Mitigating Catastrophic Overfitting in Fast Adversarial Training via Label Information Elimination

Chao Pan[1,2], Ke Tang[1], Qing Li[2], Xin Yao[3]*
[1]Southern University of Science and Technology, Shenzhen 518055, China
[2]The Hong Kong Polytechnic University, Hong Kong, China
[3]Lingnan University, Hong Kong, China
11930665@mail.sustech.edu.cn

Abstract

Fast Adversarial Training (FAT) employs the single-step Fast Gradient Sign Method (FGSM) to generate adversarial examples, reducing the computational costs of traditional adversarial training. However, FAT suffers from Catastrophic Overfitting (CO), where models' robust accuracy against multi-step attacks plummets to zero during training. Recent studies indicate that CO occurs because single-step adversarial perturbations contain label information that models exploit for prediction, leading to overfitting and diminished robustness against more complex attacks. In this paper, we discover that after CO occurs, the label information of certain samples can transfer across different samples, significantly increasing the likelihood of modified images being classified as the intended label. This discovery offers a new perspective on why various adversarial initialization strategies are effective. To address this issue, we introduce an innovative FAT strategy that leverages special samples to capture transferable label information and proactively removes potential label information during training, complemented by a non-uniform label smoothing technique to further eliminate label information. Experimental results across three datasets demonstrate that our method maintains competitive robustness against several attacks compared to other FAT approaches, with ablation studies confirming the effectiveness of our methodology. The code is available at https://github.com/fzjcdt/LIET.

1. Introduction

Fast Adversarial Training (FAT) has gained popularity in adversarial machine learning as an efficient alternative to traditional methods. By generating Adversarial Examples (AEs) using single-step gradient propagation, FAT significantly reduces computational costs compared to multi-step approaches [3, 7, 14, 28]. However, this efficiency comes at a cost: FAT is highly susceptible to Catastrophic Overfitting (CO), a phenomenon where the model suddenly loses its robustness against multi-step adversarial attacks, such as those generated by the Projected Gradient Descent (PGD) method [7, 16, 18, 24, 28]. The contrast between increasing single-step robust accuracy and the sudden drop to zero of multi-step robust accuracy exposes a critical vulnerability in FAT.

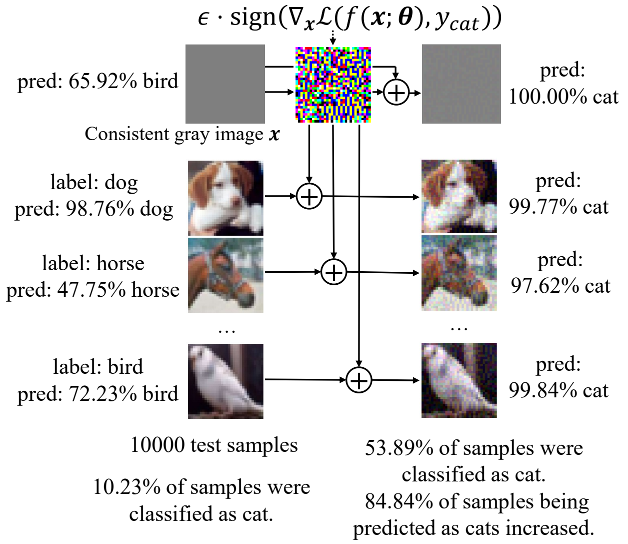

$\epsilon \cdot \mathrm{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(f(\boldsymbol{x}; \boldsymbol{\theta}), y_{cat}))$
pred: 65.92% bird
Consistent gray image $\boldsymbol{x}$
pred: 100.00% cat
label: dog
pred: 98.76% dog
pred: 99.77% cat
label: horse
pred: 47.75% horse
pred: 97.62% cat
...
...
label: bird
pred: 72.23% bird
pred: 99.84% cat
10000 test samples
10.23% of samples were classified as cat.
53.89% of samples were classified as cat.
84.84% of samples being predicted as cats increased.

Figure 1. Label information transferability post-catastrophic overfitting. A ResNet18 model, FAT-trained on CIFAR10 and experiencing CO, was tested. A single-step perturbation targeting "cat" (y_{cat}), generated from a gray image (value 0.5), was applied to 10,000 test images. "Cat" classification surged from 10.23% to 53.89%. This highlights that single-step perturbations after CO encode **transferable** label information.

*Corresponding Author.

Recent investigations into the nature of adversarial examples generated during CO have led to a pivotal discovery: the decoupling of single-step adversarial perturbations into data-information and self-information components [11]. This study reveals that the label information embedded within single-step perturbations plays a dominant role in guiding model predictions post-CO, thus facilitating the network's unintended learning focus on self-information. Consequently, models tend to rely solely on the label information present in single-step AEs for making predictions, thereby compromising their defense capability against multi-step AEs.

In this study, we observed that after experiencing CO, the transferability of label information is remarkably effective across different samples. Utilizing a ResNet18 [10] model trained on the CIFAR10 [15] dataset with standard FAT and subjected to CO, we generated a single-step adversarial perturbation $\boldsymbol{\delta}$ targeting the label y_{cat} for a uniform gray image (value 0.5) using the Fast Gradient Sign Method (FGSM) [7]. Initially, among 10,000 original test images, only 10.23% were classified as "cat". However, after applying $\boldsymbol{\delta}$, a striking 53.89% of the samples were recognized as "cat", marking an 84.84% increase in the likelihood of such classification. This significant shift in classification, as illustrated in Figure 1, underscores the strong transferability of label information embedded within single-step perturbations across a diverse set of images.

Building on this new discovery, we propose a novel perspective to understand why various adversarial initialization strategies [13, 20] within the FAT method are effective. Adversarial initialization, in essence, inadvertently eliminates the label information contained within single-step adversarial perturbations, thereby aiding in the mitigation of CO. The experiments we conducted further validate this viewpoint, demonstrating that the strategic removal of label information at the initialization phase significantly contributes to the robustness and effectiveness of the FAT method against adversarial attacks.

The discovery that label information plays a crucial role in the phenomenon of CO and its notable transferability across different samples has inspired us to introduce a novel FAT strategy. Our method, LIET: Label Information Elimination Training, is designed to proactively eliminate label information from adversarial perturbations during the training process, thereby tackling the underlying cause of CO directly. By implementing non-uniform label smoothing, we further enhance the process of removing label information, establishing a more effective defense mechanism against adversarial attacks.

This work makes several contributions to fast adversarial training:

1. We empirically demonstrate that label information embedded within single-step adversarial perturbations exhibits remarkable transferability across different samples post-CO. This insight sheds light on the intrinsic properties of single-step adversarial perturbations and their impact on model classification behaviors.
2. Building upon this discovery, we propose a novel explanation of why adversarial initialization strategies, commonly employed in FAT, are effective. Through rigorous experimentation, we demonstrate that adversarial initialization inadvertently facilitates the removal of label information from single-step adversarial perturbations, thereby mitigating the risk of CO.
3. Leveraging the above insights, we introduce a new FAT methodology, *LIET: Label Information Elimination Training*, specifically designed to proactively eliminate label information from adversarial perturbations during the training process. We validate the competitiveness of our approach across three datasets, with ablation studies further corroborating the efficacy of our strategy.

2. Related Work

2.1. Adversarial Attack and Training

Adversarial attacks aim at crafting examples that lead to incorrect model predictions by adding perturbations to the input data [8, 9, 19, 21, 23]. The objective is to find a perturbation within a specified range $\boldsymbol{S}$ that maximizes the loss function, thereby increasing the likelihood of misclassification. This process can be formulated as an optimization problem, expressed as:

$$\boldsymbol{\delta}^* = \arg\max_{\boldsymbol{\delta}\in\boldsymbol{S}} \mathcal{L}(f(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta}), y), \tag{1}$$

where $\boldsymbol{\delta}^*$ represents the optimal perturbation aimed at maximizing the loss $\mathcal{L}$, with $\boldsymbol{\theta}$ denoting the model parameters, and y the true label of $\boldsymbol{x}$.

Adversarial Training (AT) seeks to improve model robustness by incorporating adversarial examples into the training process, formulated as a min-max optimization problem [16, 22]:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\boldsymbol{\delta}\in\boldsymbol{S}} \mathcal{L}(f(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta}), y)\right]. \tag{2}$$

2.2. Fast Adversarial Training and Catastrophic Overfitting

Fast adversarial training employs single-step attack methods, such as the FGSM, to generate adversarial examples [7]. However, FAT is vulnerable to catastrophic overfitting, a sudden drop in robust accuracy against adversarial examples generated by multiple adversarial attacks, primarily due to overfitting on specific adversarial patterns generated in early training phases [28].

2.2.1. Random Initialization

To counter CO, random initialization introduces variability in the adversarial examples by starting the perturbation process with a random noise. This approach, including methods like Random Start FGSM (FGSM-RS) and Noise-FGSM (N-FGSM), has shown effectiveness in enhancing model robustness and preventing CO [3, 28].

2.2.2. Adversarial Initialization

Adversarial initialization strategies, including Adversarial Training with Transferable Adversarial Examples (ATTA) [30], Prior-Guided Initialization (PGI) [13], and Universarial Adversarial Perturbation (FGSM-UAP) [20], are designed to commence the adversarial example generation process using inputs that are inherently adversarial.

2.2.3. Regularization

Regularization techniques, like Gradient Alignment Regularization [1] and Nuclear-Norm regularization [27], are employed to stabilize the model's response to adversarial perturbations by aligning gradients and regularizing the output space, thereby enhancing the smoothness of the decision boundary and mitigating CO.

3. Proposed Strategy for Label Information Elimination

This section explores the transferability of label information post-CO, introduces a novel understanding of adversarial initialization, proposes a new FAT method, and discusses the implementation of non-uniform label smoothing as a defense mechanism.

3.1. Transferability of Label Information after Catastrophic Overfitting

This section aims to demonstrate a critical characteristic of single-step adversarial perturbations generated after CO: the transferability of label information. It is important to distinguish this concept from adversarial example transferability, where a single adversarial example can fool multiple models. Here, label information transferability refers to the ability of the label information embedded within one carefully crafted perturbation to influence the classification of different input samples.

