Robust Overfitting Does Matter: Test-Time Adversarial Purification With FGSM

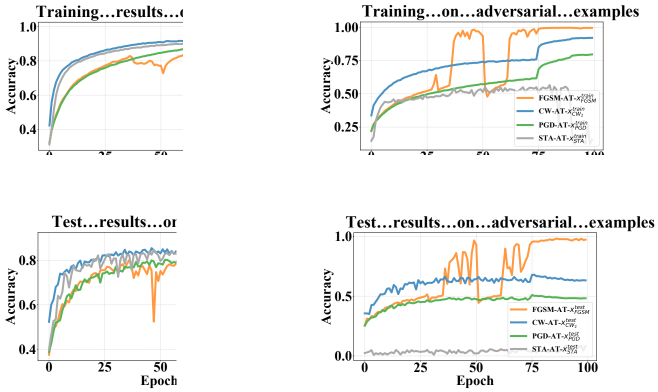
Linyu Tang, Lei Zhang*
School of Microelectronics and Communication Engineering, Chongqing University, China
linyutang@cqu.edu.cn, leizhang@cqu.edu.cn

Abstract

Numerous studies have demonstrated the susce, of deep neural networks (DNNs) to subtle adversar turbations, prompting the development of many ac adversarial defense methods aimed at mitigating ac ial attacks. Current defense strategies usually trai for a specific adversarial attack method and can good robustness in defense against this type of ac ial attack. Nevertheless, when subjected to eval involving unfamiliar attack modalities, empirical e reveals a pronounced deterioration in the robust DNNs. Meanwhile, there is a trade-off between the classification accuracy of clean examples and adversarial examples. Most defense methods often sacrifice the accuracy of clean examples in order to improve the adversarial robustness of DNNs. To alleviate these problems and enhance the overall robust generalization of DNNs, we propose the ***T****est-Time* ***P****ixel-Level* ***A****dversarial* ***P****urification (TPAP) method. This approach is based on the robust overfitting characteristic of DNNs to the fast gradient sign method (FGSM) on training and test datasets. It utilizes FGSM for adversarial purification, to process images for purifying unknown adversarial perturbations from pixels at testing time in a "counter changes with changelessness" manner, thereby enhancing the defense capability of DNNs against various unknown adversarial attacks. Extensive experimental results show that our method can effectively improve both overall robust generalization of DNNs, notably over previous methods. Code is available https://github.com/tly18/TPAP.*

1. Introduction

Despite the substantial achievements of computer vision tasks such as facial recognition, autonomous driving, and medical image processing, the emergence of adversarial attacks [38] seriously threatens the deployment of computer vision models. Adversarial attacks aim to inject human-imperceptible and malicious noise that are carefully crafted by the adversary into origin clean examples [38], causing the loss of discriminative capacity in DNNs.



(c) (d)

Figure 1. (a) and (b) represents the training set results of clean and adversarial examples through adversarial training with FGSM, CW_2, PGD and STA attacks, respectively, on the ResNet18 using the training set of CIFAR-10, and (c) and (d) show the results of cifar10 test set. The horizontal axis represents epochs, while the vertical axis represents classification accuracy.

In response to adversarial attacks, the concept of adversarial defense was introduced to improve the robustness of DNNs. However, these defense efforts often train DNNs only under the project gradient descent (PGD) [25] attack method, without considering whether the networks can defend against other types of attack strategies. As shown in Tab. 1, our experiment shows a pronounced weakness in defending against attacks such as faster Wasserstein attack (FWA) [48] and spatially transform attack (STA) [49]. This is easy to understand, because PGD attack is an algorithm reliant on gradient-based technique for crafting adversarial examples, whereas FWA is based on geometrically measured Wasserstein distance in pixel space, and STA introduces method for minimizing spatial deformations through pixel manipulations to generate adversarial examples. Therefore, PGD adversarial training networks cannot learn the defense knowledge of other attack strategies, leading to a serious lack of robust generalization of the networks. In this way, it seems that DNNs are indeed powerless to unknown adversarial attacks?

Some intriguing experimental results from prior researches that provide crucial inspiration for the defensive method we proposed in the following sections. Given that PGD adversarial training (PGD-AT) [25] is notably time-consuming, Wong *et al*. [46] introduced a method involving random initialization of fast gradient sign method [9] based adversarial training (FGSM-AT). They observe a phenomenon known as "catastrophic overfitting" (referred to as FGSM robust overfitting in this paper). Specifically, FGSM-AT networks exhibit classification error rate approaching 100% when tested on adversarial examples crafted by PGD attack on CIFAR-10 dataset. Andriushchenko *et al*. [2] conducted the same experiments, further unveiling a noteworthy phenomenon. They observe that FGSM-AT networks achieved an incredibly high rate of correct classification, reaching up to 80%, when tested on adversarial test examples generated by FGSM attack. As shown in Fig. 1, we verify the training process of FGSM-AT, CW-AT, PGD-AT, STA-AT that only DNNs trained by FGSM attack reach the robust overfitting state. Figs. 1a and 1c show the classification results of the four ATs for clean examples on the training and test sets, respectively, and Figs. 1b and 1d show the classification results of the four ATs for adversarial examples generated by their respective adversarial attack methods on the training and test sets, respectively, and only the FGSM-AT has a close to 100% accuracy on CIFAR-10 for the adversarial examples generated by its own adversarial attack method. Since FGSM-AT networks exhibit perfect (overfitting) robustness when dealing with FGSM adversarial examples, this discovery inspires us to explore the possibility that whether the networks can correctly classify unknown adversarial examples after performing FGSM adversarial purification on the testing phase. If successful, this could contribute to enhancing the robust generalization of DNNs.

Based on the preceding discussion, we conduct practical experiments and propose the **T**est-Time **P**ixel-Level **A**dversarial **P**urification (TPAP) method to enhance the robust generalization of DNNs against unknown adversarial attacks. Specifically, during training phase, we harness the DNN robust overfitting characteristic of FGSM adversarial training to create a network highly adept at classifying clean examples and defending against FGSM attack. In the testing phase, images are first fed into the DNN to obtain the pre-predicted labels and their cross-entropy loss which help the input images adapt to the robust overfitting network. These prior knowledge are used to perform FGSM adversarial purification on the image pixels to mitigate their adversarial perturbations, and then classified by the DNN.

