\text{WP}}$ including their experimental settings (Section 5.1), performance evaluation (Section 5.2) and ablation studies (Section 5.3).

5.1. Experimental Settings

Our implementation is based on PyTorch and the code is publicly available[1]. We conduct extensive experiments across three benchmark datasets (CIFAR10 (Krizhevsky et al., 2009), SVHN (Netzer et al., 2011) and CIFAR100 (Krizhevsky et al., 2009)) and two threat models (L_∞ and L_2). We use PreAct ResNet-18 (He et al., 2016) and Wide ResNet-34-10 (Zagoruyko & Komodakis, 2016) as the network structure following (Rice et al., 2020). For training, the model 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, respectively. Standard data augmentation including random crops with 4 pixels of padding and random horizontal flips are applied for CIFAR10 and CIFAR100, and no data augmentation is used on SVHN. For adversary, 10-step PGD attack is applied: for L_∞ threat model, perturbation size $\epsilon = 8/255$, step size $\alpha = 1/255$ for SVHN, and $\alpha = 2/255$ for both CIFAR10 and CIFAR100; for L_2 threat model, perturbation size $\epsilon = 128/255$, step size $\alpha = 15/255$ for all datasets, which is a standard setting for PGD-based adversarial training (Madry et al., 2017). For testing, model robustness is evaluated by measuring the accuracy on test data under different adversarial attacks, including 20-step PGD (PGD-20) (Madry et al., 2017) and Auto Attack (AA)

[1] https://github.com/ChaojianYu/Understanding-Robust-Overfitting

Table 1. Test robustness (%) on CIFAR10. We omit the standard deviations of 5 runs as they are very small (< 0.6%).

Network	Threat Model	Method	PGD-20			AA		
			Best	Last	Diff	Best	Last	Diff
PreAct ResNet-18	L_∞	AT	52.29	44.43	-7.86	47.99	42.08	-5.91
		MLCAT$_{\text{LS}}$	56.90	56.87	**-0.03**	28.12	26.93	-1.19
		MLCAT$_{\text{WP}}$	**58.48**	**57.65**	-0.83	**50.70**	**50.32**	**-0.38**
	L_2	AT	69.27	65.86	-3.41	67.70	64.64	-3.06
		MLCAT$_{\text{LS}}$	73.16	72.48	-0.68	49.7	48.94	-0.76
		MLCAT$_{\text{WP}}$	**74.38**	**73.86**	**-0.52**	**70.46**	**70.15**	**-0.31**
Wide ResNet-34-10	L_∞	AT	55.57	47.37	-8.20	52.13	46.09	-6.04
		MLCAT$_{\text{LS}}$	**64.73**	**63.94**	-0.79	35.00	34.51	-0.49
		MLCAT$_{\text{WP}}$	62.50	61.91	**-0.59**	**54.65**	**54.56**	**-0.09**
	L_2	AT	71.57	69.99	-1.58	70.44	68.92	-1.52
		MLCAT$_{\text{LS}}$	75.05	74.97	**-0.08**	55.31	55.11	**-0.20**
		MLCAT$_{\text{WP}}$	**76.92**	**76.55**	-0.37	**74.35**	**73.97**	-0.38

Table 2. Test robustness (%) on SVHN. We omit the standard deviations of 5 runs as they are very small (< 0.6%).

Network	Threat Model	Method	PGD-20			AA		
			Best	Last	Diff	Best	Last	Diff
PreAct ResNet-18	L_∞	AT	52.88	45.29	-7.59	45.09	40.36	-4.73
		MLCAT$_{\text{LS}}$	**64.28**	**62.30**	**-1.98**	34.48	32.33	-2.15
		MLCAT$_{\text{WP}}$	60.34	57.79	-2.55	**51.90**	**49.76**	**-2.14**
	L_2	AT	66.68	64.75	-1.93	63.55	62.14	-1.41
		MLCAT$_{\text{LS}}$	**75.32**	**74.06**	-1.26	53.35	52.29	-1.06
		MLCAT$_{\text{WP}}$	72.58	71.59	**-0.99**	**67.21**	**66.27**	**-0.94**
Wide ResNet-34-10	L_∞	AT	55.72	50.44	-5.28	48.00	42.41	-5.59
		MLCAT$_{\text{LS}}$	**78.96**	**77.38**	-1.58	34.66	34.21	-0.45
		MLCAT$_{\text{WP}}$	63.18	61.71	**-1.47**	**54.29**	**53.93**	**-0.36**
	L_2	AT	67.29	65.18	-2.11	62.88	61.06	-1.82
		MLCAT$_{\text{LS}}$	**85.00**	**83.47**	-1.53	55.74	54.15	-1.59
		MLCAT$_{\text{WP}}$	75.43	74.08	**-1.35**	**68.91**	**68.12**	**-0.79**

(Croce & Hein, 2020b). AA is regarded as the most reliable robustness evaluation to date, which is an ensemble of complementary attacks, consisting 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)). The degree of robust overfitting is evaluated by the robust accuracy gap during training. For hyperparameter in MLCAT$_{\text{LS}}$ and MLCAT$_{\text{WP}}$, we set the minimum loss conditions $\ell_{min} = 1.5$ for CIFAR10 and SVHN, and $\ell_{min} = 4.0$ for CIFAR100. Other hyperparameters of the baselines are configured as per their original papers.