To empirically verify this phenomenon, we conducted an experimental study following the settings of He et al. [11]. We trained a ResNet18 [10] model on the CIFAR10 [15] dataset using FGSM-AT.

After obtaining models that had undergone CO, we generated single-step perturbations for various inputs, including uniform gray images with all values set to 0 and 0.5, the training data mean, uniformly distributed noise, and the first sample from the test data. The perturbation for a given class c was computed using the formula:

$$\boldsymbol{\delta_c} = \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(f(\boldsymbol{x};\boldsymbol{\theta}), c)), \tag{3}$$

where $\boldsymbol{x}$ represents the input, c the class, and ϵ the perturbation magnitude.

In order to evaluate the ability of these single-step perturbations to be transferred across different samples, we applied the perturbation $\boldsymbol{\delta_c}$ to N samples from the test set and observed the model's responses. To quantify the transferability of these perturbations, we introduced two specific metrics: $P_{abnormal}$ and $P_{dominate}$. These metrics are defined as follows:

$$P_{abnormal} = \frac{1}{NC} \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbb{1}\big(f(\boldsymbol{x_i} + \boldsymbol{\delta_c})_c > f(\boldsymbol{x_i})_c\big), \tag{4}$$

$$P_{dominate} = \frac{1}{NC} \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbb{1}\big(\text{argmax} f(\boldsymbol{x_i} + \boldsymbol{\delta_c}) = c\big). \tag{5}$$

$P_{abnormal}$ is designed to measure the frequency with which the application of perturbation $\boldsymbol{\delta_c}$ to an input $\boldsymbol{x_i}$ increases the model's confidence in class c. $P_{dominate}$ evaluates the proportion of instances where the perturbation $\boldsymbol{\delta_c}$ not only increases the model's confidence in class c but also makes class c the most probable class. Together, these metrics quantify the effectiveness of transferring label information across samples, with higher values indicating stronger transferability of the adversarial features associated with specific classes after CO.

Input x	$P_{abnormal}$ **(%)**	$P_{dominate}$ **(%)**
Uniform Gray (0)	42.2 ± 32.55	20.29 ± 28.62
Uniform Gray (0.5)	81.81 ± 20.00	44.47 ± 27.44
Training Mean	81.36 ± 20.47	45.45 ± 28.42
Uniform Noise	50.70 ± 2.35	11.08 ± 1.72
Test Sample 0	73.08 ± 18.95	26.06 ± 15.03

Table 1. Transferability of label information for different inputs on the CIFAR-10 dataset with a perturbation size of 16/255.

As evidenced in Table 1, our experiments reveal that single-step perturbations crafted from uniform gray images (value 0.5) and the training data mean exhibit remarkable transferability. This is notable because these inputs are devoid of specific object features. Perturbations from a neutral gray image (0.5) likely capture more generalizable label information, as the model is forced to rely on more fundamental features rather than image-specific details. Similarly, the training data mean, representing a central point in the data distribution, generates perturbations that effectively exploit the model's learned feature space. Consequently, these perturbations are more likely to transfer and mislead the model across diverse inputs compared to those generated from more specific or noisy inputs.

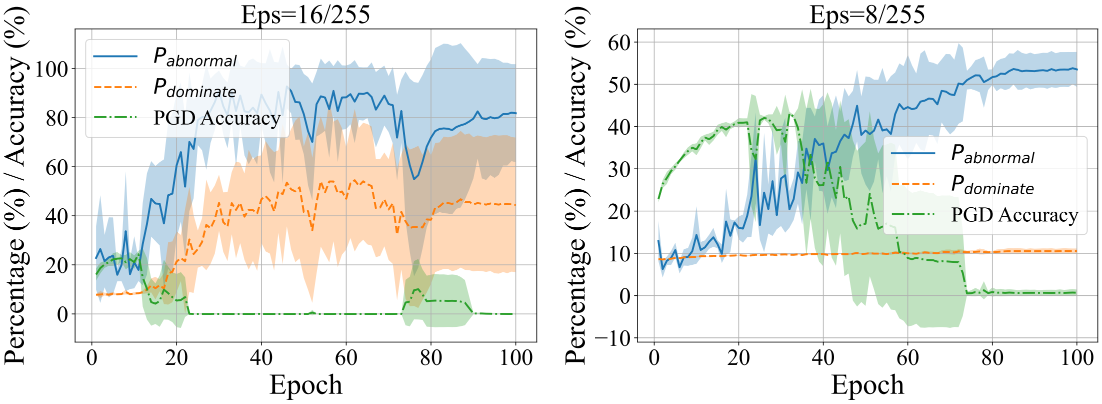


Figure 2. Transferability of label information for uniform gray image (value 0.5) on the CIFAR10 dataset with a perturbation size of 16/255 and 8/255. The shaded regions represent the standard deviation across five experiments.

Additionally, we analyzed the progression of $P_{abnormal}$ and $P_{dominate}$, along with PGD accuracy for uniformly gray images (values set to 0.5). As shown in Figure 2, for $\epsilon = 16/255$, the PGD accuracy decline, signaling CO, is closely linked to a concurrent rise in $P_{abnormal}$ and $P_{dominate}$. This temporal relationship strongly indicates that transferable label information emerges alongside CO and is embedded within single-step adversarial perturbations. For $\epsilon = 8/255$, while $P_{abnormal}$ increases, $P_{dominate}$ remains low, likely due to the perturbation's limited strength in dominating the output. These observations reveal that transferable label information is a byproduct of CO, residing in single-step perturbations. Therefore, developing methods to prevent or remove this label information from single-step perturbations is essential for FAT methods.

Due to page limitations, we provide detailed experimental settings, results for perturbation size $8/255$ in Table 1, and results across five seeds in Figure 2 in Appendix A.

3.2. Revisiting Adversarial Initialization: A Novel Understanding

Building upon the observation that transferable label information, a byproduct of CO, resides within single-step adversarial perturbations, this section explores adversarial initialization strategies in FAT. We propose a novel understanding: their effectiveness stems not merely from enhanced adversarial example quality, but from their unintentional removal of label information.

Adversarial initialization strategies play a pivotal role in FAT, significantly mitigating the risk of CO. These strategies aim to begin the adversarial example generation process with inputs that are already adversarial in nature.

FGSM-PGI [13] employs a momentum mechanism, accumulating historical gradients to guide perturbation generation. Specifically, in epoch $t+1$, the perturbation $\boldsymbol{\delta}_{\boldsymbol{x}}^{t+1}$ for sample $\boldsymbol{x}$ is updated based on momentum $\boldsymbol{m}_{\boldsymbol{x}}^{t+1}$:

$$\boldsymbol{m}_{\boldsymbol{x}}^{t+1} = \mu \cdot \boldsymbol{m}_{\boldsymbol{x}}^{t} + \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(f(\boldsymbol{x} + \boldsymbol{\delta}_{\boldsymbol{x}}^{t}; \boldsymbol{\theta}), y)), \tag{6}$$

$$\boldsymbol{\delta}_{\boldsymbol{x}}^{t+1} = \Pi_{\boldsymbol{S}}(\boldsymbol{\delta}_{\boldsymbol{x}}^{t} + \alpha \cdot \text{sign}(\boldsymbol{m}_{\boldsymbol{x}}^{t+1})). \tag{7}$$

Here, $\boldsymbol{\delta}_{\boldsymbol{x}}^{t}$ from the previous epoch serves as adversarial initialization, incorporating historical gradient information via momentum $\boldsymbol{m}_{\boldsymbol{x}}$.

Similarly, class-based FGSM-UAP [20] initializes each class y with a unique Universal Adversarial Perturbation (UAP) $\boldsymbol{\delta}_{\boldsymbol{y}}$. The update rule for $\boldsymbol{\delta}_{\boldsymbol{y}}$ is:

$$\boldsymbol{m}_{\boldsymbol{y}} = \mu \cdot \boldsymbol{m}_{\boldsymbol{y}} + \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(f(\boldsymbol{x} + \boldsymbol{\delta}_{\boldsymbol{y}}; \boldsymbol{\theta}), y)), \tag{8}$$

$$\boldsymbol{\delta}_{\boldsymbol{y}} = \Pi_{\boldsymbol{S}}(\boldsymbol{\delta}_{\boldsymbol{y}} + \alpha \cdot \text{sign}(\boldsymbol{m}_{\boldsymbol{y}})). \tag{9}$$

In this case, $\boldsymbol{\delta}_{\boldsymbol{y}}$ acts as a class-specific adversarial initialization.

Both FGSM-PGI and FGSM-UAP employ adversarial initialization derived from single-step attacks targeting the true label y. Consequently, the adversarial initializations $\boldsymbol{\delta}_{\boldsymbol{x}}^{t}$ and $\boldsymbol{\delta}_{\boldsymbol{y}}$ inherently contain label information.

A conventional understanding might suggest that adversarial initialization's effectiveness stems from generating higher quality single-step adversarial examples [13, 20]. If this were the sole mechanism, removing adversarial initialization should diminish or even negate its benefits, potentially leading to a degradation in adversarial example quality.

However, our findings, illustrated in Figure 3, challenge this perspective. We observe that both adding and subtracting adversarial initialization prove beneficial. As depicted in the bottom-left panel of Figure 3, subtracting adversarial initialization ensures that regardless of the subsequent FGSM attack direction, the final adversarial example is unlikely to retain label information that contributes to CO.

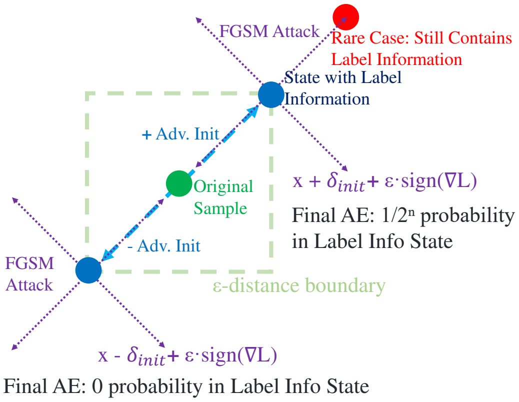

FGSM Attack
Rare Case: Still Contains Label Information
State with Label Information
+ Adv. Init
Original Sample
x + δ_{init}+ ε·sign(∇L)
Final AE: 1/2ⁿ probability in Label Info State
- Adv. Init
FGSM Attack
ε-distance boundary
x - δ_{init}+ ε·sign(∇L)
Final AE: 0 probability in Label Info State

Figure 3. Impact of adversarial initialization (added and subtracted) on the label information component of adversarial examples generated by FGSM attacks.