Our main contributions are summarized as follows:

- We redefine FGSM robust overfitting deep neural networks (FGSM-RO-DNNs), explore for the first time the effect of hyperparameters on training FGSM-RO-DNNs, and validate the effectiveness of our method on multiple datasets and various DNNs.
- Although the adversarial examples are misclassified, they still contain the image semantic information representing their own labels. We propose the TPAP method, which utilizes pre-classification prior knowledge to guide untargeted purification of specific adversarial noise within images, ultimately obtaining correctly classified purified examples.
- Our method does not require the extensive use of additional data for training, significantly reducing training time of DNNs and imposing minimal time overhead during testing phase. Empirical experiments show our method presents superior effectiveness against both pixel-constrained and spatially-constrained unseen types of attacks and adaptive attacks, while improving the accuracy of clean examples.

2. Related Work

Adversarial attack. The adversarial noise generated by the adversarial attack method is limited by a small normball $\left\|x_a - x_c\right\|_p \leq \epsilon$, which means adversarial examples are similar to their clean examples in perception. Adversarial noise can be crafted by attacking in one or more steps along the direction of the adversarial gradient, such as fast gradient sign method (FGSM) [9], basic iterative attack (BIA) [18], momentum iterative attack (MIM) [5], the strongest first-order information based projected gradient descent (PGD) [25], and the autoattack (AA) [4] method. Furthermore, optimization-based attacks, such as Carlini and Wagner (CW) [3], decoupling direction and norm (DDN) [30], minimize the adversarial noise as part of the optimization objectives. The aforementioned attacks directly modify pixel values across the entire image without considering the semantics of the objective, such as shape, outline and posture. These are referred to as pixel-constrained attacks. Furthermore, there are spatial-constrained attacks such as faster Wasserstein attack (FWA) [48], spatial transform attack (STA) [49] and robust physical perturbations (RP2) [7], which focus on mimicking non-suspicious intentional destructive behavior via geometric structures, spatial transformations or physical modifications.

Adversarial Defense. Adversarial Training (AT), originally proposed by Ian Goodfellow *et al*. [9], is one of the most classic and effective methods for adversarial defense. Adversarial training refers to the introduction of adversarial examples into training data during network training process to effectively perform data augmentation, so that network learns attack patterns from adversarial examples during training to enhance the robustness. Research finds that FGSM adversarial training does not always enhance the adversarial robustness of DNNs [26], as the method of generating adversarial examples through a single linear construction does not evidently produce the optimal adversarial examples. [2, 12, 14, 15, 21, 22, 41, 46] focused on improving FGSM adversarial training to prevent catastrophic overfitting. Madry *et al*. [25] proposed using a stronger PGD attack for adversarial training. They formulated adversarial training as a min-max optimization problem and demonstrated PGD adversarial training can obtain a robust network. [25, 39, 56] showed that there is a negative correlation between the clean accuracy and adversarial robustness of DNNs. Therefore, Zhang *et al*. proposed TRADES [56] to decompose the prediction error for adversarial examples (robust error) as the sum of the natural (classification) error and boundary error. They design a new training objective function to balance adversarial robustness and natural accuracy. Wang *et al*. [43] found that misclassified clean examples have a significant impact on the final robustness of DNNs. They proposed the MART algorithm, which explicitly distinguishes between misclassified and correctly classified examples during training, thus constraining misclassified clean examples using a weighting coefficient on loss function to significantly improve the adversarial robustness of the network. Wei *et al*. [45] studied the preferences of different categories for adversarial perturbations and introduced a category-calibrated fair adversarial training framework that automatically tailors specific training configurations for each category. To alleviate these conflicted dynamics of the decision boundary, Xu *et al*. [51] proposed Dynamics-Aware Robust Training (DyART), which encourages the decision boundary to engage in movement that prioritizes increasing smaller margins. Wang *et al*. [44] proposed to exploit better diffusion networks to generate much extra high-quality data for adversarial training, which can improve the robustness accuracy of DNNs. [36, 37, 47, 57] were dedicated to improving training methods and training loss functions to enhance the robustness of DNNs.

Adversarial Purification. The goals of both adversarial purification and adversarial training are to enhance the resilience of DNNs against adversarial attacks. Adversarial training primarily focuses on improving robustness through network training, while adversarial purification places emphasis on purifying input data before feeding it into the classification network during testing to mitigate the impact of adversarial perturbation. [1, 24, 52, 54, 58, 59] added additional network for image purification, such as VAE [13, 16], GAN [8, 32], DUNET [23], and then jointly trained with the classification network to make the classification results and image pixel values of the adversarial examples consistent with the clean examples. However, some of these methods would reduce the accuracy of clean examples and some are computationally intensive. Testing phase adversarial purification is the process of converting adversarial examples encountered by the network in actual inference into clean data representations. Shi *et al*. [34] introduced SOAP, which employs double consistency losses during the training phase and self-supervised loss during testing phase to purify adversarial examples. SOAP provides flexibility against adversarial attacks compared to networks that use a fixed architecture during testing. [28, 42] used diffusion network for denoising adversarial examples, where Gaussian noises are gradually added to submerge the adversarial perturbations during the forward diffusion process and both of these noises can be simultaneously removed following a reverse reconstruction process. However, these methods need long training time and large memory.

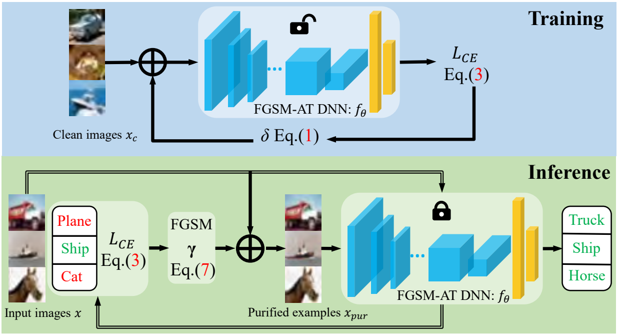


Figure 2. Overview of the training phase and testing phase inference phase. Arrows indicate data flow, and double straight arrows indicate testing phase pre-classification.

3. Method

This paper investigates how to achieve robust DNNs by utilizing the under-studied FGSM robust overfitting prior. We present the **T**est-Time **P**ixel-Level **A**dversarial **P**urification (TPAP) method, a novel defense strategy that uses a FGSM robust overfitting network and adversarial purification processing at testing phase for robust defense against unknown adversarial attacks, as illustrated in Fig. 2.

3.1. Preliminary

This paper primarily focuses on the task of image classification under adversarial attacks. We use x_c to represent clean examples, x_a for adversarial examples, x for input images including clean examples x_c and adversarial examples x_a, x_{pur} for purified examples, y for the true labels corresponding to the images, and $y_{pred}(\cdot)$ for the prediction labels. δ represents the adversarial perturbation added to the image pixels when generating adversarial examples and γ represents the adversarial purification applied to the image pixels during the test-time purification phase. ϵ and ξ respectively denote the maximum magnitude of pixel value changes in the generated adversarial examples and the purified images at testing phase. α and β denote the step size. C represents the number of categories in the dataset. we use θ to denote the weight parameters of a DNN f.