5.2. Performance Evaluation

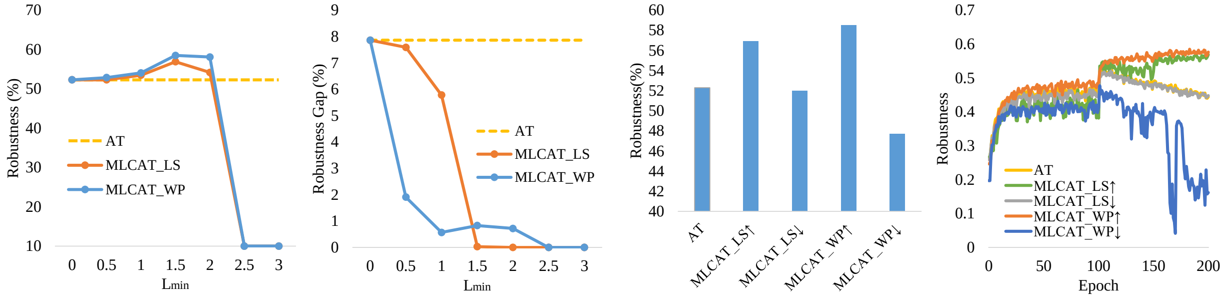
In this part, we report the experimental results of MLCAT$_{\text{LS}}$ and MLCAT$_{\text{WP}}$, and more experimental results are provided in Appendix C.1.

CIFAR10 Results. The evaluation results on CIFAR10 dataset are summarized in Table 1, where "Best" is the highest robustness that ever achieved during training; "Last" is the test robustness at the last epoch checkpoint; "Diff" denotes the robust accuracy gap between the "Best" and "Last". First, it is observed that both MLCAT$_{\text{LS}}$ and MLCAT$_{\text{WP}}$ achieve superior robustness performance under PGD-20 attack. Then, for AA attack, MLCAT$_{\text{WP}}$ can still boost adversarial robustness, while MLCAT$_{\text{LS}}$ achieves the worse robustness performance than AT. This is because loss scaling technique make network sensitive to the logit scaling attack (Hitaj et al., 2021). Finally and most importantly, MLCAT$_{\text{LS}}$ and MLCAT$_{\text{WP}}$ significantly narrow the robustness gaps under both PGD-20 attack and AA attack, which indicates they can effectively eliminate robust overfitting in adversarial training across different network architectures and threat models.

SVHN Results. We further report results on the SVHN dataset, which are summarized in Table 2. Experimental results show that the proposed method improve adversarial robustness and narrow the robustness gap by a large margin under both PGD-20 attack and AA attack, demonstrating

Table 3. Test robustness (%) on CIFAR100. We omit the standard deviations of 5 runs as they are very small (< 0.6%).

Network	Threat Model	Method	PGD-20			AA		
			Best	Last	Diff	Best	Last	Diff
PreAct ResNet-18	L_∞	AT	28.01	20.39	-7.62	23.61	18.41	-5.20
		MLCAT$_{\text{LS}}$	20.09	18.14	-1.95	13.41	11.35	-2.06
		MLCAT$_{\text{WP}}$	**31.27**	**30.57**	**-0.70**	**25.66**	**25.28**	**-0.38**
	L_2	AT	41.38	35.34	-6.04	37.94	33.58	-4.36
		MLCAT$_{\text{LS}}$	31.23	30.80	**-0.43**	22.06	21.72	-0.34
		MLCAT$_{\text{WP}}$	**45.49**	**44.84**	-0.65	**41.22**	**41.15**	**-0.07**
Wide ResNet-34-10	L_∞	AT	30.74	24.89	-5.85	26.98	23.07	-3.91
		MLCAT$_{\text{LS}}$	22.86	22.18	-0.68	14.61	14.05	-0.56
		MLCAT$_{\text{WP}}$	**34.97**	**34.64**	**-0.33**	**29.49**	**29.25**	**-0.24**
	L_2	AT	44.12	41.29	-2.83	41.39	39.34	-2.05
		MLCAT$_{\text{LS}}$	34.09	33.66	**-0.43**	25.06	24.31	-0.75
		MLCAT$_{\text{WP}}$	**50.17**	**49.51**	-0.66	**46.05**	**45.77**	**-0.28**



(a) Impact of minimum loss condition ℓ_{min}

(b) Role of loss adjustment strategy $\mathcal{S}$

Figure 3. The ablation study experiment results on CIFAR10 dataset.

the effectiveness of the proposed MLCAT prototype.

CIFAR100 Results. We also conduct experiments on CIFAR100 dataset. Note that this dataset is more challenging than CIFAR10 as the number of classes/training images per class is ten times larger/smaller than that of CIFAR10. As shown by the results in Table 3, the proposed methods are still able to eliminate robust overfitting and improve adversarial robustness even on more difficult datasets. It verifies that MLCAT prototype eliminates robust overfitting reliably and is general across different datasets, network architectures and threat models.

5.3. Ablation Studies

In this part, we investigate the impacts of algorithmic components using PreAct ResNet-18 on CIFAR10 under L_∞ threat model following the same experimental setting as Section 5.1.