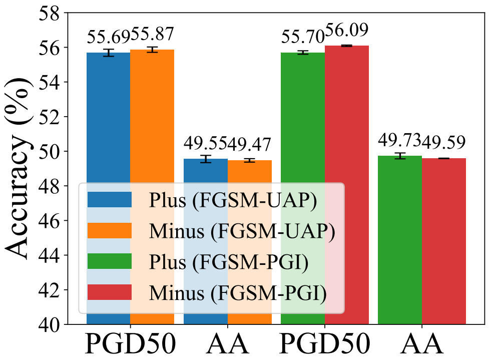


Figure 4. Comparison of robust accuracy between models trained using FGSM-PGI and FGSM-UAP, under plus and minus adversarial initialization scenarios.

Conversely, as shown in the top-right panel of Figure 3, adding adversarial initialization, while starting from a point containing label information, forces the subsequent FGSM attack to move away from this region. This drastically reduces the probability of the final adversarial example still encoding label information, theoretically to $1/2^n$ where n is the number of pixels.

To empirically validate this, we conducted experiments comparing the robust accuracy of models trained with FGSM-PGI [13] and FGSM-UAP [20] under both "plus" and "minus" adversarial initialization conditions. These results, presented in Figure 4 alongside benchmarks against PGD50 [16] and AutoAttack (AA) [2], demonstrate negligible differences in robust accuracy between the "plus" and "minus" initialization scenarios.

This outcome provides compelling evidence that the effectiveness of adversarial initialization goes beyond simply improving single-step AE quality. Instead, we propose a novel interpretation: adversarial initialization's efficacy is significantly attributed to its unintentional removal of label information from single-step adversarial perturbations, offering a fresh perspective on its role in mitigating CO.

3.3. LIET: Label Information Elimination Training

To effectively counteract CO in FAT, leveraging the transferability of label information presents a novel pathway. This approach is predicated on the observation that single-step perturbations generated from a gray image with all values set to 0.5 contain substantial label information, which can be transferred across different samples. By strategically manipulating these perturbations during the training process, it is possible to preemptively eliminate the embedded label information, thereby mitigating the risk of CO.

The process begins with the generation of class-specific, single-step adversarial perturbations, $\boldsymbol{LI_c}$, from a gray image, $\boldsymbol{x_{gray}}$, which inherently carry transferable label information. These perturbations are formulated as follows:

$$\boldsymbol{LI_c} = \epsilon \cdot \mathrm{sign}(\nabla_{\boldsymbol{x_{gray}}} \mathcal{L}(f(\boldsymbol{x_{gray}}; \boldsymbol{\theta}), c)), \tag{10}$$

where ϵ is the perturbation magnitude, $\nabla_{\boldsymbol{x_{gray}}}$ denotes the gradient with respect to $\boldsymbol{x_{gray}}$, $\mathcal{L}$ is the loss function, f represents the model parameterized by $\boldsymbol{\theta}$, and c is the class label.

In the context of FAT, where adversarial examples are generated through single-step attacks, as we demonstrated in the previous section, initializing the training samples, $\boldsymbol{x}$, by either subtracting or adding $\boldsymbol{LI_c}$ effectively removes the label information of y from the generated perturbation.

The adversarial perturbation $\boldsymbol{\delta_x}$ is then defined as:

$$\boldsymbol{\delta_x} = \epsilon \cdot \mathrm{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(f(\boldsymbol{x} \pm \boldsymbol{LI_y}; \boldsymbol{\theta}), y)). \tag{11}$$

To enhance the effectiveness of model training with generated adversarial examples and to ensure a smoother loss surface, we adopt a dual approach. Drawing inspiration from the regularization-based FAT method, we incorporate Jensen-Shannon divergence to promote a smoother loss landscape [1, 27]. This approach systematically minimizes the discrepancy between predictions for $\boldsymbol{x}$ and its adversarially perturbed counterpart $\boldsymbol{x}+\boldsymbol{\delta_x}$, using the Jensen-Shannon divergence [17].

The training process is guided by two loss components. The first component, loss_1, focuses on the standard training objective, ensuring accurate predictions on perturbed inputs:

$$\mathrm{loss}_1 = \mathcal{L}(f(\boldsymbol{x} + \boldsymbol{\delta_x}; \boldsymbol{\theta}), y). \tag{12}$$

The second component, loss_2, employs the Jensen-Shannon divergence to minimize the discrepancy between the model's predictions on the original and the perturbed inputs, thus encouraging prediction consistency:

$$\mathrm{loss}_2 = \lambda \cdot \mathcal{L}_{JSD}(f(\boldsymbol{x}; \boldsymbol{\theta}), f(\boldsymbol{x} + \boldsymbol{\delta_x}; \boldsymbol{\theta})). \tag{13}$$

Method Type	**Method Name**	**PGD-10 (%) ↑**	**PGD-20 (%) ↑**	**PGD-50 (%) ↑**	**AA (%) ↑**	**Training Cost**
Multi-step AT	PGD-AT [16]	53.76 ± 0.18 37.10 ± 0.25	52.83 ± 0.11 28.41 ± 0.09	52.60 ± 0.13 25.80 ± 0.36	48.68 ± 0.12 20.07 ± 0.05	353.70 min 59.87 PFLOPs
	PGD-AT-WA [24]	54.90 ± 0.15 41.09 ± 0.07	54.21 ± 0.15 33.22 ± 0.25	54.08 ± 0.12 31.17 ± 0.21	50.26 ± 0.14 25.09 ± 0.21	358.21 min 59.87 PFLOPs
Random Initialization -based FAT	N-FGSM [3]	53.78 ± 0.13 **39.29 ± 0.45**	53.28 ± 0.07 33.00 ± 1.05	53.17 ± 0.09 31.65 ± 1.36	47.97 ± 0.12 22.50 ± 1.33	74.15 min 10.89 PFLOPs
	FGSM-RS [28]	48.22 ± 7.64 15.11 ± 5.15	47.77 ± 7.46 9.33 ± 6.20	47.68 ± 7.46 5.42 ± 5.12	42.82 ± 6.70 0.00 ± 0.00	74.36 min 10.89 PFLOPs
Adversarial Initialization -based FAT	FGSM-PGI [13]	56.36 ± 0.19 19.02 ± 0.34	55.81 ± 0.13 12.96 ± 0.38	55.70 ± 0.10 8.78 ± 0.28	49.73 ± 0.17 0.11 ± 0.00	99.71 min 10.89 PFLOPs
	FGSM-UAP [20]	56.60 ± 0.03 15.56 ± 1.35	56.10 ± 0.04 9.72 ± 1.11	55.89 ± 0.02 6.03 ± 0.91	49.36 ± 0.11 0.03 ± 0.00	108.71 min 16.33 PFLOPs
Regularization -based FAT	NuAT [27]	55.43 ± 0.08 14.93 ± 0.61	54.79 ± 0.04 7.29 ± 0.63	54.64 ± 0.05 3.51 ± 0.37	**50.04 ± 0.13** 0.13 ± 0.02	127.81 min 16.33 PFLOPs
	Grad-Align [1]	53.52 ± 0.16 27.72 ± 12.53	52.99 ± 0.20 23.09 ± 9.28	52.93 ± 0.20 21.82 ± 8.42	47.99 ± 0.15 15.43 ± 3.94	228.64 min 16.33 PFLOPs
Other FAT Methods	FGSM-AT [7]	38.66 ± 1.92 15.15 ± 2.49	28.91 ± 1.36 10.13 ± 2.89	19.47 ± 1.65 5.98 ± 2.41	0.00 ± 0.00 0.00 ± 0.00	74.14 min 10.89 PFLOPs
	Free-AT [25]	52.23 ± 0.15 31.91 ± 2.17	51.60 ± 0.09 21.66 ± 1.98	51.42 ± 0.08 12.47 ± 0.50	47.43 ± 0.14 0.00 ± 0.00	70.53 min 10.89 PFLOPs
	COAT [14]	53.10 ± 0.22 36.27 ± 6.44	52.38 ± 0.17 24.64 ± 6.94	52.33 ± 0.09 18.17 ± 1.52	37.99 ± 0.31 4.58 ± 3.24	90.03 min 13.61 PFLOPs
	GAT [26]	55.03 ± 0.05 9.48 ± 0.79	54.23 ± 0.20 5.98 ± 2.88	54.05 ± 0.21 4.44 ± 3.94	49.39 ± 0.22 3.14 ± 4.41	126.75 min 16.33 PFLOPs
Ours	LIET	**56.70 ± 0.06** 37.17 ± 2.11	**56.14 ± 0.05** **33.55 ± 1.59**	**56.08 ± 0.07** **33.10 ± 1.44**	50.01 ± 0.09 **25.22 ± 0.41**	101.29 min 10.93 PFLOPs

Table 2. Comparison of robust accuracy and training cost (time in minutes and computation in PFLOPs) on CIFAR-10 using ResNet-18. Each method is evaluated with perturbation sizes of 8/255 (first row) and 16/255 (second row). The best results of the FAT method are highlighted in **bold**.

The overall objective combines these two terms, with λ serving as a balancing parameter between the standard training loss and the loss aimed at ensuring prediction consistency across perturbed and unperturbed inputs. We provide the pseudocode of the LIEF algorithm in Appendix B.

3.4. Non-uniform Label Smoothing

As discussed in the preceding sections, label information is a crucial factor contributing to CO. To further mitigate CO by reducing this label information, we explore Non-uniform Label Smoothing (NLS) as a practical trick. Building upon the idea of standard label smoothing, NLS aims to more effectively obscure label information.