3.2. Robust Overfitting Prior of FGSM-AT

In this paper, we redefine the network robust overfi as follows: on the training set, the classification accu of adversarial examples generated by a specific and tra attack method is higher than 90%, and the classificatior curacy of other kinds of adversarial examples is less 10%. Most importantly, on the test set, the classificatior curacy of clean examples and adversarial examples cra through known attack on DNNs is high. This FGSM ro overfitting is shown after 80 epochs in Figs. 4a and 4b.

During the training phase of DNN, the network undergoes adversarial training using FGSM adversarial examples until DNN reaches the state of FGSM robust overfitting. Formally, FGSM adversarial examples with ℓ_∞-norm for $\alpha = \epsilon$ are computed by,

$$\delta = \alpha * sign(\frac{\partial L_{CE}(f_\theta(x_c), y)}{\partial x_c}) \tag{1}$$

$$x_a = x_c + \delta \tag{2}$$

where L_{CE} represents cross-entropy loss defined as,

$$\begin{aligned} L_{CE}(f_\theta(x), y) &= -\sum_{s=0}^{C-1} P(x, s) * log \frac{e^{f_\theta(x,s)}}{\sum_{k=0}^{C-1} e^{f_\theta(x,k)}} \\ &= -log \frac{e^{f_\theta(x,y)}}{\sum_{k=0}^{C-1} e^{f_\theta(x,k)}} \end{aligned} \tag{3}$$

where $P(x, s) \in \mathbb{R}^1$ denotes the probability categorized as s in the ground-truth label $P(x) \in \mathbb{R}^C$. $f_\theta(x) \in \mathbb{R}^C$ is the output of DNN, $f_\theta(x, k) \in \mathbb{R}^1$ represents the value of the output neuron k. The partial derivatives of the cross-entropy loss function with respect to the network weights are calculated and updated as,

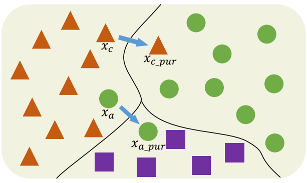
$$\theta = \theta - \frac{\partial L_{CE}(f_\theta(x_a), y)}{\partial \theta} \tag{4}$$

3.3. Test-time Pixel-Level Purification

During the test phase, the parameters of the FGSM robust overfitting network (FGSM-RO-DNN) obtained during the training phase are frozen. We expect to start from the pixels of images, purifying adversarial examples to ensure accurate classification without affecting the correct classification results of the purified clean examples. Our idea is expressed by the following equation,

$$\gamma = \underset{||\gamma||_p \leq \xi}{\text{argmin}}\, L_{\text{CE}}(f_\theta(x + \gamma), \text{Label}(f_\theta(x))) \tag{5}$$

In the specific implementation, the images containing clean and adversarial examples are directly fed into the FGSM-RO-DNN to obtain pre-predicted labels and the partial derivatives of their cross-entropy loss function for each pixel on the input images. We choose the FGSM robust overfitting network based on 2 considerations: 1) it is highly robust to pairs of clean examples and FGSM adversarial examples and 2) is not resistant to other unknown types of attacks. In other words, FGSM-RO-DNN is more suitable for the purification process in the testing phase than other networks because even though images are easily attacked by other types of attack methods, they can be corrected in FGSM adversarial purification. As shown in Fig. 3a, after purification, it is able to classify the x_c and x_a correctly, however Fig. 3b is not. Our experiments in Tab. 7 validate the experimental results using other attack methods instead of FGSM. As shown in Fig. 3a, for clean examples, if they are correctly classified before purification processing, the classification after processing is amount to the classification of adversarial examples with a known attack type. Our method ensures that FGSM-RO-DNN has high classification accuracy on clean data and FGSM adversarial examples, and hence it can correctly classify clean examples with high confidence. If clean examples are misclassified before purification, the likelihood of misclassification after purification is high, but and negligible due to its low possibility. For adversarial examples, if they are misclassified before purification processing, maximum confidence classification results are misclassified labels. Using these misclassified labels for FGSM untargeted adversarial purification effectively bring adversarial examples back to the decision boundary, eliminating adversarial perturbation. Then, purified examples are fed into the classification network, and FGSM-RO-DNN can correctly classify them due to its robustness to FGSM attcak.



(b)

Figure 3. (a) and (b) represent the processing of FGSM robust overfitting and other adversarial training methods for DNNs in the test purification phase, respectively. The black curves indicate the categorization boundaries, and the triangles, circles, and squares indicate the 3 different categories, respectively.

In adversarial purification process, it is necessary to satisfy pixel change constraints and ensure that the image semantic information is not disrupted. Formally, the output of the DNN w.r.t. input x is represented as $f_\theta(x) = a_0, a_2, \ldots, a_{c-1}$. The predicted label is the position of the largest neuron denoted as,

$$\begin{aligned} y_{pred}(x) &= \text{Position}(\max(a_0, a_2, \ldots, a_{c-1})) \\ &= \text{Label}(f_\theta(x)) \end{aligned} \tag{6}$$

The FGSM purification implementation with ℓ_∞-norm for $\beta = \xi$ is computed by,

$$\gamma = \beta * sign(\frac{\partial L_{CE}(f_\theta(x), y_{pred}(x))}{\partial x}) \tag{7}$$

$$x_{pur} = x + \gamma \tag{8}$$

Finally, the purified example is fed into the network again to obtain the final classification prediction results.

$$y_{pred}(x_{pur}) = \text{Label}(f_\theta(x_{pur})) \tag{9}$$

The algorithm of TPAP is summarized in Algorithm 1.

Algorithm 1 : The Algorithm of TPAP.

Training phase: FGSM-RO-DNN training

```
1: Input: Training set $x_c$, network $f_\theta(\cdot)$ parameterized by $\theta$, batch size $B$, perturbation radius $\epsilon$, total number of iterations $epochs$;
2: Output: Robust overfitting network trained with TPAP.
3: for $i = 1$ to $epochs$ do
4:     for $j = 1$ to $B$ (in parallel) do
5:         Obtain FGSM adversarial perturbations using Eq. (1).
6:         Obtain FGSM adversarial examples $x_a$ using Eq. (2).
7:         Update network weights with the optimizer in Eq. (4).
8:     end for
9: end for
10: return the network weight parameters $\theta$.
```

Testing phase: Test-time purification processing

```
1: Input: Test set $x$ including clean examples $x_c$ and adversarial examples $x_a$, pre-trained robust overfitting network $f_\theta(\cdot)$, purification radius $\xi$;
2: Output: Prediction labels $y_{pred}$.
3: for $i = 1$ to $B$ (in parallel) do
4:     Compute the pre-prediction of input images using Eq. (6).
5:     Perform adversarial purification using Eq. (7).
6:     Obtain purified examples $x_{pur}$ using Eq. (8).
7:     Compute the output of $x_{pur}$ and obtain the maximum classification results as their final prediction labels in Eq. (9).
8: end for
9: return $y_{pred}(x_{pur})$
```

4. Experiments

We conduct comprehensive experiments on CIFAR-10, CIFAR-100 [17], SVHN [27], Tiny-ImageNet [19] datasets with ResNet-18 [10], VGG-16 [35] and WideResNet-34 [55] to evaluate the effectiveness of our proposed method.