The Impact of Minimum Loss Condition ℓ_{min}. To validate the effectiveness of introducing minimum loss constraint in our MLCAT prototype, we investigate the effect of different ℓ_{min} for the robustness performance and robustness gap (the gap between "best" and "last" robust accuracy). The value of minimum loss condition ℓ_{min} vary from 0 to 3.0, and the results are summarized in Figure 3(a). As expected, increasing ℓ_{min} leads to the smaller robustness gap. For robustness performance, when ℓ_{min} is small, increasing ℓ_{min} leads to the higher robust accuracy than AT. When ℓ_{min} is greater than 1.5, it is observed increasing ℓ_{min} makes model robustness decrease and even leads to the training collapses, implying that the additional measures, such as loss scaling and weight perturbation, are inherently detrimental to robustness improvement. It sheds light on the importance of ℓ_{min}, whose responsibility is to distinguish between small-loss data and large-loss data in the MLCAT prototype. Similar pattern can also be observed in SVHN and CIFAR100 dataset (shown in Appendix C.2). Therefore, we uniformly adopt $\ell_{min} = 1.5$ for CIFAR10 and SVHN, and $\ell_{min} = 4.0$ for CIFAR100 in consideration of the elimination of robust overfitting as well as robustness improvement.

The Role of Loss Adjustment Strategy $\mathcal{S}$. We further investigate the role of loss adjustment strategy within the MLCAT prototype by comparing several different schemes: 1) additive mapping, which increases the loss of small-loss data by Eq.(4) and Eq.(8). We denote them as $\text{MLCAT}_{\text{LS}\uparrow}$ and $\text{MLCAT}_{\text{WP}\uparrow}$, respectively; 2) identical mapping, which keeps the loss of small-loss data unchanged (equivalent to standard AT); 3) subtractive mapping, which decreases the loss of small-loss data. They are implemented by dividing the scaling coefficient $\frac{\ell_{min}}{\ell_i}$ in Eq.(4) and subtracting the weight perturbation v in Eq.(8), which are denoted as $\text{MLCAT}_{\text{LS}\downarrow}$ and $\text{MLCAT}_{\text{WP}\downarrow}$, respectively. Their robustness performance and test accuracy curves are summarized in Figure 3(b). It is observed that decreasing the loss of small-loss data not only fails to suppress robust overfitting but also leads to worse adversarial robustness. In contrast, increasing the loss of small-loss data not only eliminate robust overfitting but also facilitates models to learn these data and further improve adversarial robustness. These comparisons echo our approach's philosophy of turning waste into treasure and making full use of each adversarial data.

6. Conclusion

In this paper, we investigate robust overfitting from the perspective of data distribution and identify that some small-loss data lead to robust overfitting under strong adversary modes. Following this, we propose *minimum loss constrained adversarial training* (MLCAT) prototype. The proposed prototype distinguish itself from others by using additional measures to increase the loss of small-loss data, which prevents the model from fitting these data, and thus effectively avoid robust overfitting. We further provide two specific MLCAT implementations: loss scaling derived from loss correction and weight perturbation derived from parameter correction. Comprehensive experiments show that two realizations of MLCAT can eliminate robust overfitting and improve adversarial robustness across different network architectures, threat models and benchmark datasets.

Acknowledgement

BH is supported by the RGC Early Career Scheme No. 22200720, NSFC Young Scientists Fund No. 62006202, and Guangdong Basic and Applied Basic Research Foundation No. 2022A1515011652. LS is supported by Science and Technology Innovation 2030 –"Brain Science and Brain-like Research" Major Project (No. 2021ZD0201402 and No. 2021ZD0201405). JY is sponsored by CAAI-Huawei MindSpore Open Fund (CAAIXSJLJJ-2021-016B), Anhui Province Key Research and Development Program (202104a05020007). CG is supported by NSF of China (No: 61973162), NSF of Jiangsu Province (No: BZ2021013), and the Fundamental Research Funds for the Central Universities (Nos: 30920032202, 30921013114). MMG is supported by ARC DE210101624. TLL is partially supported by Australian Research Council Projects DE-190101473, IC-190100031, and DP-220102121.

References

Andriushchenko, M., Croce, F., Flammarion, N., and 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.

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

Bai, Y., Zeng, Y., Jiang, Y., Xia, S.-T., Ma, X., and Wang, Y. Improving adversarial robustness via channel-wise activation suppressing. *arXiv preprint arXiv:2103.08307*, 2021.

Cai, Q.-Z., Du, M., Liu, C., and Song, D. Curriculum adversarial training. *arXiv preprint arXiv:1805.04807*, 2018.

Carmon, Y., Raghunathan, A., Schmidt, L., Liang, P., and Duchi, J. C. Unlabeled data improves adversarial robustness. *arXiv preprint arXiv:1905.13736*, 2019.

Chen, C., Zhang, J., Xu, X., Hu, T., Niu, G., Chen, G., and Sugiyama, M. Guided interpolation for adversarial training. *arXiv preprint arXiv:2102.07327*, 2021.

Chen, T., Zhang, Z., Liu, S., Chang, S., and Wang, Z. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2020.

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

Croce, F. and 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, 2020b.

Croce, F., Gowal, S., Brunner, T., Shelhamer, E., Hein, M., and Cemgil, T. Evaluating the adversarial robustness of adaptive test-time defenses. *arXiv preprint arXiv:2202.13711*, 2022.

Dong, Y., Xu, K., Yang, X., Pang, T., Deng, Z., Su, H., and Zhu, J. Exploring memorization in adversarial training. In *International Conference on Learning Representations*, 2022.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gowal, S., Qin, C., Uesato, J., Mann, T., and Kohli, P. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hitaj, D., Pagnotta, G., Masi, I., and Mancini, L. V. Evaluating the robustness of geometry-aware instance-reweighted adversarial training. *arXiv preprint arXiv:2103.01914*, 2021.

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

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lee, S., Lee, H., and Yoon, S. 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.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*, 2019.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Pang, T., Yang, X., Dong, Y., Su, H., and Zhu, J. Bag of tricks for adversarial training. *arXiv preprint arXiv:2010.00467*, 2020.

