CAN STANDARD TRAINING WITH CLEAN IMAGES OUTPERFORM ADVERSARIAL ONE IN ROBUST ACCURACY?

Anonymous authors
Paper under double-blind review

ABSTRACT

The deep learning network has achieved great success in almost every field. Unfortunately, it is very vulnerable to adversarial attacks. A lot of researchers have devoted themselves to making the network robust. The most effective one is adversarial training, where malicious examples are generated and fed to train the network. However, this will incur a big computation load. In this work, we ask: "Can standard training with clean images outperform adversarial one in robust accuracy?" Surprisingly, the answer is YES. This success stems from two innovations. The first is a novel loss function that combines the traditional cross-entropy with the feature smoothing loss that encourages the features in the intermediate layer to be uniform. The collaboration between these terms sets up the grounds for our second innovation, namely Active Defense. When a clean or adversarial image feeds to the network, the defender first adds some random noise, then induces this sample to a new smoother one via promotion of feature smoothing. At that point, it can be classified correctly with high probability. Thus the perturbations carefully generated by the attacker can be diminished. While there is an inevitable clean accuracy drop, it is still comparable with others. The great benefit is the robust accuracy outperforms most of the existing methods and is quite resilient to the increase of perturbation budget. Moreover, adaptive attackers also fail to generate effective adversarial samples as the induced perturbations overweight the initial ones imposed by an adversary.

1 INTRODUCTION

The seminal work of (Goodfellow et al., 2015) pointed out a surprising weakness of modern deep neural networks: although they can perform on par with human beings, their reliability is far from satisfaction. Almost imperceptibly added perturbations will be enough to mislead the network to output a wrong class label with high confidence. It will dramatically undermine the deployment of networks in some safety-critical applications: autonomous driving, image-based ID verification, and medical image analysis.

Since then, researchers have heavily investigated this risk exposure and proposed different defense strategies. One direction is some prepossessing techniques such as bit-depth reduction (Xu et al., 2018), JPEG compression, total variance minimization, image quilting (Guo et al., 2018), and Defense-GAN (Samangouei et al., 2018). The idea is to mitigate the effect of added noise and save the network to some extent. Unfortunately, (Athalye et al., 2018) showed that most of these approaches are based on obfuscated gradients and can be defeated.

The other line of research adopts various adversarial training techniques where malicious examples are generated and fed to the network. A simple rationale behind this is if the network has this knowledge, it will become wise in test time. While there are different mechanisms such as k-winners-take-all (Xiao et al., 2020), network ensemble (Pang et al., 2019), Mixup inference (Pang et al., 2020), feature scattering (Zhang & Wang, 2019), Adversarial Vertex Mixup (Lee et al., 2020), and feature denoising (Xie et al., 2019), they all share the same philosophy.

While people are astonished that imperceptibly added perturbations can fool the network, some theoretical works such as (Tsipras et al., 2019; Schmidt et al., 2018) showed that it is not entirely unexpected. Unfortunately, there are no solutions without the awareness of attack models. Ideally, all defenses should be ignorant of this. But this knowledge is essential to the adversarial training