4.1. Implementation Details

4.1.1 Datasets

CIFAR-10 and **CIFAR-100** [17] contain 10 and 100 categories, respectively. **SVHN** [27] is from house number plates in Google Street View images, containing a sequence of Arabic numerals '0-9'. **Tiny-ImageNet** [19] is a subset of the ImageNet dataset. It consists of 200 classes.

4.1.2 Training Phase

We set the batch size (bs) to 128 for CIFAR-10, CIFAR-100 and SVHN, and 64 for Tiny-ImageNet under ResNet-18. For VGG-16, we set the batch size as 128 for CIFAR-10, CIFAR-100, SVHN and Tiny-Imagenet. For WideResNet-34, we set the batch size to 64 for CIFAR-10, CIFAR-100 and SVHN, and 32 for Tiny-ImageNet. We adopt stochastic gradient descent (SGD) [29, 31] optimizer with momentum factor of 0.9, an initial learning rate of 0.1 or 0.01 divided by 10 at the 75-th, 90-th and 140-th epochs and a weight decay factor of 1×10^{-3}. The total number of epochs is set to 150. For CIFAR-10 and CIFAR-100, we augment the training data by random cropping and random horizontal flipping after filling 4 pixels. For Tiny-Imagenet, we only use random horizontal flipping. We use FGSM-AT and set the maximum ℓ_∞-norm of adversarial perturbation to ϵ = 8/255 or 12/255. Adversarial purification radius is set to ξ = 8/255. All experiments are implemented on the GeForce 1080 and 2080 TI GPU.

4.1.3 Evaluation Phase

We comprehensively consider various attack angles and potential vulnerabilities, and employ many different types of methods to generate adversarial examples to more comprehensively test and evaluate the overall robustness of deep learning models. We choose FGSM [9], PGD [25] (PGD-20 and PGD-100), CW [3], DDN [30], STA [49], FWA [48], AutoAttack (AA) [4] and TI-DIM [6, 50] attack methods in the white-box non-targeted attack setting. The code for attack methods comes from advertorch, torchattacks and authors. The adversarial perturbation strength of all attack methods under ℓ_∞-norm except CW and DDN under ℓ_2-norm is set to 8/255. We compare TPAP with the current mainstream adversarial training and image pre-processing methods [11, 44, 53], including PGD-AT [25], TRADES [56], MART [43] (Training the network with PGD-10) and SOAP [34]. The combination of our proposed method and some of these methods are respectively referred to as TPAP+TRADES and TPAP+MART.

4.2. Training Robust Overfitting Network

Existing work has never explored to train a FGSM-RO-DNN to achieve robustness for both clean and adversarial examples. We find that three hyperparameters have a significant impact on FGSM robust overfitting, namely the learning rate of the network, the size of the training batch and the strength of the adversarial perturbation of the FGSM attack. Figs. 4a and 4b show FGSM adversarial training

Table 1. Classification accuracy rates (percentage) against white attacks on ResNet-18, VGG-16, WideResNet-34 for CIFAR-10, CIFAR-100, SVHN and Tiny-ImageNet datasets. The best results are highlighted in **bold** and the second best in underline.

ResNet-18	CIFAR-10										CIFAR-100									
Method	Clean	FGSM	PGD-20	PGD-100	CW$_2$	DDN$_2$	AA	STA	FWA	TI-DIM	Clean	FGSM	PGD-20	PGD-100	CW$_2$	DDN$_2$	AA	STA	FWA	TI-DIM
PGD-AT [25]	84.54	55.11	48.91	47.7	59.15	19.15	43.3	0.35	3.19	49.23	57.77	28.68	25.52	24.99	30.42	11.03	21.21	0.03	2.55	25.79
TRADES [56]	83.22	58.51	54.97	54.07	71.62	24.24	48.97	1.09	5.38	55.14	53.93	30.22	28.06	27.77	35.35	14.16	23.01	0	5.25	28.25
MART [43]	82.14	59.57	55.39	54.8	74.3	25.83	47.84	2.06	6.43	56.24	55.52	30.83	28.38	28.16	35.57	13.78	23.01	0.02	3.44	**28.51**
SOAP [34]	84.07	51.02	51.42	-	73.95	-	-	-	-	-	52.91	22.93	27.55	-	50.26	-	-	-	-	-
TPAP(Ours)	**86.25**	**61.41**	**79.06**	**80.5**	61.37	64.5	**76.34**	31.4	**52.83**	**75.21**	57.43	**35.64**	44.69	**42.23**	48.7	50.84	**47.48**	43.8	**32.23**	27.84
TPAP+TRADES	84.07	44.16	73.02	66.12	90.87	87.29	74.94	80.38	**51.34**	31.52	57.67	27.71	37.82	32.93	**70.49**	**65.62**	35.23	**66.92**	27.06	15.41
TPAP+MART	84.06	43.6	73.69	69.78	**92.38**	**90.05**	72.11	**85.7**	46.69	23.25	**61.03**	32.6	**44.9**	39.49	68.61	61.38	46.19	66.39	28.45	15.66