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

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. Adversarially robust generalization requires more data. *Advances in Neural Information Processing Systems*, 31:5014–5026, 2018.

Song, C., He, K., Lin, J., Wang, L., and Hopcroft, J. E. Robust local features for improving the generalization of adversarial training. In *International Conference on Learning Representations*, 2020.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Uesato, J., Alayrac, J.-B., Huang, P.-S., Stanforth, R., Fawzi, A., and Kohli, P. Are labels required for improving adversarial robustness? *arXiv preprint arXiv:1905.13725*, 2019.

Wang, Y., Ma, X., Bailey, J., Yi, J., Zhou, B., and Gu, Q. On the convergence and robustness of adversarial training. In *ICML*, volume 1, pp. 2, 2019a.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019b.

Wu, B., Pan, H., Shen, L., Gu, J., Zhao, S., Li, Z., Cai, D., He, X., and Liu, W. Attacking adversarial attacks as a defense. *arXiv preprint arXiv:2106.04938*, 2021.

Wu, D., Xia, S.-T., and Wang, Y. Adversarial weight perturbation helps robust generalization. *arXiv preprint arXiv:2004.05884*, 2020.

Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., and Sugiyama, M. Are anchor points really indispensable in label-noise learning? In *NeurIPS*, 2019.

Xia, X., Liu, T., Han, B., Wang, N., Gong, M., Liu, H., Niu, G., Tao, D., and Sugiyama, M. Part-dependent label noise: Towards instance-dependent label noise. In *NeurIPS*, 2020.

Yan, H., Zhang, J., Niu, G., Feng, J., Tan, V. Y., and Sugiyama, M. Cifs: Improving adversarial robustness of cnns via channel-wise importance-based feature selection. *arXiv preprint arXiv:2102.05311*, 2021.

Yu, C., Han, B., Gong, M., Shen, L., Ge, S., Du, B., and Liu, T. Robust weight perturbation for adversarial training. *arXiv preprint arXiv:2205.14826*, 2022.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zhai, R., Cai, T., He, D., Dan, C., He, K., Hopcroft, J., and Wang, L. Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*, 2019.

Zhang, H. and Xu, W. Adversarial interpolation training: A simple approach for improving model robustness. 2019.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pp. 7472–7482. PMLR, 2019.

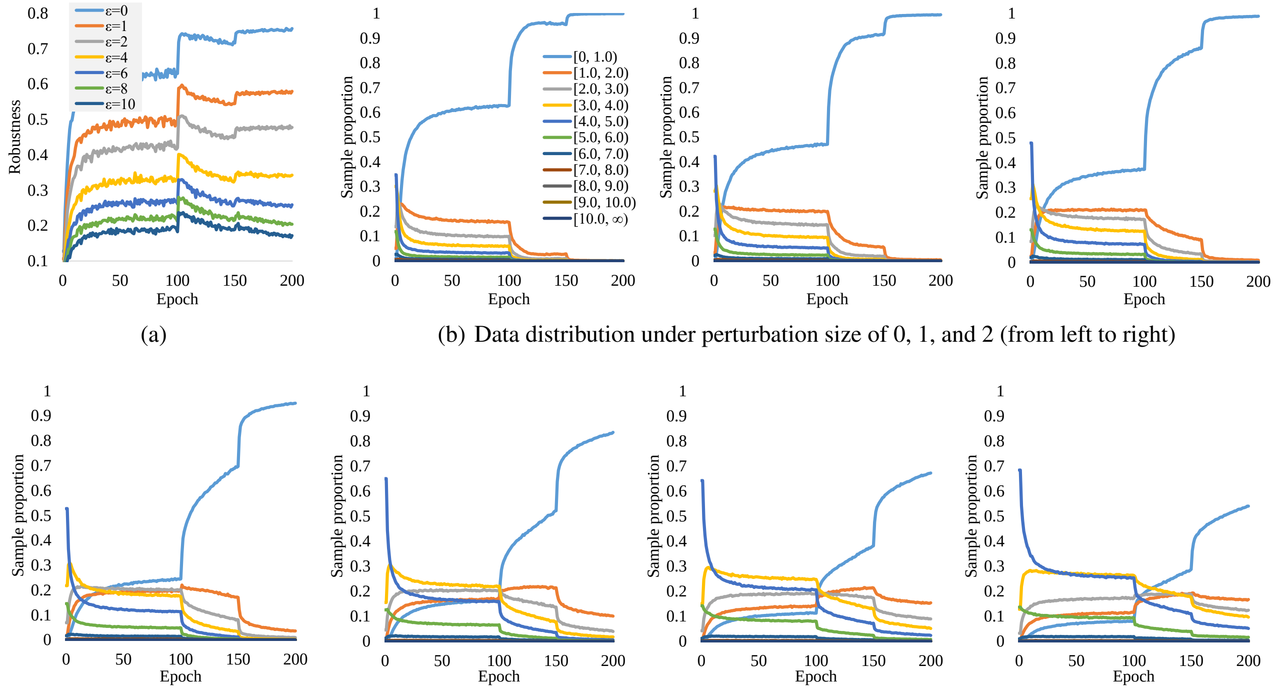
Zhang, J., Xu, X., Han, B., Niu, G., Cui, L., Sugiyama, M., and Kankanhalli, M. Attacks which do not kill training make adversarial learning stronger. In *International Conference on Machine Learning*, pp. 11278–11287. PMLR, 2020a.

Zhang, J., Zhu, J., Niu, G., Han, B., Sugiyama, M., and Kankanhalli, M. Geometry-aware instance-reweighted adversarial training. *arXiv preprint arXiv:2010.01736*, 2020b.