Traditional label smoothing enhances adversarial robustness by softening the target distribution [5, 6]. Given a one-hot encoded label vector $\mathbf{y}$, standard label smoothing produces a smoothed label vector $\hat{\mathbf{y}}$ using the formula:

$$\hat{y}_i = y_i(1-\alpha) + \frac{\alpha}{K}, \tag{14}$$

where $\hat{y}_i$ is the i-th element of the smoothed label vector $\hat{\mathbf{y}}$, y_i is the i-th element of the original one-hot label vector $\mathbf{y}$, α is the smoothing parameter, and K is the number of classes.

Non-uniform Label Smoothing (NLS) is a simple modification to further reduce label information. Instead of uniformly distributing the smoothing probability α across incorrect classes, NLS distributes it non-uniformly. Specifically, for the correct class y, the smoothed probability remains $1-\alpha$. For the incorrect classes, we assign random probabilities r_i such that they sum to α. The non-uniform smoothed label vector $\hat{\mathbf{y}}$ is then defined as:

$$\hat{y}_i = \begin{cases} 1-\alpha & \text{if } i = y \\ r_i & \text{if } i \neq y \end{cases}, \tag{15}$$

where $\sum_{i \neq y} r_i = \alpha$ and y is the index of the correct class in the one-hot label vector $\mathbf{y}$.

This trick of non-uniform distribution introduces additional randomness and uncertainty into the label smoothing process, making it harder for adversarial perturbations to exploit label information and thus further preventing CO.

4. Experimental Study

4.1. Experimental Setup

Our research conducted experiments on three widely used datasets to evaluate the robustness against adversarial attacks, namely CIFAR-10, CIFAR-100 [15], and Tiny ImageNet [4].

Each experiment was replicated three times under different random seeds to improve the reliability of our results. To evaluate the robustness of our model, we subjected it to several adversarial attack methods, including PGD [16] and AutoAttack (AA) [2]. Due to page limitations, detailed experimental configurations can be found in Appendix C.

We have fine-tuned the comparison algorithms, incorporating techniques such as weight averaging [12] and label smoothing [6], among others, to ensure a fair comparison.

4.2. Results and Performance Analysis

In this section, we present the results of our experiments conducted on CIFAR-10, CIFAR-100 [15], and Tiny ImageNet [4] datasets. We compare the performance of our proposed FAT method with existing state-of-the-art FAT methods under various attack scenarios. The complete experimental results across the three datasets, as well as results on other architectures [29], are provided in Appendix D.

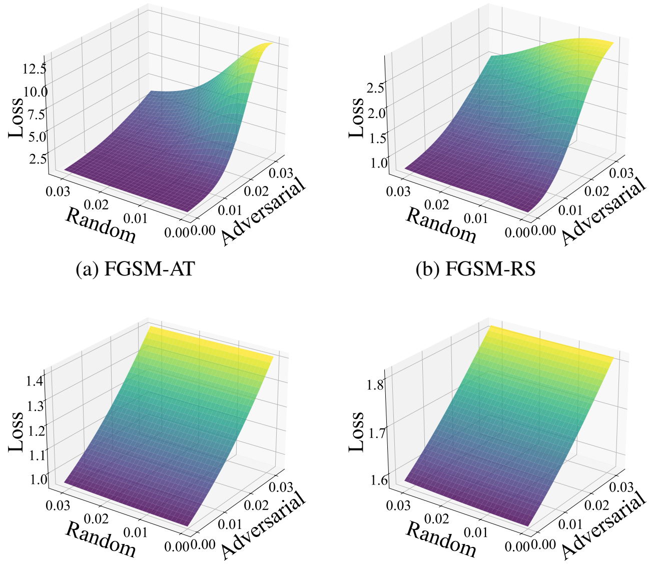
The average results and standard deviations of three runs are shown in Table 6 and Table 7, respectively. Compared to other FAT methods, our approach demonstrates superior robust accuracy under both PGD [16] and Auto Attack [2] scenarios, even reaching levels close to the multi-step adversarial training method PGD-AT [16] and PGD-AT-WA [12]. This indicates the effectiveness of our method in enhancing the robustness of neural networks.

Our model's training time is approximately 1.25 times faster than the best previous FAT methods, including GAT [26] and NuAT [27], and 2.3 times faster than Grad-Align [1]. Furthermore, our training time is comparable to FGSM-PGI [13], yet our performance surpasses it. Notably, FGSM-PGI requires additional memory equivalent to the size of all training data, whereas our method only needs extra memory proportional to the number of classes.

4.3. Loss Landscape Analysis

Figure 5 illustrates the loss landscape of different FAT methods, including FGSM-AT [7], FGSM-RS [28], PGD-AT-WA [12, 16], and our proposed method.

We utilized the first 500 samples from the CIFAR-10 test dataset to plot their cross-entropy loss, as it varies in two dimensions: randomly and along the gradient direction of PGD50. It was observed that for FGSM-AT and FGSM-RS, their loss surfaces are not smooth and exhibit significant variations along the gradient direction, making them still susceptible to attacks. In contrast, PGD-AT-WA and our LIET method demonstrate more linear loss landscapes with minimal variations in loss, indicating that our approach better preserves the local linearity of the target model, thereby enhancing its robustness.



(a) FGSM-AT

(b) FGSM-RS

(c) PGD-AT-WA

(d) LIET (Ours)

Figure 5. Loss landscape comparison of FGSM-AT, FGSM-RS, PGD-AT, and our LIET method by using 500 CIFAR10 test samples, showcasing variations in cross-entropy loss across a random and the PGD50 gradient direction.

4.4. Ablation Experiment

We conducted ablation experiments to validate the effectiveness of our proposed method. As observed from Table 4, initializing with perturbations that contain label information significantly outperforms the initialization using uniform [28] or Bernoulli [27] distributions. Moreover, applying non-uniform label smoothing further enhances the robust accuracy under AutoAttack.

5. Conclusion

In conclusion, our study contributes to the understanding and advancement of fast adversarial training by uncovering the transferability of label information. We not only reveal why adversarial initialization strategies are effective but also introduce a novel FAT methodology, LIET, which proactively eliminates label information from single-step adversarial perturbations to mitigate the risk of CO. Our experimental validations across multiple datasets underscore the effectiveness and competitiveness of our approach.

Method Name	**CIFAR100**			**Tiny-ImageNet**		
	PGD-50 (%)↑	**AA (%)↑**	**Training Cost**	**PGD-50 (%)↑**	**AA (%)↑**	**Training Cost**
PGD-AT [16]	28.87 ± 0.27 11.56 ± 0.16	25.48 ± 0.11 9.17 ± 0.12	353.65 min 59.88 PFLOPs	19.86 ± 1.23 13.15 ± 0.00	16.00 ± 1.02 9.54 ± 0.19	2432.63 min 479.01 PFLOPs
PGD-AT-WA [12]	32.22 ± 0.25 17.45 ± 0.27	26.84 ± 0.28 12.66 ± 0.15	358.80 min 59.88 PFLOPs	26.06 ± 0.34 18.43 ± 0.14	19.62 ± 0.24 12.31 ± 0.02	2439.34 min 479.01 PFLOPs
N-FGSM [3]	30.41 ± 0.29 15.95 ± 0.18	25.31 ± 0.28 11.18 ± 0.21	74.62 min 10.89 PFLOPs	25.10 ± 0.23 16.64 ± 0.14	18.76 ± 0.24 11.11 ± 0.18	498.95 min 87.09 PFLOPs
FGSM-RS [28]	20.84 ± 9.55 1.61 ± 0.32	16.70 ± 7.84 1.05 ± 0.21	74.62 min 10.89 PFLOPs	22.70 ± 1.27 2.25 ± 0.01	16.28 ± 1.03 1.33 ± 0.01	493.77 min 87.09 PFLOPs
FGSM-PGI [13]	32.31 ± 0.11 1.50 ± 1.33	26.76 ± 0.07 0.57 ± 0.70	99.85 min 10.89 PFLOPs	26.39 ± 0.17 4.02 ± 0.14	19.52 ± 0.07 1.33 ± 0.27	652.09 min 87.09 PFLOPs
FGSM-UAP [20]	31.81 ± 0.05 1.35 ± 0.47	26.29 ± 0.03 0.55 ± 0.67	114.03 min 16.33 PFLOPs	25.84 ± 0.04 1.89 ± 0.40	19.45 ± 0.14 0.71 ± 0.59	717.30 min 130.64 PFLOPs
NuAT [27]	21.62 ± 0.58 4.07 ± 0.11	13.77 ± 0.58 1.94 ± 0.12	131.49 min 16.33 PFLOPs	26.37 ± 0.14 10.59 ± 4.58	19.55 ± 0.12 5.47 ± 3.37	865.64 min 130.64 PFLOPs
Grad-Align [1]	31.54 ± 0.23 9.50 ± 5.47	26.18 ± 0.07 6.43 ± 3.93	229.34 min 16.33 PFLOPs	24.21 ± 0.41 14.22 ± 0.72	17.65 ± 0.28 8.71 ± 0.59	1514.23 min 130.64 PFLOPs
FGSM-AT [7]	1.88 ± 0.32 0.82 ± 0.17	0.19 ± 0.03 0.51 ± 0.49	74.17 min 10.89 PFLOPs	16.84 ± 8.55 1.58 ± 0.21	16.60 ± 0.36 0.84 ± 0.29	492.96 min 87.09 PFLOPs
Free-AT [25]	29.96 ± 0.20 9.79 ± 2.21	24.34 ± 0.42 5.84 ± 1.56	71.97 min 10.89 PFLOPs	23.58 ± 0.26 13.99 ± 0.73	16.34 ± 0.32 8.42 ± 0.58	474.87 min 87.09 PFLOPs
COAT [14]	22.70 ± 0.33 4.45 ± 0.61	18.93 ± 0.07 2.75 ± 0.38	88.85 min 13.61 PFLOPs	17.25 ± 0.22 7.38 ± 0.86	11.56 ± 0.02 3.71 ± 0.60	578.07 min 108.87 PFLOPs
GAT [26]	26.54 ± 0.16 0.57 ± 0.06	21.97 ± 0.15 0.06 ± 0.03	126.85 min 16.33 PFLOPs	17.01 ± 0.28 0.60 ± 0.02	11.53 ± 0.15 0.20 ± 0.01	851.30 min 130.64 PFLOPs
LIET (Ours)	**32.74 ± 0.14** **17.22 ± 0.42**	**27.05 ± 0.09** **12.04 ± 0.30**	103.74 min 11.11 PFLOPs	**26.54 ± 0.11** **17.31 ± 0.19**	**19.79 ± 0.04** **11.29 ± 0.15**	682.47 min 90.56 PFLOPs

Table 3. Comparisons of robust accuracy, training time (minutes), and computation (PFLOPs) on CIFAR-100 (using ResNet-18) and Tiny-ImageNet (using PreResNet-18). For each method, the first row shows results with perturbation $\epsilon = 8/255$, while the second row uses $\epsilon = 16/255$ for CIFAR-100 and $\epsilon = 12/255$ for Tiny-ImageNet. The best results of the FAT method are highlighted in **bold**.