ResNet-18	SVHN										Tiny-ImageNet									
Method	Clean	FGSM	PGD-20	PGD-100	CW$_2$	DDN$_2$	AA	STA	FWA	TI-DIM	Clean	FGSM	PGD-20	PGD-100	CW$_2$	DDN$_2$	AA	STA	FWA	TI-DIM
PGD-AT [25]	91.66	**87.93**	63.86	44.57	72.65	8.84	31.35	6.51	10.1	**68.19**	**49.06**	24.26	22.09	21.44	28	30.41	16.82	0.21	0.99	22.06
TRADES [56]	91.32	73.38	59.01	56.31	72.96	5.19	47.03	1.57	0.31	59.18	46.59	22.9	21.46	21.03	28.87	28.8	15.99	0	1.8	21.54
MART [43]	91.81	75.31	56.55	51.28	71.45	6.96	42.08	0.9	0.86	56.68	46.21	25.73	24.16	23.59	30.58	30.24	17.85	0.34	1.75	24.23
TPAP(Ours)	89.62	67.56	**83.62**	**85.25**	51.39	62.07	**88.76**	55.12	**60.56**	40.99	48.72	**46.6**	37.88	**36.87**	31.48	45.28	**39.8**	12.43	**38.31**	**32.93**
TPAP+TRADES	91.36	41.22	80.31	72.77	92.14	**88.96**	66.57	88.31	28.67	59.64	46.22	8.96	17.93	14.07	**48.54**	**47.95**	20.5	**42.48**	5.4	5.5
TPAP+MART	**93.74**	26.92	81.62	66.31	**93.28**	88.36	63.26	**90.06**	28.72	26.65	48.88	18.46	**38.79**	36.53	41.23	43.42	31.72	26.81	36.49	26.93

VGG-16	CIFAR-10										CIFAR-100									
Method	Clean	FGSM	PGD-20	PGD-100	CW$_2$	DDN$_2$	AA	STA	FWA	TI-DIM	Clean	FGSM	PGD-20	PGD-100	CW$_2$	DDN$_2$	AA	STA	FWA	TI-DIM
PGD-AT [25]	81.11	51.91	45.18	43.78	56.12	24.75	39.26	0.26	4.55	45.51	50.68	24.4	20.87	20.31	25.27	11.9	17.84	0	3.29	21.1
TRADES [56]	78.75	52.84	49.23	48.21	66.68	29.03	43.01	1.02	6.79	49.41	48.41	**26.77**	24.52	24.19	31.25	14.51	20.23	0	6.52	24.72
MART [43]	77.79	**54.03**	50.42	49.57	67.47	29.48	43.4	1.75	6.04	50.72	49.05	25.46	23.02	22.53	28.4	12.53	19.3	0.07	4.78	23.12
TPAP(Ours)	77.02	53.4	63.99	57.68	40.57	31.2	38.71	**89.5**	29.29	54.65	48.11	5.93	55.25	54.9	26.8	16.9	29.65	5.73	7.16	**44.76**
TPAP+TRADES	**89.13**	41.55	78.13	74.01	**88.98**	**87.38**	74.64	61.17	**31.48**	48.09	59.85	25.36	49.04	42.99	**61.15**	**55.66**	47.71	**53.18**	26.54	22.9
TPAP+MART	88.1	23.44	**89.98**	**88.42**	83.51	78.37	**85.08**	51.52	31.13	**56.77**	**62.04**	19.41	**61.79**	**58.28**	59	**51.45**	**50.23**	35.41	**29.67**	31.82

VGG-16	SVHN										Tiny-ImageNet									
Method	Clean	FGSM	PGD-20	PGD-100	CW$_2$	DDN$_2$	AA	STA	FWA	TI-DIM	Clean	FGSM	PGD-20	PGD-100	CW$_2$	DDN$_2$	AA	STA	FWA	TI-DIM
PGD-AT [25]	92.11	65.05	53.64	52.18	64.15	6.6	43.85	0.51	0.79	55.12	37.74	14.51	11.74	10.83	13.97	19.43	9.42	0	1.46	11.8
TRADES [56]	90.83	66.27	56.43	55.15	68.83	6.25	45.89	1.96	0.53	57.46	34.8	**16.8**	15.08	14.7	20.26	20.7	11.88	0	3.16	15.25
MART [43]	92.01	**69.02**	56.64	54.78	68.78	12.04	43.21	1.87	3.25	**58.23**	36.56	15.38	13.33	12.76	16.52	20.19	10.63	0	2.28	13.51
TPAP(Ours)	94.09	52.01	90.73	88.52	94.73	94.21	**84.99**	80.45	65.6	47.65	37.93	11.69	35.94	35.41	35.71	39.68	**35.04**	33.69	**19.46**	**23.03**
TPAP+TRADES	93.92	57	86.64	79.05	**95.5**	93.87	73.23	76.99	53.35	44.78	**50.43**	12.55	27.56	19.37	**54.41**	**53.22**	25.01	**43.88**	9.25	6.93
TPAP+MART	**94.24**	52.28	**90.8**	**88.69**	94.51	**94.23**	84.85	**80.53**	**65.94**	48.35	48.36	8.46	**41.56**	**35.69**	41.97	44.3	30.07	16.34	18.15	14.66

WideResNet-34	CIFAR-10										CIFAR-100									
Method	Clean	FGSM	PGD-20	PGD-100	CW$_2$	DDN$_2$	AA	STA	FWA	TI-DIM	Clean	FGSM	PGD-20	PGD-100	CW$_2$	DDN$_2$	AA	STA	FWA	TI-DIM
PGD-AT [25]	**87.41**	59.27	51.59	50.7	57.48	19.28	47.55	0.11	6.77	52.24	61.91	32.75	29.2	28.63	33.67	10.16	25.18	0	2.06	29.53
TRADES [56]	84.01	60.04	56.51	56.13	72.4	21.78	51.82	0.62	4.83	56.86	58	33.26	31.16	30.81	36.78	12.42	26.7	0	4.04	31.17
MART [43]	85.67	**61.98**	55.97	55.15	68.19	21.37	49.68	0.88	7.29	**57.08**	58.92	35.14	32.32	31.92	39.06	12.62	27	0.01	3.26	32.73
TPAP(Ours)	82.73	38.59	**72.69**	**73.63**	58.28	58.78	67.23	26.96	19.97	55.6	**64.34**	**38.49**	37.42	**39.92**	42.87	44.71	38.97	38.94	20.72	**33.37**
TPAP+TRADES	84.13	39.6	60.03	53.07	**91.5**	**86.59**	60.25	**79.52**	25.75	28.59	58.04	31.88	37.17	33.38	**73.17**	**66.71**	38.76	65.67	20.15	20.48
TPAP+MART	85.66	36.27	70.12	64.29	87.86	81.63	**70.55**	78.85	**33.89**	23.47	56.36	25.42	**41.94**	38.77	70.41	62.54	**42.28**	**67.58**	**25.89**	16.9