Zhou, D., Wang, N., Han, B., and Liu, T. Modeling adversarial noise for adversarial defense. *arXiv preprint arXiv:2109.09901*, 2021.

A. More Evidences for Robust Overfitting and Data Distribution

In this section, we provide more empirical evidences for the robust overfitting behaviors and their data distributions across different datasets, model architectures and threat models. We use the same strategy in Section 3.1 to adjust the strength of adversary. Specifically, for L_∞ threat model, we vary ϵ from 0, 1, 2, 4, 8 to 10; for L_2 threat model, we vary ϵ from 0, 16, 32, 64, 128 to 160. As shown in Figure 4 to Figure 7, we can always observe that there is no robust overfitting when the adversary is weak, and the robust overfitting phenomenon is particularly significant when the adversary is strong. Moreover, it can be seen that the data distribution of adversarial training with weak adversary mainly contains small-loss data, and the data distribution of adversarial training with strong adversary usually contains a considerable proportion of small-loss data and large-loss data. These evidences suggest that the observed robust overfitting behaviors and data distributions are general in adversarial training.



(a)

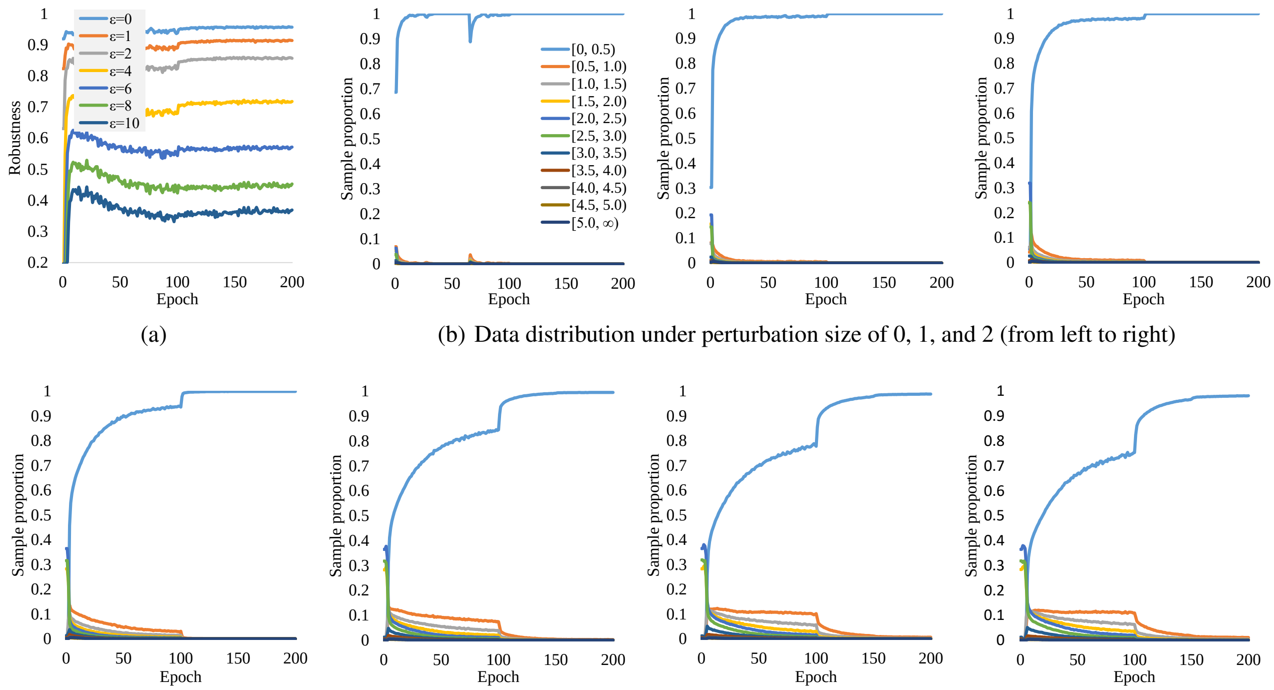
(b) Data distribution under perturbation size of 0, 1, and 2 (from left to right)

(c) Data distribution under perturbation size of 4, 6, 8, and 10 (from left to right)

Figure 4. Robust overfitting behaviors and data distribution on CIFAR100 using PreAct ResNet-18 under L_∞ threat model. (a): The test robustness of adversarial training with various perturbation size ϵ; (b) and (c): The distribution of training data in different loss ranges under various perturbation size ϵ.

B. More Evidences for the Causes of Robust Overfitting

In this section, we further provide more evidences to verify that the small-loss data causes robust overfitting in strong adversary mode. We conduct data ablation adversarial training experiments across different datasets, network architectures and threat models. Specifically, we use $\epsilon = 8$ for L_∞ threat model and $\epsilon = 128$ for L_2 threat model. We remove training data from various loss ranges during adversarial training. As shown in Figure 8, robust overfitting phenomenon is basically unchanged after removing the large-loss data. However, the robust overfitting phenomenon can be eliminated after removing the small-loss data. These evidences clearly show that the robust overfitting in adversarial training is caused by these small-loss data.



(a)

(b) Data distribution under perturbation size of 0, 1, and 2 (from left to right)

(c) Data distribution under perturbation size of 4, 6, 8, and 10 (from left to right)

Figure 5. Robust overfitting behaviors and data distribution on SVHN using PreAct ResNet-18 under L_∞ threat model. (a): The test robustness of adversarial training with various perturbation size ϵ; (b) and (c): The distribution of training data in different loss ranges under various perturbation size ϵ.