Method	**Configuration**			
	Config 1	**Config 2**	**Config 3**	**Config 4**
Initialization Methods:				
Uniform	✓			
Bernoulli		✓		
Label Information (Ours)			✓	✓
Label Smoothing Methods:				
Uniform Label Smoothing			✓	
Non-uniform Label Smoothing (Ours)	✓	✓		✓
Performance Metrics:				
Robust Accuracy (%) at $\varepsilon = 8/255$	5.60	48.29	49.64	**50.01**
Robust Accuracy (%) at $\varepsilon = 16/255$	21.52	20.31	24.59	**25.22**

Table 4. Comparison of robust accuracy under AutoAttack for different initialization methods and the impact of applying non-uniform label smoothing at various perturbation budgets (ε). Best results are highlighted in **bold**.

6. Acknowledgments

We thank the anonymous reviewers for their insightful comments. This work was supported by the National Natural Science Foundation of China under Grant 62250710682, Guangdong Provincial Key Laboratory under Grant 2020B121201001, and the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under Grant 2017ZT07X386. This project has been supported by the Hong Kong Research Grants Council under General Research Fund (project no. 15200023) as well as Research Impact Fund (project no. R1015-23).

References

[1] Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33:16048–16059, 2020. 3, 5, 6, 7, 8, 15, 16, 17, 18

[2] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, pages 2206–2216. PMLR, 2020. 5, 7, 13

[3] Pau de Jorge Aranda, Adel Bibi, Riccardo Volpi, Amartya Sanyal, Philip Torr, Grégory Rogez, and Puneet Dokania. Make some noise: Reliable and efficient single-step adversarial training. *Advances in Neural Information Processing Systems*, 35:12881–12893, 2022. 1, 3, 6, 8, 11, 15, 16, 17, 18

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009. 7

[5] Chaohao Fu, Hongbin Chen, Na Ruan, and Weijia Jia. Label smoothing and adversarial robustness. *arXiv preprint arXiv:2009.08233*, 2020. 6

[6] Morgane Goibert and Elvis Dohmatob. Adversarial robustness via label-smoothing. *arXiv preprint arXiv:1906.11567*, 2019. 6, 7

[7] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 1, 2, 6, 7, 8, 11, 15, 16, 17, 18

[8] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020. 2

[9] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34:4218–4233, 2021. 2

[10] 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*, pages 770–778, 2016. 2, 3, 11, 12

[11] Zhengbao He, Tao Li, Sizhe Chen, and Xiaolin Huang. Investigating catastrophic overfitting in fast adversarial training: A self-fitting perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2313–2320, 2023. 2, 3, 11

[12] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. 7, 8, 12, 17

[13] Zhang Yong Jia, Xiaojun, Xingxing Wei, Baoyuan Wu, Ke Ma, Jue Wang, and Xiaochun Cao. Prior-guided adversarial initialization for fast adversarial training. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 567–584. Springer, 2022. 2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18

[14] Hoki Kim, Woojin Lee, and Jaewook Lee. Understanding catastrophic overfitting in single-step adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8119–8127, 2021. 1, 6, 8, 15, 16, 17, 18

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 3, 7

[16] 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. 1, 2, 5, 6, 7, 8, 13, 15, 16, 17

[17] María Luisa Menéndez, JA Pardo, L Pardo, and MC Pardo. The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318, 1997. 5

[18] Liang-bo Ning, Zeyu Dai, Wenqi Fan, Jingran Su, Chao Pan, Luning Wang, and Qing Li. Joint universal adversarial perturbations with interpretations. *arXiv preprint arXiv:2408.01715*, 2024. 1

[19] Liang-bo Ning, Zeyu Dai, Jingran Su, Chao Pan, Luning Wang, Wenqi Fan, and Qing Li. Interpretation-empowered neural cleanse for backdoor attacks. In *Companion Proceedings of the ACM Web Conference 2024*, pages 951–954, 2024. 2

[20] Chao Pan, Qing Li, and Xin Yao. Adversarial initialization with universal adversarial perturbation: A new approach to fast adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 21501–21509, 2024. 2, 3, 4, 5, 6, 8, 15, 16, 17

[21] Chao Pan, Yu Wu, Ke Tang, Qing Li, and Xin Yao. Efficient robustness evaluation via constraint relaxation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6263–6271, 2025. 2

[22] Tianyu Pang, Min Lin, Xiao Yang, Jun Zhu, and Shuicheng Yan. Robustness and accuracy could be reconcilable by (proper) definition. *arXiv preprint arXiv:2202.10103*, 2022. 2

[23] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021. 2

[24] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020. 1, 6, 15, 16

[25] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32, 2019. 6, 8, 15, 16, 17, 18

[26] Addepalli Sravanti Sriramanan, Gaurang, Arya Baburaj, et al. Guided adversarial attack for evaluating and enhancing adversarial defenses. *Advances in Neural Information Processing Systems*, 33:20297–20308, 2020. 6, 7, 8, 15, 16, 17, 18

[27] Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, et al. Towards efficient and effective adversarial training. *Advances in Neural Information Processing Systems*, 34: 11821–11833, 2021. 3, 5, 6, 7, 8, 15, 16, 17, 18

[28] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 1, 2, 3, 6, 7, 8, 15, 16, 17, 18

[29] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016. 7, 12

[30] Haizhong Zheng, Ziqi Zhang, Juncheng Gu, Honglak Lee, and Atul Prakash. Efficient adversarial training with transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1181–1190, 2020. 3

A. Supplementary Experiments for Section 3.1

This section supplements Section 3.1 of the main paper by providing additional experimental details and results. We present the complete experimental setup described in Section 3.1, results for perturbation size of 8/255 in Table 5, and comprehensive results across five seeds for varying perturbation sizes as illustrated in Figures 6 and 7.

To explore the phenomenon of the transferability of adversarial perturbations following catastrophic overfitting, we engaged in an experimental study based on the settings established by He et al. [11]. We initiated our study by training a ResNet18 [10] model on the CIFAR-10 dataset using the Fast Gradient Sign Method Adversarial Training (FGSM-AT) [7] for 100 epochs. The training was conducted with perturbation sizes (ϵ) set to 8/255 and 16/255, employing a Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 0.1. The learning rate was programmed to diminish by a factor of 0.1 upon reaching the 80th and 90th epochs. Additionally, the model training incorporated a batch size of 128, and the images underwent preprocessing, which included padding of 4 pixels on each side, followed by random cropping and horizontal flipping. To precisely replicate the conditions leading to CO, we adopted zero initialization for generating adversarial samples and set the weight decay to zero. This setup was chosen to maintain consistency with He et al. [11] and to ensure the stable reproduction of CO, thereby facilitating a clear examination of the transferability of adversarial perturbations under these conditions.

We delve into the specifics of the experimental outcomes for each seed, as illustrated in Figures 6 and 7, to shed light on the underlying dynamics of $P_{abnormal}$ and PGD accuracy in relation to catastrophic overfitting. For all five seeds, we observed a gradual increase in $P_{abnormal}$ during the initial stages of training. However, a striking observation was made at the point of CO, where PGD accuracy plummeted to approximately 0, underscoring a sudden and severe degradation in the model's ability to counter adversarial attacks. Correspondingly, $P_{abnormal}$ experienced a sharp escalation, reinforcing the strong linkage between the onset of CO and the dramatic increase in $P_{abnormal}$. This pattern was consistent across different seeds. The detailed analysis for each seed further corroborates the significant impact of CO on the transferability of adversarial perturbations.

B. Pseudocode of the LIET Algorithm for Section 3.3

This section supplements Section 3.3 of the main paper by presenting the pseudocode for the LIET algorithm (Algorithm 1). To enhance clarity, the pseudocode simplifies certain operations. For instance, an implementation detail concerns the clipping of adversarial perturbations. Specifically, for a perturbation budget of $\epsilon = 8/255$, we clip the perturbation to stay within this bound. For larger budgets, however, we adopt the strategy from [3] and omit the clipping step to generate stronger adversaries. For a comprehensive implementation, we refer the reader to the provided source code.

Algorithm 1: LIET: Label Information Elimination Training

```
Input: A classifier $f_\theta$ with loss function $\mathcal{L}$; Dataset
       $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$; Perturbation magnitude
       $\epsilon$; Gray image $\boldsymbol{x_{gray}}$; Hyperparameter $\lambda$;
       Number of epochs $E$.
Output: Robust model parameters $\boldsymbol{\theta}$
for $e = 1$ to $E$ do
    for each class $c$ in dataset do
        $\boldsymbol{LI_c} = \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x_{gray}}} \mathcal{L}(f(\boldsymbol{x_{gray}}; \boldsymbol{\theta}), c))$
          {Generate class-specific label information}
    end
    for each batch $\mathcal{B} = (\boldsymbol{x}, y) \subset \mathcal{D}$ do
        if random() $< 0.5$ then
            $\boldsymbol{x'} = \boldsymbol{x} + \boldsymbol{LI_y}$ {Randomly add label
              information}
        else
            $\boldsymbol{x'} = \boldsymbol{x} - \boldsymbol{LI_y}$ {Randomly subtract
              label information}
        end
        $\boldsymbol{\delta_x} = \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x'}} \mathcal{L}(f(\boldsymbol{x'}; \boldsymbol{\theta}), y))$ {Generate
          adversarial perturbation}
        $\boldsymbol{x_{adv}} = \boldsymbol{x} + \boldsymbol{\delta_x}$ {Create adversarial
          example}
        $\text{loss}_1 = \mathcal{L}(f(\boldsymbol{x_{adv}}; \boldsymbol{\theta}), y)$ {Standard
          adversarial training loss}
        $\text{loss}_2 = \lambda \cdot \mathcal{L}_{JSD}(f(\boldsymbol{x}; \boldsymbol{\theta}), f(\boldsymbol{x_{adv}}; \boldsymbol{\theta}))$ {JS
          divergence for smoother loss surface}
        $\text{total\_loss} = \text{loss}_1 + \text{loss}_2$ {Combined loss
          function}
    end
    $\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \cdot \nabla_{\boldsymbol{\theta}} \text{total\_loss}$ {Update model
      parameters}
end
```

C. Experiment Details for Section 4.1

This section provides supplementary information to Section 4.1 of the main paper. Here, we present detailed experimental configurations and parameters that were utilized in our study but were omitted from the main paper for brevity.