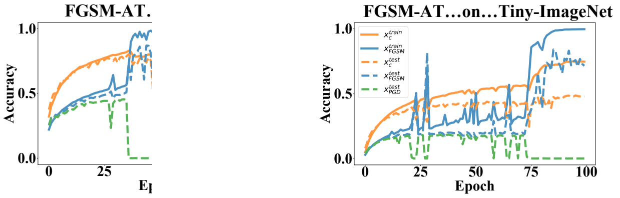
WideResNet-34	SVHN										Tiny-ImageNet									
Method	Clean	FGSM	PGD-20	PGD-100	CW$_2$	DDN$_2$	AA	STA	FWA	TI-DIM	Clean	FGSM	PGD-20	PGD-100	CW$_2$	DDN$_2$	AA	STA	FWA	TI-DIM
PGD-AT [25]	92.88	79.39	71.2	50.5	66.12	6.61	37.21	3.23	0.73	54.77	53.3	29.63	26.84	25.99	33.7	34.58	21.81	0.26	1.43	27.08
TRADES [56]	**94.05**	83.61	69.51	60.7	81.72	5.95	46.4	1.66	0.8	**65.18**	53.78	29.69	27.83	27.1	33.31	34.25	21.98	0.25	1.43	**28.02**
MART [43]	92.19	82.85	70	51.78	66.87	9.04	40.57	6.21	0.91	58.02	52.18	**29.95**	27.8	27.14	34.17	33.73	21.86	0	1.54	27.96
TPAP(Ours)	93.97	**85.79**	84.85	**87.16**	83.28	75.92	**89.12**	77.33	**48.47**	57.86	48.04	23.69	30.46	**29.25**	23.8	35.32	27.76	4.49	**36.32**	17.61
TPAP+TRADES	93.27	45.68	83.26	73.37	**93.83**	**89.19**	64.62	**89.18**	4.55	55.17	**56.1**	17.31	**33.17**	28.98	**67.01**	**65.32**	**40.93**	61.49	7.59	10.94
TPAP+MART	92.73	21.45	**88.07**	78.9	88.47	84.75	62.77	84.61	7.54	51.49	50.96	10.29	22.97	21.79	66.07	62.08	31.27	**63.38**	6.48	5.19

process, reflecting the effect of learning rate on the FGSM-RO-DNN. The initial learning rate is divided by 10 at the 75th and 90th epochs. A high learning rate leads to dramatic oscillations in the classification results of the adversarial examples, and as the learning rate decreases, the classification results of the clean and FGSM adversarial examples gradually increase and stabilize.

Figs. 5a and 5b explore the effects of perturbation strength and batch size on robust overfitting characteristic of DNN, respectively. We conduct an ablation study on CIFAR-10 dataset using ResNet-18 to explore these key hyperparameters. The figures respectively illustrate line plots depicting the variation in classification results during the training phase as the perturbation strength ranges from 8/255 to 16/255 with a step size of 2/255 and the batch size varies across 32, 64, 128, 256 and 512. The ablation results show that a trade-off between ϵ and batch size is required to obtain FGSM-RO-DNN in TPAP.

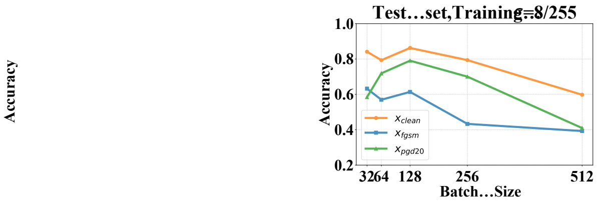
4.3. Experimental Results

The white-box attack results for CIFAR-10, CIFAR-100, SVHN and Tiny-ImageNet are shown in Tab. 1. On these datasets, TPAP and its variants maintain higher accuracy on clean examples compared to AT based methods. In terms of adversarial robustness, PGD adversarial training network cannot resist STA and FWA attacks, but TPAP greatly im-



(a) (b)

Figure 4. (a) and (b) respectively represent the FGSM adversarial training process on the ResNet18 using CIFAR-10 and Tiny-ImageNet datasets. The horizontal axis represents epochs and the vertical axis represents classification accuracy. Solid lines repre-



(a) (b)

Figure 5. (a) and (b) represent ablation study of robust training in TPAP under various perturbation strengths and batch sizes.

proves the accuracy of adversarial examples under these attacks. Also, TPAP outperforms other methods in most adversarial attack scenarios. This is because most existing defense methods ignore the diversity of attacks. Our method can perform pixel-level purification of adversarial examples generated by unknown attacks in a "counter changes with changelessness" manner, and reliably pulls the adversarial examples back within the boundary of correct classification. This benefits from the vulnerability of the FGSM-RO-DNN to non-FGSM adversarial examples and the robustness to FGSM adversarial examples.

However, TPAP is usually less robust to 8/255 FGSM adversarial examples than AT, which is explainable and easy to understand as shown in x_{FGSM} of Fig. 3a. The image purification process makes the correctly classified 8/255 FGSM examples subject to FGSM attack far from the correct label, which is amount to generating 16/255 FGSM adversarial examples. But the network is not trained on 16/255 FGSM adversarial examples, so it cannot classify them with high accuracy. To deal with this problem, we try to use both 8/255 and 16/255 FGSM adversarial examples to train the network, and the results are shown in Tab. 2. Although it enhances the robustness of 8/255 FGSM adversarial examples, it reduces the classification accuracy on clean examples from 86.25% to 82.11%. Further, we use clean examples, 8/255 and 16/255 FGSM adversarial examples to train the network, but may not obtain a FGSM-RO-DNN.

Experiments further validate the robust overfitting of TPAP acting on large-size images, such as Caltech-101, consisting of a total of 9146 images from 101 object classes, as well as an additional background/clutter class. The image size is 300×200. Each object category contains between 40 and 800 images on average. To account for domain gap within Caltech-101, we utilize the ResNet-18 network pre-trained on ImageNet provided by official PyTorch implementation. In Tab. 3, * indicates the use of pre-trained network. The train and test sets are split randomly in 8/2 ratio. Compared with PGD-AT* (ϵ=8/255), TPAP-TRADES* (ϵ=16/255, bs=32) shows superior robustness on both clean examples and PGD-adversarial examples (ϵ=8/255). More comparative experiments with image pre-processing methods on CIFAR-10 are presented in Tab. 4.

Table 2. Classification accuracy against adversarial examples on CIFAR-10. TPAP denotes the robust overfitting ResNet-18 trained with 8/255 FGSM adversarial examples, while TPAP* includes both 8/255 and 16/255 FGSM adversarial examples.

ResNet-18	CIFAR-10									
Method	Clean	FGSM	PGD-20	PGD-100	CW_2	DDN	AA	STA	FWA	TI-DIM
TPAP	86.25	61.41	79.06	80.5	61.37	64.5	76.34	31.4	52.83	75.21
TPAP*	82.11	73.23	80.84	79.69	80.65	79.04	75.72	69.87	66.08	80.85

Table 3. Caltech-101.

Method	Clean	FGSM	PGD-20
PGD-AT*	72.44	65.55	57.74
TPAP*	70.2	55.93	59.16
TPAP-TRADES*	76.11	60.8	70.91
TPAP-MART*	69.27	58.39	60.63

Table 4. Comparative experiment.