C. More Experimental Results

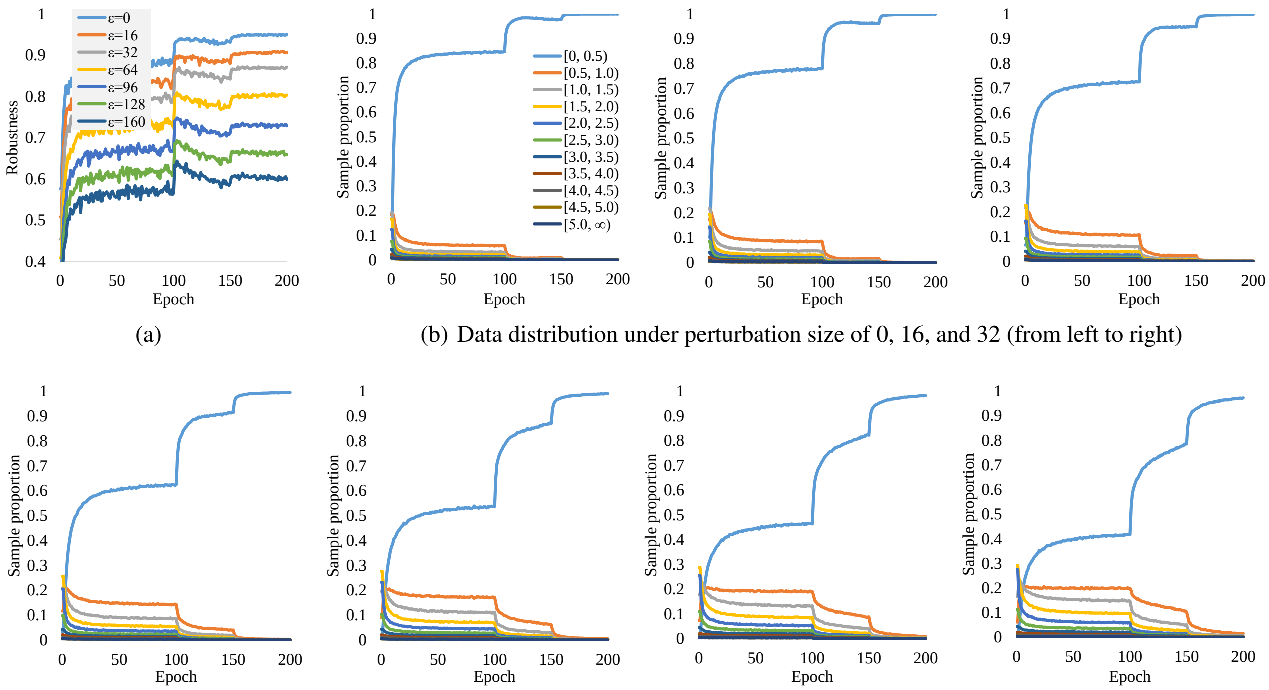
C.1. Performance Evaluation

In this part, we provide more performance evaluations of MLCAT$_{\text{LS}}$ and MLCAT$_{\text{WP}}$ on CIFAR10 dataset using PreAct ResNet-18 under L_∞ threat model.

Natural Accuracy. The natural accuracy of AT, MLCAT$_{\text{LS}}$ and MLCAT$_{\text{WP}}$ are summarized in Table 4. It is observed that both MLCAT$_{\text{LS}}$ and MLCAT$_{\text{WP}}$ can achieve a fairly small performance gap between “Best” and ”Last” on natural accuracy. Notably, MLCAT$_{\text{WP}}$ is able to maintain the comparable natural accuracy to AT.

Extension of MLCAT to TRADES. We extend the proposed prototype to another well-recognized adversarial training variant TRADES. Specifically, for MLCAT-based TRADES (MLCTRADES), the inner maximization pass and outer minimization pass are in accordance with the TRADES method. In MLCTRADES$_{\text{LS}}$ and MLCTRADES$_{\text{WP}}$, we adopt the same $\ell_{min} = 1.5$ to distinguish small-loss data from large-loss data. As shown by the results in Table 5, it is evident that the proposed prototype significantly narrows robustness gap and MLCTRADES$_{\text{WP}}$ outperforms the baseline method with a clear margin, demonstrating its effectiveness.

Comparison with AWP. Although both methods use the weight perturbation technique, MLCAT$_{\text{WP}}$ and AWP are fundamentally different. First, their optimization objectives are different. MLCAT adopts an implicit adversarial example scheduling technique to eliminate robust overfitting, while AWP adopts the weight loss landscape. Besides, the algorithm stability of MLCAT$_{\text{WP}}$ is better than AWP. MLCAT$_{\text{WP}}$ can work on both global and layer-wise perturbation scaling, while AWP suffers from training collapse on global perturbation scaling. Last but not least, we perform robustness comparison between MLCAT$_{\text{WP}}$ and AWP, and the comparison results are summarized in Table 6. MLCAT$_{\text{WP}}$ consistently outperforms AWP on all types of attacks, which fully demonstrates that our MLCAT can avoid robust overfitting and boost the robustness of adversarial training.