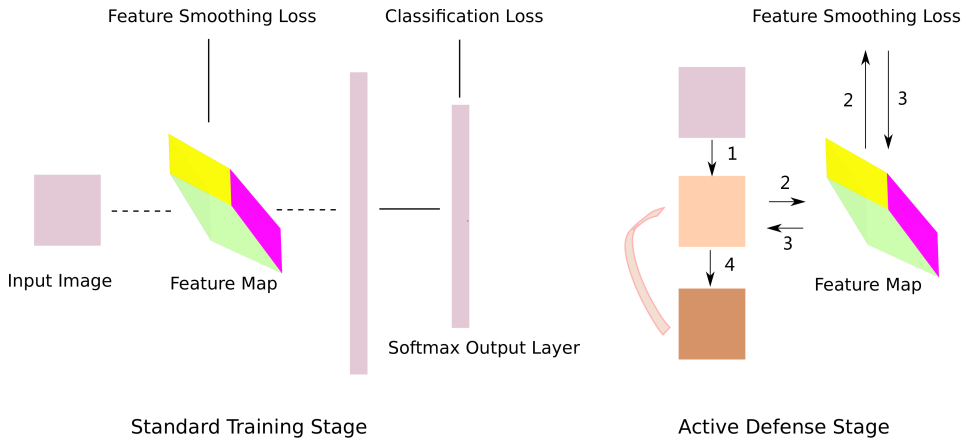


Figure 1: Schematic diagram of our approach. On the left is our standard training stage. The input image undergoes a series of deep neural network units, for example, convolution, ReLu, batch normalization, etc., until the final softmax layer outputs the decision probability. Each input image will produce corresponding feature maps. In addition to the usual classification loss, we introduce the feature smoothing loss to constrain the intermediate feature space. On the right is our Active Defense stage. In total, there are four steps. In step 1, some random noise injects into the input. This noisy image feeds into the network. During the forward pass (step 2), we get the feature smoothing loss, and the derivative to the input in backward pass(step 3) will be adopted to smooth this noisy image (step 4). This smoothing process (steps 2,3, and 4) can iterate many times until it meets some criterion, then the final result is fed into the network for classification.

method that remains the most effective, although at the cost of a large computation load. Here the big question arises:

Can standard training with clean images outperform adversarial one in robust accuracy?

At first glance, it seems hopeless, as a widely accepted principle in the adversarial learning community is that a network can be clever only it has been exposed to deceptions before. However, on the other hand, the networks are supposed to generalize well after standard training. How can it perform so badly for adversarial attacks? As a possible answer to this, (Ilyas et al., 2019) investigated the cause of adversarial examples and concluded that neural networks tend to exploit predictive yet brittle features to make classifications. These are incomprehensible to humans thus can be modified by adversarial attackers to mislead the networks, but (Ilyas et al., 2019) did not show how to teach the network to disregard these non-robust features and discover the robust one to make final decisions.

From this perspective, as it is difficult to tell the network to learn robust features, what if we add some hints in the loss function and let the network become robust in an implicit way? More specifically, in addition to the classical cross-entropy loss, we use a feature smoothing term that encourages the features in an intermediate layer to be uniform, as shown in the left of Figure 1. It sounds counterintuitive as this term will constrain the space of features that may lead to a wrong classification. However, due to the high capacity of networks, a very high standard accuracy can also be achieved with this additional term.

When training completes, given an input, whether clean or crafted, extra perturbations can always be created by some added random noise followed by the promotion of feature smoothing at the cost of reduced accuracy. So long as these intentional perturbations overweight the adversary's and the reduction in accuracy is affordable, the model will become robust. We call this procedure Active Defense, as shown in the right of Figure 1. We find experimentally, a clean sample can be perturbed with $L_\infty = 25 \sim 32$, three to four times of $L_\infty = 8$ usually adopted by an adversary, yet be classified with a high success rate. This fact sets up the adequate space for Active Defense for eliminating the effects of attacks. Our approach is independent of any attack models compared with other state-of-the-art methods, and its performance is much more stable under attacks with different budgets. The contributions of this work are summarized as follows:

• We propose a novel training scheme with an extra feature smoothing loss term that only takes clean images as inputs, fundamentally different from all existing adversarial training methods that need supplementary crafted data.

• We present Active Defense that adds the second round of perturbations through random noise and feature smoothing. It modifies the malicious samples in a way that is friendly to the network. This deviates from conventional passive ones that keep the input intact.

2 Related Works

Due to adversarial threats to deep learning applications, there are many works to improve the robustness. Most of them adopt adversarial training. Among them, only a few take care of the features in intermediate layers as listed in the following.

Feature denoising in (Xie et al., 2019) found that small perturbations in the pixel space can lead to very substantial noise in the feature maps of the network and proposed various filters to denoise. (Zhang & Wang, 2019) proposed to generate adversarial images for training through feature scattering in the latent space. In essence, perturbed images are produced collaboratively via optimal transport distance minimization. (Zhang & Wang, 2019) used the feature maps as a guide to making new examples. Compared with these two, we are trying to force the intermediate feature map to be uniform through an additional loss term within the standard training framework without any modification of the network as in (Xie et al., 2019) or any other manipulations of features as in (Zhang & Wang, 2019).

Regarding Active Defense, we have not seen any similar work. Perhaps the most related one is (Yang et al., 2019), which used sophisticated matrix completion techniques to reconstruct the random masked images. Our motivation is very different, as we try to exploit the deep network itself to enhance robustness without borrowing any third-party algorithms.