Method	Clean	PGD	Architecture
(Yang et al., 2019)(p:0.4→0.6) [53]	84	68.2	ResNet-18
(Hill et al., 2021) [11]	84.12	78.91	WRN-28-10
(Wang et al., 2023) [44]	92.58	68.43	WRN-28-10
TPAP	86.25	79.06	ResNet-18

4.4. Computational overhead and ablation study

We use DeepSpeed from Microsoft to compute FLOPs for TPAP and PGD-AT (trained on PGD-10 adversarial examples) in the same conditions, presented in Tab. 5. Compared with PGD-AT, TPAP reduces computation cost on adversarial examples generation during training, but increases in testing phase due to the adversarial purification operation. This is common because the proposal is a test-time approach. Tab. 6 presents ablation experiment of TPAP and verifies the RO-FGSM-DNN performs well on both clean data and FGSM adversarial examples. After adversarial purification, the accuracy of TPAP for clean data just reduces by 0.08%. This indicates our model is plastic to clean data. Furthermore, we replace FGSM attack with CW_2 and PGD attacks, and as shown in Tab. 7, the accuracy on clean samples is decreased. This further supports our finding that FGSM attack is unique for robust overfitting.

Table 5. Comparison of computational overhead.

ResNet-18	CIFAR-10(ϵ=8/255, Batch size = 128)				
Method	Training time (s)/epoch	Test time (s)/epoch	Training FLOPs (T)/epoch	Test FLOPs (T)/epoch	Params (M)
PGD-AT	273.89	2.02	1831.5	11.1	11.17
TPAP	66.26	8.26	333	44.4	11.17

Table 6. Ablation study of TPAP.

ResNet-18		CIFAR-10(ϵ=8/255,bs=128)		
Purification	RO-FGSM-DNN	Clean	FGSM	PGD-20
×	✓	86.33	94.41	0.18
✓	✓	86.25	61.41	79.06

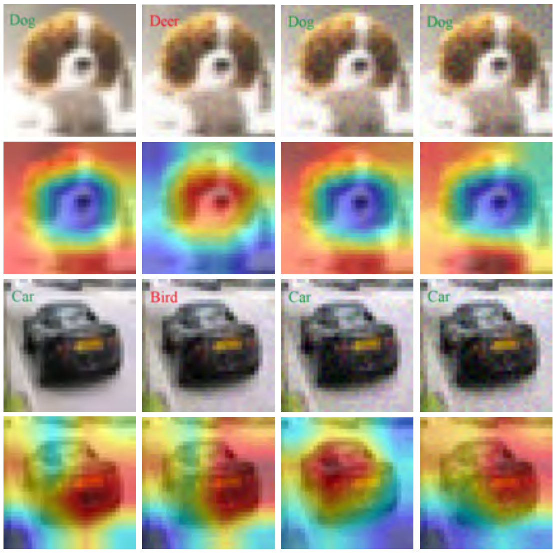
Table 7. Under different attacks.

ResNet-18	CIFAR-10(ϵ=8/255,bs=128)				
Training	Clean	CW2	PGD-20	FGSM	AA
CW2	68.19	84.7	37.4	39.95	56.66
PGD-10	56.17	75.62	42.74	44.5	82
FGSM	86.25	61.37	79.06	61.41	76.34

4.5. Analysis of the Visualization Experiments

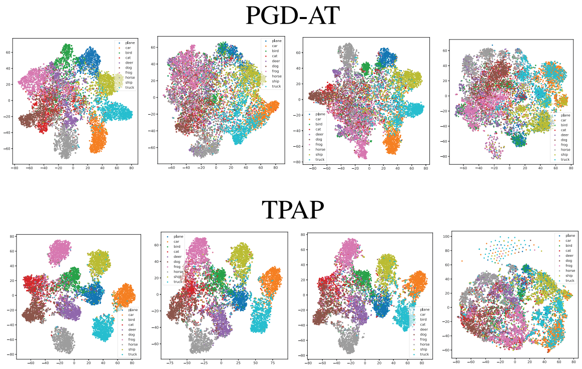
Grad-CAM [33] **Visualization**. Grad-CAM is a visualization technique used to explain the decision-making process of models, highlighting crucial regions of an image feature map. Fig. 6 demonstrates the visualization experiment of TPAP. We see the adversarial examples in (b) have biased attention regions compared to those of correctly classified examples in (a), (c) and (d), although their attention regions can focus on the target object. The results in (a), (c) and (d) show that the FGSM-RO-DNN focuses on the global outline or internal regions of the target category for correctly classified examples. There are two puzzling questions: *Why does the FGSM-RO-DNN focus on the global outline of the target category?* and *why does the FGSM-RO-DNN result in misclassification when focusing on the interior of the target object?* In low-resolution image classification tasks, there is minimal spurious correlation between labels and backgrounds due to the similarity in background colors for different classes. The network finds it challenging to learn correct classification knowledge from non-causal factors such as background. Therefore, we argue that the FGSM-RO-DNN learns one of the most significant distinctions between categories, i.e. the global outline. Additionally, different target categories share similar internal regions, such as cats and dogs having similar fur colors and facial features, or cars and trucks having similar colors. Meanwhile, the FGSM-RO-DNN is highly vulnerable to other types of attacks except FGSM, which also leads to misclassification even focused on interior of target class.

Feature Visualization. Fig. 7 visualizes the feature distribution from the penultimate layer of network. We use t-SNE [40] to project CIFAR-10 features onto a two-dimensional plane, where the top row comes from the baseline (PGD-AT) and the second row represents TPAP. We observe adversarial attacks often distort the discriminative feature distribution of PGD-AT network, whereas TPAP can effectively improve the adversarial distributions with better feature clustering (i.e., inter-class separability and intra-class compactness). Robustness is indicated.



(a) x_c (b) x_{pgd20} (c) x_{c_pur} (d) x_{pgd20_pur}

Figure 6. Attention visualization of TPAP. We show 2 examples of *dog* and *car* respectively. The first and second column respectively indicate the attention map of clean and the PGD-20 adversarial examples, represented as x_c and x_{pgd20}. The third and fourth column respectively represents the attention maps of purified examples x_{c_pur} and x_{pgd20_pur} obtained by our TPAP.



(a) x_c (b) x_{pgd20} (c) x_{AA} (d) x_{STA}

Figure 7. t-SNE feature distribution visualization of PGD-AT and our proposed TPAP on clean and adversarial examples.