Our research conducted experiments on three widely recognized datasets to evaluate the robustness against adversarial attacks, namely CIFAR-10, CIFAR-100, and

Input x	**Perturbation: 16/255**		**Perturbation: 8/255**	
	$P_{abnormal}$ (%)	$P_{dominate}$ (%)	$P_{abnormal}$ (%)	$P_{dominate}$ (%)
Uniform Gray (0)	42.20 ± 32.55	20.29 ± 28.62	49.89 ± 1.34	10.08 ± 0.16
Uniform Gray (0.5)	81.81 ± 20.00	44.47 ± 27.44	53.52 ± 4.13	10.52 ± 0.62
Training Mean	81.36 ± 20.47	45.45 ± 28.42	53.84 ± 4.31	10.55 ± 0.68
Uniform Noise	50.70 ± 2.35	11.08 ± 1.72	48.67 ± 0.93	9.87 ± 0.10
Test Sample 0	73.08 ± 18.95	26.06 ± 15.03	51.18 ± 0.49	10.12 ± 0.13

Table 5. Transferability of label information for different inputs on the CIFAR-10 dataset with perturbation sizes of 16/255 and 8/255.

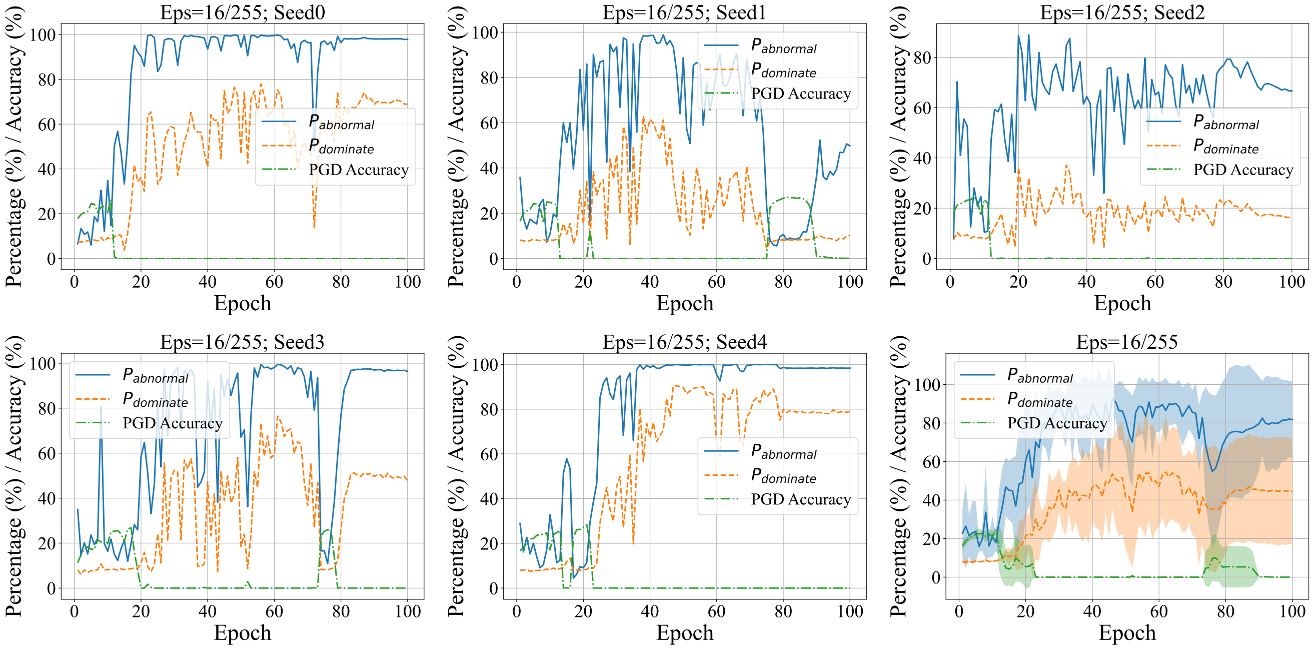


Figure 6. Transferability of label information for uniform gray image (value 0.5) on the CIFAR10 dataset with a perturbation size of 16/255.

Tiny ImageNet. For CIFAR-10 and CIFAR-100 datasets, we employed both the ResNet-18 architecture [10] and WideResNet-28-10 [29] as the network architectures, while for Tiny ImageNet, we opted for PreActResNet18 due to its enhanced performance on more complex datasets.

For the CIFAR-10, CIFAR-100, and Tiny ImageNet datasets, we carved out validation sets comprising 1000, 1000, and 2000 images, respectively, from the training data. During the training phase, we evaluated the model's performance on these validation sets using the PGD-10 accuracy metric. The model that achieved the highest accuracy on the validation set was selected as the final model. This validation strategy was consistently applied across all compared algorithms to maintain uniformity in model evaluation.

We set a batch size of 128 and applied a series of preprocessing steps on the images. These steps included padding the images with 4 pixels on each side, followed by random cropping and horizontal flipping to augment the dataset and improve model generalization.

We utilized the Stochastic Gradient Descent (SGD) as our optimization algorithm, with an initial learning rate set at 0.1, a weight decay parameter of 5e-4, and momentum of 0.9. The training process was conducted over 100 epochs, incorporating a OneCycleLR scheduler to adjust the learning rate dynamically. To stabilize the training process, we implemented a Weight Averaging (WA) [12] technique with a τ value of 0.9995. Each experiment was replicated three times under different random seeds to ensure the reliability of our results. For perturbation magnitude of $8/255$, we set λ values at 100, 200, and 100, respectively. Furthermore, we employed non-uniform label smoothing values of 0.6 for both CIFAR-10 and CIFAR-100, and 0.8 for Tiny ImageNet, to fine-tune the model's performance across diverse datasets. For perturbation magnitude of $16/255$, we set λ to

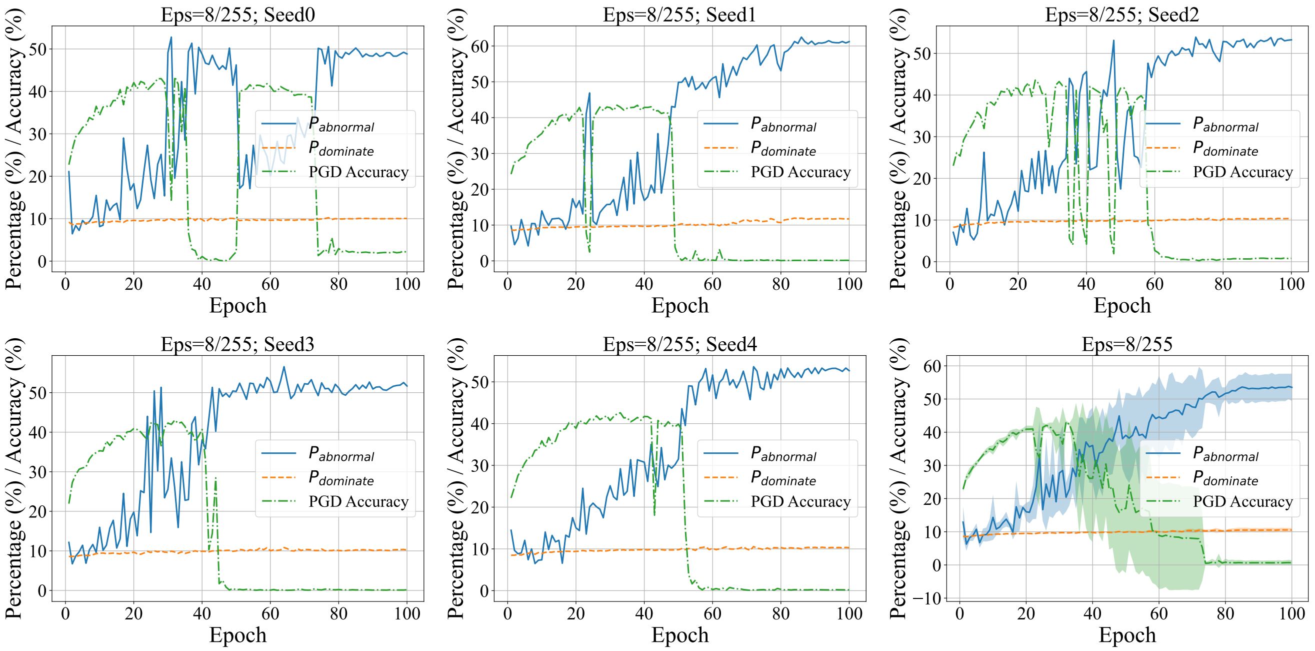


Figure 7. Transferability of label information for uniform gray image (value 0.5) on the CIFAR10 dataset with a perturbation size of 8/255.

20 and label smoothing value to 0.4.

As highlighted in our paper, initializing the training samples, $\boldsymbol{x}$, by either adding or subtracting $\boldsymbol{LI_c}$ proved effective in diminishing the label information, y, from the generated perturbation, $\boldsymbol{\delta_x}$. This initialization strategy was employed randomly to enhance the unpredictability of our defense mechanism against adversarial inputs.