3 Background

In the classification problem, given the training data set of image-label pairs $D = \{(x_i, y_i)\}_{i=1}^n$ where $y_i \in \{1, 2, ..., M\}$, the goal is to find an output probability vector $F(x)$ of length M indexed by j, ideally such that $y = \arg\max_j F_j(x)$. Of course, there is always a mismatch between these two terms. The key thing here is to find a suitable loss function such that the empirical risk minimization (ERM) of $\frac{1}{n}\sum_{i=1}^{n} L(F(x_i), y_i)$ can be implemented with loss function L. Note that $F(x_i)$ is a vector, while y_i is a scalar of the label, the very first thing is to transform y_i into a vector through a vector function $G(y_i)$. People usually adopt the one-hot coding $H(y_i)$ of length M with all elements being 0 except $H_{y_i}(y_i) = 1$. The two probability distribution vectors $F(x_i)$ and $H(y_i)$ can be compared with cross-entropy.

An adversary crafts an adversarial example x_{adv} which is closest to x with $\|x_{adv} - x\|_p \leq \varepsilon$ but misclassified as some other class. In this paper, we only consider attacks with $p = \infty$. The most commonly used strategy is the iterative projected gradient descent method(PGD)

$$x_{adv}^{t+1} = P(x_{adv}^t + \beta \times \text{sign}(\nabla_x L(x_{adv}^t, G(y)))), \tag{1}$$

where β is the step size and P projects the generated example to the feasible region. Please note that L in adversary attack is different from L in training. From the white-box attacker's view, the only knowledge is parameters. People may choose traditional cross-entropy loss with one-hot coding or CW loss (Carlini & Wagner, 2017) to implement Equation 1. Currently, A budget-aware step size free variant of PGD has been proposed by (Croce & Hein, 2020), and since that, an ensemble of diverse parameter-free attacks called AutoAttak has become the de facto routine for robust accuracy evaluation.

4 METHOD

In general, our method is very simple. In training, for feature map F^l of a particular layer l with $W \times H \times C$,we use the loss

$$L = L_{ce} + max\left(L_{F^l}, \delta\right) \tag{2}$$

$$L_{F^l} = \frac{1}{W \times H \times C} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{C} \left| {F^l}_{i,j,k} - mean(F^l) \right|. \tag{3}$$

Here L has two terms. L_{ce} is for cross-entropy loss, and L_{F^l} is our novel feature smoothing loss function. It is quite similar to the L_1 norm of the particular feature cube and encourages the cube to be uniform. In order to avoid overfitting to the feature smoothing loss, we use $max\left(L_{F^l}, \delta\right)$ which disables the derivative of L_{F^l} when it drops below δ. In summary, we have two parameters, the feature layer l and the smoothing upper bound δ.

Although this loss function sounds straightforward for our purpose to make the intermediate feature map smooth, there is a novel insight from the perspective of the trade-off between these two terms. In order to get low L_{F^l}, L_{ce} will increase. In other words, after training, the network somehow understands that feature smoothing is not very annoying, and it only causes accuracy to drop to some extent. There is a huge implication in terms of robust accuracy. If we only use L_{ce}, the network has no ideas to deal with the crafted sample except to be fooled. However, in our case, L_{F^l} gives us a dissipation channel of malicious perturbations that we can take via feature smoothing. Hopefully, this will remove most of the perturbations generated by an adversary, which we will elaborate on in our Active Defense design. The other concern may relate to the feature space constraints. Actually, due to the high capacity of networks, there is almost no difference in standard training accuracy between ours and cross-entropy loss.

From the discussion above, our Active Defense is very intuitive. It consists of four steps depicted in the right of Figure 1. In step 1, we add some random noise which can somehow reduce the effect of adversarial disruptions. Steps 2 and 3 are just forward / backward passes related to L_{F^l}. In step 4, the smoother noisy image feeds the network for another round of feature smoothing. The overall procedure for the proposed approach is in Algorithm 1. There are only three parameters, i.e., σ for uniform noise, β for the updating step size, and $\tilde{\delta}$ for the upper bound of feature loss in the test that is usually lower than δ for training in Equation 2. The updated sample through Active Defense will feed to the network for final class decision.