(a)

(b) Data distribution under perturbation size of 0, 16, and 32 (from left to right)

(c) Data distribution under perturbation size of 64, 96, 128, and 160 (from left to right)

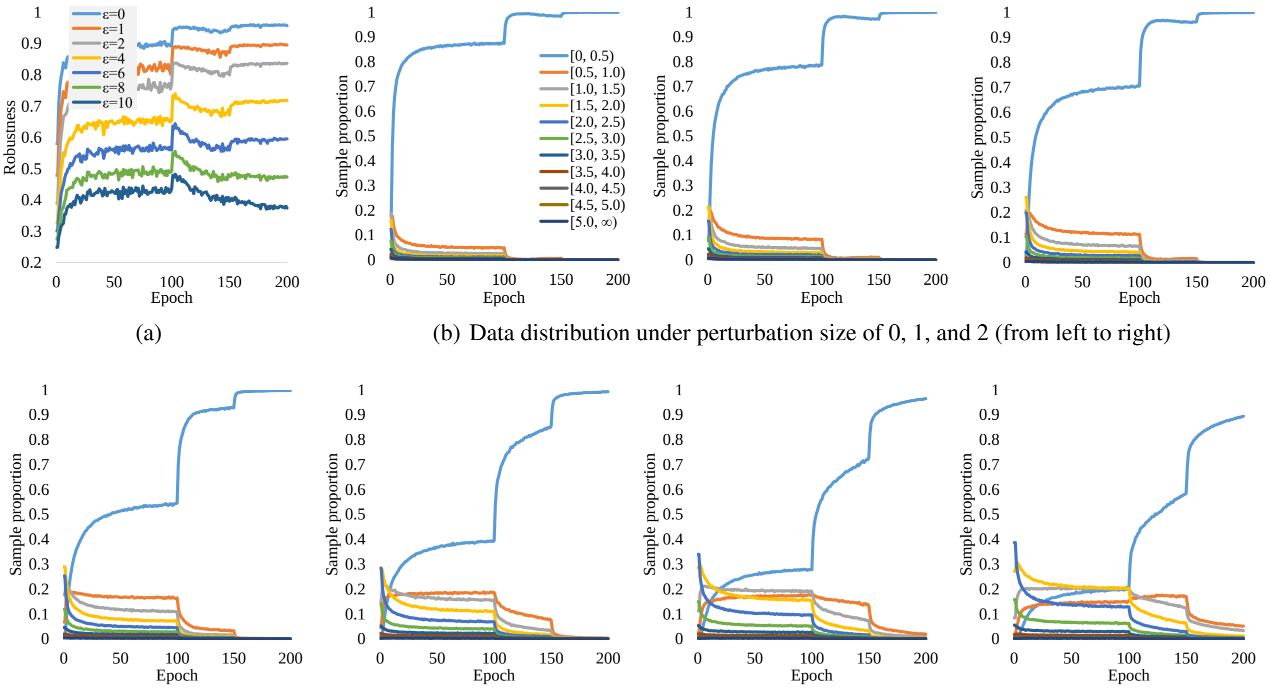
Figure 6. Robust overfitting behaviors and data distribution on CIFAR10 using PreAct ResNet-18 under L_2 threat model. (a): The test robustness of adversarial training with various perturbation size ϵ; (b) and (c): The distribution of training data in different loss ranges under various perturbation size ϵ.

Table 4. Natural accuracy (%) of AT, MLCAT$_{\text{LS}}$ and MLCAT$_{\text{WP}}$.

Method	Natural		
	Best	Last	Diff
AT	82.11 ± 0.45	84.72 ± 0.85	2.61
MLCAT$_{\text{LS}}$	78.73 ± 0.77	79.48 ± 0.91	0.75
MLCAT$_{\text{WP}}$	**84.1** ± **0.23**	**84.77** ± **0.35**	**0.67**

C.2. Ablation Studies

In this part, we provide the complete experimental results of ablation studies about the impact of minimum loss condition ℓ_{min} on CIFAR100 and SVHN datasets. Specifically, we vary the value of ℓ_{min} from 0 to 5.0 for CIFAR100, and from 0 to 3.0 for SVHN. The experimental results for the robustness performance and robustness gap are summarized in Figure 9. It is observed that increasing ℓ_{min} consistently leads smaller robustness gap on CIFAR100 and SVHN datasets, and MLCAT$_{\text{WP}}$ with a wide range of ℓ_{min} achieves better adversarial robustness than AT, demonstrating the importance of minimum loss condition ℓ_{min} in the MLCAT prototype.



(a)

(b) Data distribution under perturbation size of 0, 1, and 2 (from left to right)

(c) Data distribution under perturbation size of 4, 6, 8, and 10 (from left to right)