In line with our strategy to boost diversity within the model, we randomly substituted 10% to 50% of the elements in $\boldsymbol{LI_c}$ with values uniformly distributed between $-\epsilon$ and ϵ when the perturbation size was 8/255. For a perturbation size of 16/255, we randomly substituted 0% to 100% of the elements in $\boldsymbol{LI_c}$ with values uniformly distributed between -2ϵ and 2ϵ. This approach was aimed at enriching the robustness of our model against adversarial attacks. To ensure our model adapts to evolving adversarial tactics, we updated $\boldsymbol{LI_c}$ at different intervals depending on the dataset: every 10 batches for CIFAR10 and every 20 batches for CIFAR100 and Tiny-ImageNet, aligning with our strategy to maintain model resilience over time.

To evaluate the robustness of our model, we subjected it to several adversarial attack methods, including PGD [16] and AutoAttack (AA) [2]. We varied the number of iterations for PGD attacks to 10, 20, and 50, which are henceforth referred to as PGD-20, and PGD-50, respectively.

The maximum allowed perturbation was set to 8/255 and 16/255 for CIFAR-10 and CIFAR-100 datasets, respectively. For Tiny-ImageNet, we used perturbation bounds of 8/255 and 12/255. These dataset-specific perturbation settings reflect the different sensitivity levels of each dataset to adversarial attacks and provide a more comprehensive evaluation of our defense mechanism.

D. Experiment Results for Section 4.2

This section supplements the results presented in Section 4.2 of the main paper by providing more detailed experimental findings. Specifically, Tables 6, 7, and 8 display the average results from three sets of experiments: CIFAR-10 and CIFAR-100 using ResNet-18, and Tiny ImageNet using PreActResNet-18. These results include clean accuracy and robust accuracy measured against PGD-10, PGD-20, PGD-50, and AutoAttack. Additionally, Tables 9 and 10 present experimental results for CIFAR-10 and CIFAR-100 using the larger WideResNet-28-10 architecture. Due to the computational demands of this larger model, these experiments were conducted only once.

For the training costs reported in Tables 6, 7, and 8, we measured training time on an NVIDIA V100 GPU. To calculate computational complexity (PFLOPs), we approximated the backward propagation cost as equivalent to the forward propagation cost. The total FLOPs for each method were determined by calculating the number of forward and backward passes required during training.

It is worth noting that while some methods have identical FLOPs calculations in our tables, their training times differ significantly. This discrepancy arises because a substantial portion of training time is consumed by parameter updates, and some methods require maintaining computational graphs in memory, which introduces additional overhead not captured in FLOPs measurements alone.

Method	Clean (%) ↑	PGD-10 (%) ↑	PGD-20 (%) ↑	PGD-50 (%) ↑	AA (%) ↑	Training Cost
PGD-AT [16]	82.32 ± 0.39 70.91 ± 1.37	53.76 ± 0.18 37.10 ± 0.25	52.83 ± 0.11 28.41 ± 0.09	52.60 ± 0.13 25.80 ± 0.36	48.68 ± 0.12 20.07 ± 0.05	353.70 min 59.87 PFLOPs
PGD-AT-WA [24]	82.00 ± 0.38 71.10 ± 0.31	54.90 ± 0.15 41.09 ± 0.07	54.21 ± 0.15 33.22 ± 0.25	54.08 ± 0.12 31.17 ± 0.21	50.26 ± 0.14 25.09 ± 0.21	358.21 min 59.87 PFLOPs
N-FGSM [3]	78.79 ± 0.46 63.09 ± 2.38	53.78 ± 0.13 **39.29 ± 0.45**	53.28 ± 0.07 33.00 ± 1.05	53.17 ± 0.09 31.65 ± 1.36	47.97 ± 0.12 22.50 ± 1.33	74.15 min 10.89 PFLOPs
FGSM-RS [28]	73.06 ± 10.15 66.21 ± 7.19	48.22 ± 7.64 15.11 ± 5.15	47.77 ± 7.46 9.33 ± 6.20	47.68 ± 7.46 5.42 ± 5.12	42.82 ± 6.70 0.00 ± 0.00	74.36 min 10.89 PFLOPs
FGSM-PGI [13]	80.32 ± 1.09 88.06 ± 0.35	56.36 ± 0.19 19.02 ± 0.34	55.81 ± 0.13 12.96 ± 0.38	55.70 ± 0.10 8.78 ± 0.28	49.73 ± 0.17 0.11 ± 0.00	99.71 min 10.89 PFLOPs
FGSM-UAP [20]	79.17 ± 0.27 88.07 ± 0.25	56.60 ± 0.03 15.56 ± 1.35	56.10 ± 0.04 9.72 ± 1.11	55.89 ± 0.02 6.03 ± 0.91	49.36 ± 0.11 0.03 ± 0.00	108.71 min 16.33 PFLOPs
NuAT [27]	80.78 ± 0.55 **91.82 ± 0.12**	55.43 ± 0.08 14.93 ± 0.61	54.79 ± 0.04 7.29 ± 0.63	54.64 ± 0.05 3.51 ± 0.37	**50.04 ± 0.13** 0.13 ± 0.02	127.81 min 16.33 PFLOPs
Grad-Align [1]	78.69 ± 1.13 46.93 ± 26.13	53.52 ± 0.16 27.72 ± 12.53	52.99 ± 0.20 23.09 ± 9.28	52.93 ± 0.20 21.82 ± 8.42	47.99 ± 0.15 15.43 ± 3.94	228.64 min 16.33 PFLOPs
FGSM-AT [7]	**90.98 ± 0.46** 79.75 ± 1.64	38.66 ± 1.92 15.15 ± 2.49	28.91 ± 1.36 10.13 ± 2.89	19.47 ± 1.65 5.98 ± 2.41	0.00 ± 0.00 0.00 ± 0.00	74.28 min 10.89 PFLOPs
Free-AT [25]	81.99 ± 0.95 89.15 ± 0.54	52.23 ± 0.15 31.91 ± 2.17	51.60 ± 0.09 21.66 ± 1.98	51.42 ± 0.08 12.47 ± 0.50	47.43 ± 0.14 0.00 ± 0.00	70.53 min 10.89 PFLOPs
COAT [14]	83.93 ± 0.34 84.58 ± 5.83	53.10 ± 0.22 36.27 ± 6.44	52.38 ± 0.17 24.64 ± 6.94	52.33 ± 0.09 18.17 ± 1.52	37.99 ± 0.31 4.58 ± 3.24	90.03 min 13.61 PFLOPs
GAT [26]	85.12 ± 0.04 65.02 ± 38.91	55.03 ± 0.05 9.48 ± 0.79	54.23 ± 0.20 5.98 ± 2.88	54.05 ± 0.21 4.44 ± 3.94	49.39 ± 0.22 3.14 ± 4.41	126.75 min 16.33 PFLOPs
LIET (Ours)	80.61 ± 0.44 52.72 ± 3.51	**56.70 ± 0.06** 37.17 ± 2.11	**56.14 ± 0.05** **33.55 ± 1.59**	**56.08 ± 0.07** **33.10 ± 1.44**	50.01 ± 0.09 **25.22 ± 0.41**	101.29 min 10.93 PFLOPs

Table 6. Comparison of clean accuracy, robust accuracy and training cost (time in minutes and computation in PFLOPs) on CIFAR-10 using ResNet-18. Each method is evaluated with perturbation sizes of 8/255 (first row) and 16/255 (second row). Best results are highlighted in **bold**.

Method	Clean (%) ↑	PGD-10 (%) ↑	PGD-20 (%) ↑	PGD-50 (%) ↑	AA (%) ↑	Training Cost
PGD-AT [16]	57.52 ± 0.95 48.38 ± 2.04	29.60 ± 0.23 17.03 ± 0.18	28.99 ± 0.21 12.66 ± 0.09	28.87 ± 0.27 11.56 ± 0.16	25.48 ± 0.11 9.17 ± 0.12	353.65 min 59.88 PFLOPs
PGD-AT-WA [24]	56.48 ± 1.34 45.84 ± 1.58	32.51 ± 0.31 22.44 ± 0.10	32.23 ± 0.26 18.09 ± 0.16	32.22 ± 0.25 17.45 ± 0.27	26.84 ± 0.28 12.66 ± 0.15	358.80 min 59.88 PFLOPs
N-FGSM [3]	54.77 ± 1.68 39.39 ± 2.11	30.70 ± 0.27 19.90 ± 0.30	30.47 ± 0.26 16.52 ± 0.04	30.41 ± 0.29 15.95 ± 0.18	25.31 ± 0.28 11.18 ± 0.21	74.62 min 10.89 PFLOPs
FGSM-RS [28]	37.59 ± 15.43 3.34 ± 0.58	20.94 ± 9.59 1.73 ± 0.35	20.86 ± 9.54 1.62 ± 0.33	20.84 ± 9.55 1.61 ± 0.32	16.70 ± 7.84 1.05 ± 0.21	74.62 min 10.89 PFLOPs
FGSM-PGI [13]	56.02 ± 0.21 57.29 ± 9.61	32.70 ± 0.14 3.85 ± 1.87	32.32 ± 0.10 2.08 ± 1.23	32.31 ± 0.11 1.50 ± 1.33	26.76 ± 0.07 0.57 ± 0.70	99.85 min 10.89 PFLOPs
FGSM-UAP [20]	53.54 ± 0.49 44.39 ± 28.33	32.14 ± 0.05 3.18 ± 0.66	31.83 ± 0.04 1.89 ± 0.10	31.81 ± 0.05 1.35 ± 0.47	26.29 ± 0.03 0.55 ± 0.67	114.03 min 16.33 PFLOPs
NuAT [27]	57.72 ± 2.01 62.30 ± 0.08	25.82 ± 1.28 10.72 ± 0.06	22.99 ± 0.94 5.92 ± 0.03	21.62 ± 0.58 4.07 ± 0.11	13.77 ± 0.58 1.94 ± 0.12	131.49 min 16.33 PFLOPs
Grad-Align [1]	54.87 ± 1.17 22.97 ± 13.38	31.86 ± 0.19 11.36 ± 6.63	31.60 ± 0.19 9.63 ± 5.58	31.54 ± 0.23 9.50 ± 5.47	26.18 ± 0.07 6.43 ± 3.93	229.34 min 16.33 PFLOPs
FGSM-AT [7]	21.36 ± 8.36 1.45 ± 0.45	3.04 ± 0.67 0.96 ± 0.04	2.38 ± 0.44 0.85 ± 0.15	1.88 ± 0.32 0.82 ± 0.17	0.19 ± 0.03 0.51 ± 0.49	74.17 min 10.89 PFLOPs
Free-AT [25]	58.29 ± 2.10 20.93 ± 4.08	30.47 ± 0.35 11.10 ± 2.49	30.01 ± 0.30 9.87 ± 2.21	29.96 ± 0.20 9.79 ± 2.21	24.34 ± 0.42 5.84 ± 1.56	71.97 min 10.89 PFLOPs
COAT [14]	**67.56 ± 1.13** 65.67 ± 0.27	24.55 ± 0.16 10.85 ± 1.01	23.23 ± 0.34 6.05 ± 0.73	22.70 ± 0.33 4.45 ± 0.61	18.93 ± 0.07 2.75 ± 0.38	88.85 min 13.61 PFLOPs
GAT [26]	65.24 ± 0.26 **72.42 ± 0.58**	27.61 ± 0.14 3.19 ± 0.04	26.69 ± 0.19 1.42 ± 0.02	26.54 ± 0.16 0.57 ± 0.06	21.97 ± 0.15 0.06 ± 0.03	126.85 min 16.33 PFLOPs
LIET (Ours)	51.52 ± 0.38 35.00 ± 2.85	**32.92 ± 0.12** **20.34 ± 0.86**	**32.75 ± 0.15** **17.60 ± 0.45**	**32.74 ± 0.14** **17.22 ± 0.42**	**27.05 ± 0.09** **12.04 ± 0.30**	103.74 min 11.11 PFLOPs