5. Conclusion

We propose a novel adversarial defense method and, for the first time, introduce FGSM robust overfitting to instruct test-time robustness. TPAP is easy to train and computationally efficient, and remarkably enhances the robust generalization. We hope our work can inspire future research.

Limitations. TPAP has two limitations. Firstly, TPAP does not perform well in defending against 8/255 FGSM adversarial examples, and the reason is explained in Section 4.3. Secondly, TPAP fails to defend against attacks of too strong perturbations, such as ϵ=0.3 on MNIST [20] dataset. This is due to existing adversarial attacks move clean examples away from correctly classified labels, while our adversarial purification moves adversarial examples away from misclassified labels. When the adversarial perturbation is large, the force of adversarial purification also needs to be enlarged, leading to large pixel changes in image. This severely destroys the semantics of the image and leads to a significant decrease in classification accuracy.

Acknowledgement. This work was supported by National Key R&D Program of China (2021YFB3100800), National Natural Science Fund of China (62271090), Chongqing Natural Science Fund (cstc2021jcyj-jqX0023) and National Youth Talent Project.

References

[1] Naveed Akhtar, Jian Liu, and Ajmal Mian. Defense against universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3389–3398, 2018. 3

[2] Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33:16048–16059, 2020. 2, 3

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017. 2, 5

[4] 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. 2, 5

[5] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 2

[6] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019. 5

[7] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018. 2

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 3

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2, 5

[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. 5

[11] Mitch Hill, Jonathan Mitchell, and Song-Chun Zhu. Stochastic security: Adversarial defense using long-run dynamics of energy-based models. *arXiv preprint arXiv:2005.13525*, 2020. 5, 7

[12] Zhichao Huang, Yanbo Fan, Chen Liu, Weizhong Zhang, Yong Zhang, Mathieu Salzmann, Sabine Süsstrunk, and Jue Wang. Fast adversarial training with adaptive step size. *IEEE Transactions on Image Processing*, 2023. 3

[13] Uiwon Hwang, Jaewoo Park, Hyemi Jang, Sungroh Yoon, and Nam Ik Cho. Puvae: A variational autoencoder to purify adversarial examples. *IEEE Access*, 7:126582–126593, 2019. 3

[14] Xiaojun Jia, Yong Zhang, Xingxing Wei, Baoyuan Wu, Ke Ma, Jue Wang, and Xiaochun Cao Sr. Improving fast adversarial training with prior-guided knowledge. *arXiv preprint arXiv:2304.00202*, 2023. 3

[15] Hoki Kim, Woojin Lee, and Jaewook Lee. Understanding catastrophic overfitting in single-step adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8119–8127, 2021. 3

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[18] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 2

[19] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 5

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 8

[21] Bai Li, Shiqi Wang, Suman Jana, and Lawrence Carin. Towards understanding fast adversarial training. *arXiv preprint arXiv:2006.03089*, 2020. 3

[22] Tao Li, Yingwen Wu, Sizhe Chen, Kun Fang, and Xiaolin Huang. Subspace adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13409–13418, 2022. 3

[23] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1778–1787, 2018. 3

[24] Wei-An Lin, Chun Pong Lau, Alexander Levine, Rama Chellappa, and Soheil Feizi. Dual manifold adversarial robustness: Defense against lp and non-lp adversarial attacks. *Advances in Neural Information Processing Systems*, 33:3487–3498, 2020. 3

[25] 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, 3, 5, 6

[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 2

[27] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 5

[28] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*, 2022. 3

[29] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951. 5

[30] Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial

attacks and defenses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4322–4330, 2019. 2, 5

[31] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 5

[32] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018. 3

[33] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 8

[34] Changhao Shi, Chester Holtz, and Gal Mishne. Online adversarial purification based on self-supervision. *arXiv preprint arXiv:2101.09387*, 2021. 3, 5, 6

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[36] Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, et al. Guided adversarial attack for evaluating and enhancing adversarial defenses. *Advances in Neural Information Processing Systems*, 33:20297–20308, 2020. 3

[37] 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

[38] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1

[39] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018. 3

[40] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 8

[41] BS Vivek, Arya Baburaj, and R Venkatesh Babu. Regularizer to mitigate gradient masking effect during single-step adversarial training. In *CVPR Workshops*, pages 66–73, 2019. 3

[42] Jinyi Wang, Zhaoyang Lyu, Dahua Lin, Bo Dai, and Hongfei Fu. Guided diffusion model for adversarial purification. *arXiv preprint arXiv:2205.14969*, 2022. 3

[43] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International conference on learning representations*, 2019. 3, 5, 6

[44] Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. *arXiv preprint arXiv:2302.04638*, 2023. 3, 5, 7

[45] Zeming Wei, Yifei Wang, Yiwen Guo, and Yisen Wang. Cfa: Class-wise calibrated fair adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8193–8201, 2023. 3

[46] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020. 2, 3

[47] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020. 3

[48] Kaiwen Wu, Allen Wang, and Yaoliang Yu. Stronger and faster wasserstein adversarial attacks. In *International Conference on Machine Learning*, pages 10377–10387. PMLR, 2020. 1, 2, 5

[49] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*, 2018. 1, 2, 5

[50] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2730–2739, 2019. 5

[51] Yuancheng Xu, Yanchao Sun, Micah Goldblum, Tom Goldstein, and Furong Huang. Exploring and exploiting decision boundary dynamics for adversarial robustness. *arXiv preprint arXiv:2302.03015*, 2023. 3

[52] Kaiwen Yang, Tianyi Zhou, Yonggang Zhang, Xinmei Tian, and Dacheng Tao. Class-disentanglement and applications in adversarial detection and defense. *Advances in Neural Information Processing Systems*, 34:16051–16063, 2021. 3

[53] Yuzhe Yang, Guo Zhang, Dina Katabi, and Zhi Xu. Me-net: Towards effective adversarial robustness with matrix estimation. *arXiv preprint arXiv:1905.11971*, 2019. 5, 7

[54] Jianhe Yuan and Zhihai He. Ensemble generative cleaning with feedback loops for defending adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 581–590, 2020. 3

[55] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5

[56] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019. 3, 5, 6

[57] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International conference on machine learning*, pages 11278–11287. PMLR, 2020. 3

[58] Dawei Zhou, Tongliang Liu, Bo Han, Nannan Wang, Chunlei Peng, and Xinbo Gao. Towards defending against adversarial examples via attack-invariant features. In *International Conference on Machine Learning*, pages 12835–12845. PMLR, 2021. 3

[59] Dawei Zhou, Nannan Wang, Chunlei Peng, Xinbo Gao, Xiaoyu Wang, Jun Yu, and Tongliang Liu. Removing adversarial noise in class activation feature space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7878–7887, 2021. 3