Algorithm 1 Active Defense Algorithm

procedure ACTIVE DEFENSE(t, L_{F^l}) ▷ t is a test sample.
  $t = t + uniform(-\sigma, \sigma)$ ▷ **Step 1**, σ is the parameter of the uniform distribution.
  $s = L_{F^l}\left(t\right)$ ▷ **Step 2**
  while $l > \tilde{\delta}$ **do** ▷ $\tilde{\delta}$ is the upper bound of feature loss in test.
    $d = \frac{\partial l}{\partial t}$ ▷ **Step 3**
    $t = t - \beta \times d$ ▷ **Step 4**, β is the step size.
    $s = L_{F^l}\left(t\right)$ ▷ **Step 2**
  end while
  return t
end procedure

5 EXPERIMENTS

To evaluate the performance of our approach, we run it on two datasets, CIFAR10 and CIFAR100 (Krizhevsky, 2009), and compare it with other state-of-the-art adversarial training methods. The network we choose is WideResNet-28-10 with three groups. Naturally, feature maps F^l $(l = 0, 1, 2)$ are used to denote the outputs of three groups respectively, and we choose $l = 0$, as it is closest to the input with the most strong backward pass derivative. We evaluate the robust accuracy with perturbation budget ε as both 8 and 16. It needs to emphasize here that our algorithm is independent of ε.

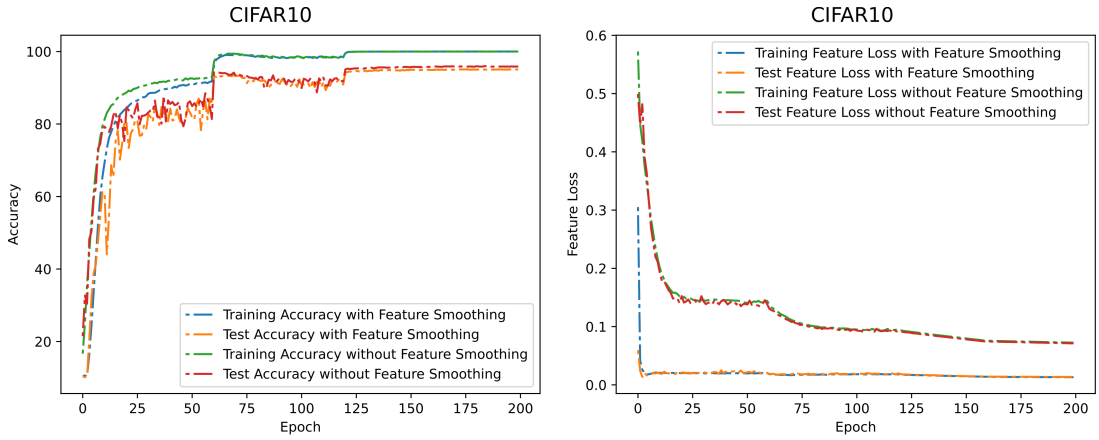


Figure 2: The left is training and test accuracy, and the right is feature loss for CIFAR10.

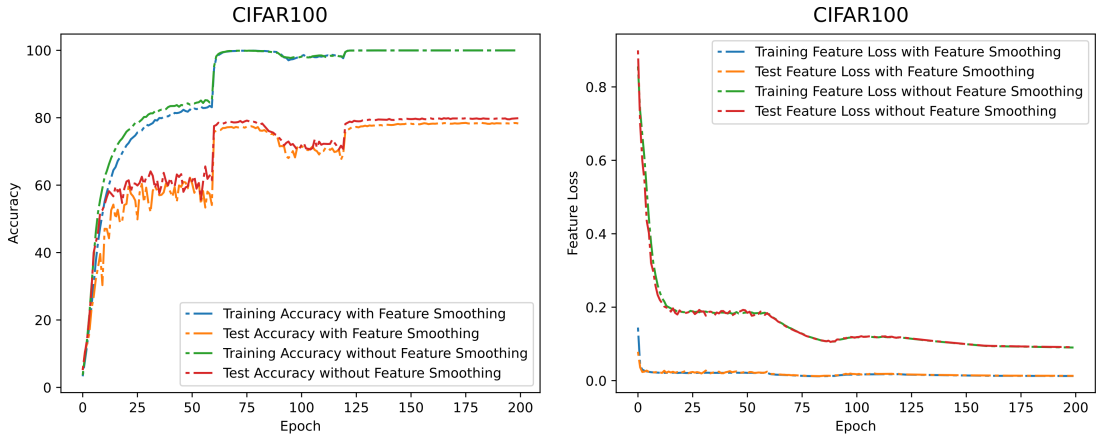


Figure 3: The left is training and test accuracy, and the right is feature loss for CIFAR100.