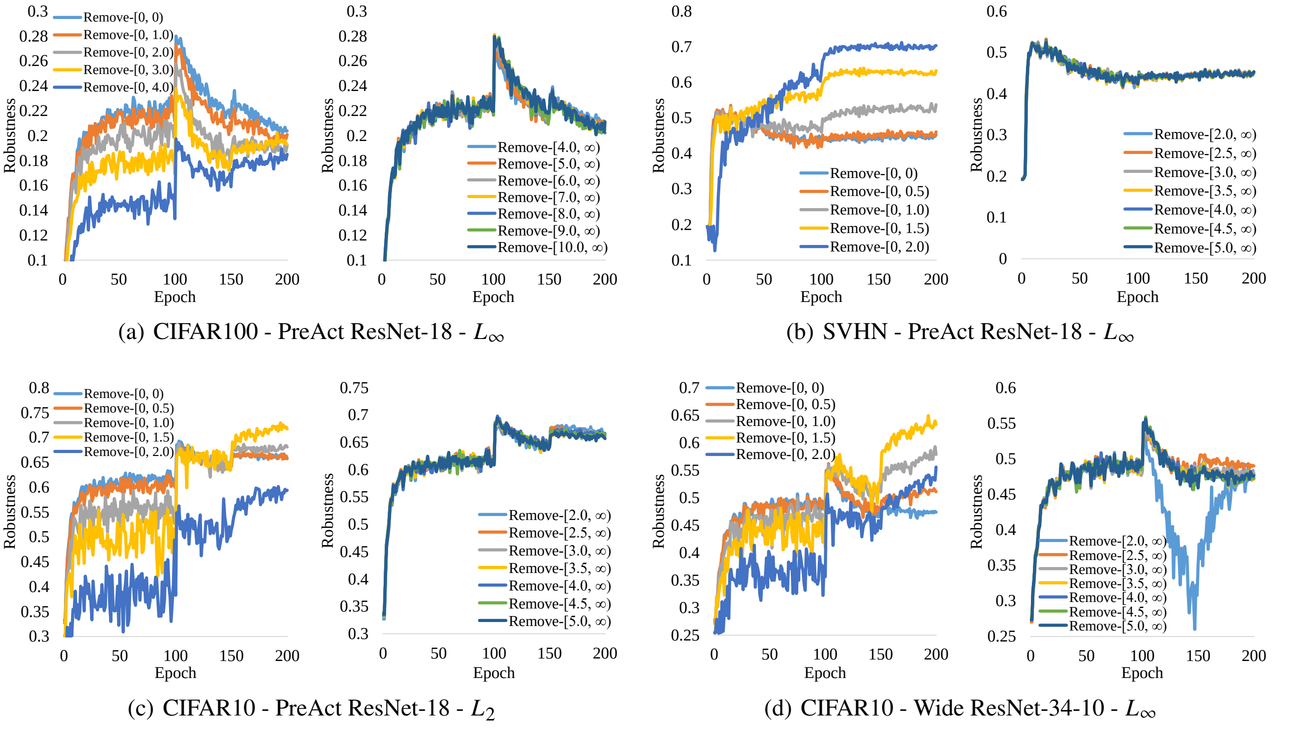
Figure 7. Robust overfitting behaviors and data distribution on CIFAR10 using Wide ResNet-34-10 under L_∞ threat model. (a): The test robustness of adversarial training with various perturbation size ϵ; (b) and (c): The distribution of training data in different loss ranges under various perturbation size ϵ.

Table 5. Test robustness (%) under PGD-20 attack on TRADES.

Method	PGD20		
	Best	Last	Diff
TRADES	52.56 ± 0.43	49.12 ± 0.39	-3.53
MLCTRADES$_{LS}$	42.82 ± 0.25	41.4 ± 0.38	-1.42
MLCTRADES$_{WP}$	**55.28** ± **0.21**	**54.99** ± **0.19**	**-0.29**

Table 6. Robustness comparison with AWP.

Method	PGD20			AA		
	Best	Last	Diff	Best	Last	Diff
AWP	55.54 ± 0.20	54.64 ± 0.25	-0.9	49.94 ± 0.08	49.69 ± 0.10	**-0.25**
MLCAT$_{WP}$	**58.48** ± **0.39**	**57.65** ± **0.19**	**-0.83**	**50.70** ± **0.11**	**50.32** ± **0.09**	-0.38



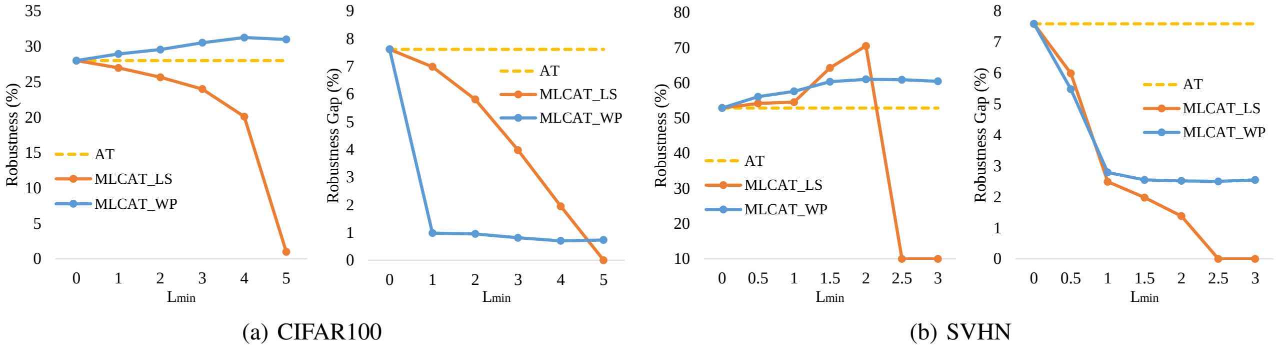
(a) CIFAR100 - PreAct ResNet-18 - L_∞

(b) SVHN - PreAct ResNet-18 - L_∞

(c) CIFAR10 - PreAct ResNet-18 - L_2

(d) CIFAR10 - Wide ResNet-34-10 - L_∞

Figure 8. The data ablation experimental results on (a) CIFAR100 dataset using PreAct ResNet-18 under L_∞ threat model; (b) SVHN dataset using PreAct ResNet-18 under L_∞ threat model; (c) CIFAR10 dataset using PreAct ResNet-18 under L_2 threat model; (d) CIFAR10 dataset using Wide ResNet-34-10 under L_∞ threat model.



(a) CIFAR100

(b) SVHN

Figure 9. The experiment results of ablation study about minimum loss condition ℓ_{min} on (a) CIFAR100 dataset; (b) SVHN dataset.