Table 7. Comparison of clean accuracy, robust accuracy and training cost (time in minutes and computation in PFLOPs) on CIFAR-100 using ResNet-18. Each method is evaluated with perturbation sizes of 8/255 (first row) and 16/255 (second row). Best results are highlighted in **bold**.

Method	Clean (%) ↑	PGD-10 (%) ↑	PGD-20 (%) ↑	PGD-50 (%) ↑	AA (%) ↑	Training Cost
PGD-AT [16]	43.60 ± 2.45 41.27 ± 2.45	20.20 ± 1.82 14.85 ± 0.27	19.90 ± 1.41 13.44 ± 0.04	19.86 ± 1.23 13.15 ± 0.00	16.00 ± 1.02 9.54 ± 0.19	2432.63 min 479.01 PFLOPs
PGD-AT-WA [12]	46.23 ± 0.85 41.89 ± 0.20	26.09 ± 0.36 19.91 ± 0.17	26.06 ± 0.34 18.60 ± 0.03	26.06 ± 0.34 18.43 ± 0.14	19.62 ± 0.24 12.31 ± 0.02	2439.34 min 479.01 PFLOPs
N-FGSM [3]	47.73 ± 0.45 37.98 ± 1.31	25.30 ± 0.26 18.11 ± 0.05	25.18 ± 0.24 16.94 ± 0.08	25.10 ± 0.23 16.64 ± 0.14	18.76 ± 0.24 11.11 ± 0.18	498.95 min 87.09 PFLOPs
FGSM-RS [28]	43.10 ± 4.09 5.64 ± 0.03	22.91 ± 1.35 2.33 ± 0.01	22.71 ± 1.30 2.26 ± 0.00	22.70 ± 1.27 2.25 ± 0.01	16.28 ± 1.03 1.33 ± 0.01	493.77 min 87.09 PFLOPs
FGSM-PGI [13]	48.59 ± 0.19 24.81 ± 1.29	**26.68 ± 0.25** 5.24 ± 0.06	26.46 ± 0.15 4.29 ± 0.04	26.39 ± 0.17 4.02 ± 0.14	19.52 ± 0.07 1.33 ± 0.27	652.09 min 87.09 PFLOPs
FGSM-UAP [20]	45.69 ± 0.99 16.89 ± 9.00	26.12 ± 0.05 2.55 ± 0.10	25.91 ± 0.02 2.05 ± 0.27	25.84 ± 0.04 1.89 ± 0.40	19.45 ± 0.14 0.71 ± 0.59	717.3 min 130.64 PFLOPs
NuAT [27]	45.55 ± 0.99 47.98 ± 4.95	26.51 ± 0.17 13.83 ± 2.69	26.38 ± 0.17 11.25 ± 4.00	26.37 ± 0.14 10.59 ± 4.58	19.55 ± 0.12 5.47 ± 3.37	865.64 min 130.64 PFLOPs
Grad-Align [1]	46.16 ± 2.03 36.83 ± 2.61	24.61 ± 0.51 15.57 ± 0.76	24.30 ± 0.40 14.43 ± 0.79	24.21 ± 0.41 14.22 ± 0.72	17.65 ± 0.28 8.71 ± 0.59	1514.23 min 130.64 PFLOPs
FGSM-AT [7]	34.68 ± 15.85 5.78 ± 1.95	17.12 ± 8.69 1.75 ± 0.08	16.92 ± 8.60 1.62 ± 0.16	16.84 ± 8.55 1.58 ± 0.21	16.60 ± 0.36 0.84 ± 0.29	492.96 min 87.09 PFLOPs
Free-AT [25]	48.19 ± 1.78 32.57 ± 3.72	23.85 ± 0.24 14.89 ± 1.00	23.63 ± 0.25 14.07 ± 0.75	23.58 ± 0.26 13.99 ± 0.73	16.34 ± 0.32 8.42 ± 0.58	474.87 min 87.09 PFLOPs
COAT [14]	**59.30 ± 0.38** **58.52 ± 1.41**	18.45 ± 0.08 10.65 ± 0.96	17.57 ± 0.16 8.15 ± 0.84	17.25 ± 0.22 7.38 ± 0.86	11.56 ± 0.02 3.71 ± 0.60	578.07 min 108.87 PFLOPs
GAT [26]	57.68 ± 0.25 33.84 ± 0.91	17.97 ± 0.23 1.02 ± 0.07	17.26 ± 0.24 0.69 ± 0.04	17.01 ± 0.28 0.60 ± 0.02	11.53 ± 0.15 0.20 ± 0.01	851.30 min 130.64 PFLOPs
LIET (Ours)	44.73 ± 0.33 35.18 ± 0.76	**26.68 ± 0.11** **18.43 ± 0.13**	**26.54 ± 0.12** **17.47 ± 0.10**	**26.54 ± 0.11** **17.31 ± 0.19**	**19.79 ± 0.04** **11.29 ± 0.15**	682.47 min 90.56 PFLOPs

Table 8. Comparison of clean accuracy, robust accuracy and training cost (time in minutes and computation in PFLOPs) on Tiny-ImageNet using PreResNet-18. Each method is evaluated with perturbation sizes of 8/255 (first row) and 12/255 (second row). Best results are highlighted in **bold**.

Method	Clean (%) ↑	PGD-10 (%) ↑	PGD-20 (%) ↑	PGD-50 (%) ↑	AA (%) ↑
N-FGSM [3]	80.60 68.10	56.06 39.17	55.76 31.62	55.67 29.86	50.31 21.66
FGSM-RS [28]	89.23 9.99	15.94 9.82	8.94 9.67	5.50 9.56	0.00 7.51
FGSM-PGI [13]	84.25 89.75	**60.03** 22.92	59.36 18.08	59.23 14.02	52.99 0.30
NuAT [27]	82.68 93.17	57.55 19.34	56.99 10.61	56.94 5.71	52.50 0.12
Grad-Align [1]	82.55 10.00	56.55 9.97	55.99 9.73	55.91 9.12	50.57 2.95
FGSM-AT [7]	**89.99** 11.46	27.07 9.73	18.51 9.51	11.01 9.34	0.01 7.20
Free-AT [25]	76.98 35.74	51.43 21.77	51.04 20.12	50.93 20.11	46.30 15.48
COAT [14]	86.69 **94.51**	53.47 34.91	53.08 20.97	53.11 8.29	41.90 0.04
GAT [26]	79.08 93.23	45.45 11.16	45.02 4.59	44.93 1.53	40.39 0.03
LIET (Ours)	82.49 68.32	59.99 **42.76**	**59.53** **35.96**	**59.48** **33.98**	**53.15** **23.84**

Table 9. Comparison of clean accuracy and robust accuracy on CIFAR-10 using WideResNet-28-10. Each method is evaluated with perturbation sizes of 8/255 (first row) and 16/255 (second row). Best results are highlighted in **bold**.

Method	Clean (%) ↑	PGD-10 (%) ↑	PGD-20 (%) ↑	PGD-50 (%) ↑	AA (%) ↑
N-FGSM [3]	57.55 43.49	**33.33** **20.50**	32.91 16.54	32.85 15.76	27.17 12.17
FGSM-RS [28]	64.43 2.82	8.31 1.07	5.94 0.91	4.06 0.88	0.00 0.40
FGSM-PGI [13]	**70.84** 57.42	19.50 9.99	15.89 6.31	13.39 5.29	1.54 3.50
NuAT [27]	66.23 67.24	22.87 12.45	19.50 6.97	17.98 4.41	12.84 1.68
Grad-Align [1]	26.80 1.80	12.25 1.34	12.19 1.13	12.20 1.01	10.00 0.64
FGSM-AT [7]	65.09 1.00	4.23 0.97	1.95 0.60	1.11 0.48	0.00 0.00
Free-AT [25]	49.72 20.94	27.57 10.64	27.49 9.21	27.48 9.28	22.20 5.67
COAT [14]	70.81 **77.49**	25.97 13.93	23.96 7.98	23.02 3.21	20.16 0.00
GAT [26]	70.66 74.93	27.62 4.30	26.67 1.84	26.62 0.93	23.09 0.04
LIET (Ours)	52.72 35.36	33.30 20.17	**33.16** **17.47**	**33.04** **17.19**	**27.33** **12.72**

Table 10. Comparison of clean accuracy and robust accuracy on CIFAR-100 using WideResNet-28-10. Each method is evaluated with perturbation sizes of 8/255 (first row) and 16/255 (second row). Best results are highlighted in **bold**.