5.1 Standard Training

Our approach adopts standard training with clean images using an additional feature loss term which boosts the intermediate features to be smooth. In all our experiments, the smoothing upper bound δ in Equation 2is set to 0.02. Figures 2 and 3 demonstrate the training and test accuracy and feature loss variation per training epoch.

As we expected, while they achieve almost 100% training accuracy, there is only a tiny drop in the test accuracy with and without feature smoothing terms: 95.02% vs 95.83% for CIFAR10, and 78.3% vs 79.86% for CIFAR100. This is due to the fact that the classifiers have a large number of parameters and are powerful enough. Here is an interesting observation of feature loss: even when we do not impose any constraints, it still goes down slowly. While we do enforce this term, it goes down very quickly at the beginning. It strongly hints feature smoothing benefits the generalization ability of networks. The feature loss with and without feature smoothing terms are dramatically different, listed in the pattern of training/test, 0.0130/0.0135 vs 0.0725/0.0713 for CIFAR10, and 0.01258/0.013 vs 0.0901/0.0909 for CIFAR100. This fact indicates the effects of this additional loss term. Also, although we set the δ to be 0.02, the feature loss is lower than that number, especially for CIFAR100, which is also evidence that classification and feature smoothing are cooperative to some extent. Our Active Defense design exactly takes advantage of this harmony.

Table 1: AA comparison (WideResNet-28-10 only), * denotes model requires additional external images.

#	paper	clean	APGD_{CE}	$\text{APGD}^{\text{T}}_{\text{DLR}}$	FAB^{T}	Square	AA
CIFAR-10 - l_∞ - $\varepsilon = 8/255$							
1	Ours(outerloop = 10)	86.69	73.71	77.64	85.30	79.03	68.54
2	(Gowal et al., 2020)*	89.48	65.64	62.83	63.29	68.74	62.76
3	Ours(outerloop = 1)	85.35	73.10	74.11	83.49	76.73	62.31
4	(Rade & Moosavi-Dezfooli, 2021)	88.16	63.75	60.99	61.39	66.67	60.97
5	(Rebuffi et al., 2021)	87.33	63.99	60.76	61.14	66.52	60.73
*	Standard(using Active Defense)	16.20	16.13	15.94	16.10	15.79	7.40
CIFAR-100 - l_∞ - $\varepsilon = 8/255$							
1	Ours(outerloop = 10)	59.43	39.36	44.78	57.63	48.92	33.49
2	(Rebuffi et al., 2021)	62.41	35.66	32.08	32.30	36.76	32.06
3	Ours(outerloop = 1)	56.98	37.90	42.85	55.04	46.80	28.51
4	(Hendrycks et al., 2019)*	59.23	33.02	28.48	28.74	34.26	28.42
5	(Rice et al., 2020)	53.83	20.57	18.98	19.24	23.57	18.95
*	Standard(using Active Defense)	2.69	2.54	2.65	2.78	2.56	1.39

5.2 Active Defense

Equation 2 takes two terms, L_{F^l} and L_{ce}. What is the consequence if we make the image smoother further, i.e., to decrease L_{F^l}? The natural effect will be the increase of cross-entropy loss L_{ce}, but the drop in classification accuracy should be small since L_{F^l} and L_{ce} are comfortable with each other. It is key to our success of Active Defense. To counter adversarial perturbations, we intentionally add some noise and smooth it out as in Algorithm 1. In all our experiments, σ = 0.075 for uniform noise and β = 80 for the updating step size, while for the upper bound of feature loss $\tilde{\delta}$, 0.0118 for CIFAR100 and 0.0124 for CIFAR10.

Some experimental results with $\varepsilon = 8$ in Figures 4 and 5 show the Active Defense can effectively recover the semantically significant structures destroyed by the adversarial attacks and our intentionally added noise, which leads to correct final classification. Moreover, comparing with the clean image, the lobsters in feature maps are much more integrated, as shown in the last row of Figure 5. This evidence highlights the role played by our Active Defense. Usually, a single Active Defense takes about 200-400 iterations of while loop body in Algorithm 1 for $\varepsilon = 8$ and up to around 500 for $\varepsilon = 16$.

We evaluate the robust accuracy of our approach using AutoAttack and compare the performance with other state-of-the-art methods in Tables 1 and 2. Since our solution is stochastic, to further improve the stability of our Active Defense, we run it through 10 times, which we denote as outerloop=10, and aggregate the output probability of each run. As AutoAttack comprises a set of attack methods, it will win if any of them succeed. It poses some challenges for our approach as each adversarial sample may get some chance to win even though its attack model is not so strong on an average of 10k images. For example, in Table 1, outerloop=1 for CIFAR10 achieves the lowest accuracy of 73.10 for attack model APGD_{CE}, but AA only gets 62.31, almost a 10 percent drop. However for other comparison methods, this gap is small. This situation improves a lot when we increase outerloop to 10.

In Table 1, we only consider WideResNet-28-10, and ours with outerloop=10 is the best. One may still wonder what if we apply our Active Defense with classic training only with cross-entropy loss. We also list the results in the rows of Standard(using Active Defense). It turns out that it is hard to exit the while loop in Algorithm 1, so we set $\tilde{\delta}$ to be 0.0562 for CIFAR10 and 0.0880 for CIFAR100. The accuracy is very low. This fact justifies the necessity of our feature smoothing loss term. Table 2 lists the AA accuracy with all model architectures, and ours is still among the best. Note that when $\varepsilon = 16$, ours significantly outperforms all others.

Table 2: AA comparison, * denotes model requires additional external images.

#	paper	architecture	clean	AA
CIFAR-10 - l_∞ - $\varepsilon = 8/255$				
1	Ours(outerloop = 10)	WideResNet-28-10	86.69	68.54
2	(Rebuffi et al., 2021)*	WideResNet-70-16	92.23	66.56
3	(Gowal et al., 2020)*	WideResNet-70-16	91.10	65.87
4	(Rebuffi et al., 2021)	WideResNet-106-16	88.50	64.58
5	(Rebuffi et al., 2021)	WideResNet-70-16	88.54	64.20
6	(Rade & Moosavi-Dezfooli, 2021)*	WideResNet-34-10	91.47	62.83
7	(Gowal et al., 2020)*	WideResNet-28-10	89.48	62.76
8	Ours(outerloop = 1)	WideResNet-28-10	85.35	62.31
9	(Rade & Moosavi-Dezfooli, 2021)	WideResNet-28-10	88.16	60.97
10	(Rebuffi et al., 2021)	WideResNet-28-10	87.33	60.73
CIFAR-10 - l_∞ - $\varepsilon = 16/255$				
1	Ours(outerloop=10)	WideResNet-28-10	86.79	54.24
2	Ours(outerloop=1)	WideResNet-28-10	85.21	49.24
3	(Rebuffi et al., 2021)	WideResNet-28-10	87.33	25.39
4	(Gowal et al., 2020)*	WideResNet-28-10	89.48	24.96
5	(Wu et al., 2020)*	WideResNet-28-10	88.25	24.00
CIFAR-100 - l_∞ - $\varepsilon = 8/255$				
1	(Gowal et al., 2020)*	WideResNet-70-16	69.15	36.88
2	(Rebuffi et al., 2021)	WideResNet-70-16	63.56	34.64
3	Ours(outerloop = 10)	WideResNet-28-10	59.43	33.49
4	(Rebuffi et al., 2021)	WideResNet-28-10	62.41	32.06
5	(Cui et al., 2020)	WideResNet-34-20	62.55	30.20
6	(Gowal et al., 2020)	WideResNet-70-16	60.86	30.03
7	(Cui et al., 2020)	WideResNet-34-10	60.64	29.33
8	(Rade & Moosavi-Dezfooli, 2021)	PreActResNet-18	61.50	28.88
9	(Wu et al., 2020)	WideResNet-34-10	60.38	28.86
10	Ours(outerloop = 1)	WideResNet-28-10	56.98	28.51
11	(Rebuffi et al., 2021)	PreActResNet-18	56.87	28.50
CIFAR-100 - l_∞ - $\varepsilon = 16/255$				
1	Ours(outerloop=10)	WideResNet-28-10	59.16	22.76
2	Ours(outerloop=1)	WideResNet-28-10	57.10	19.15
3	(Rebuffi et al., 2021)	WideResNet-28-10	62.41	12.45
4	(Hendrycks et al., 2019)*	WideResNet-28-10	59.23	8.89

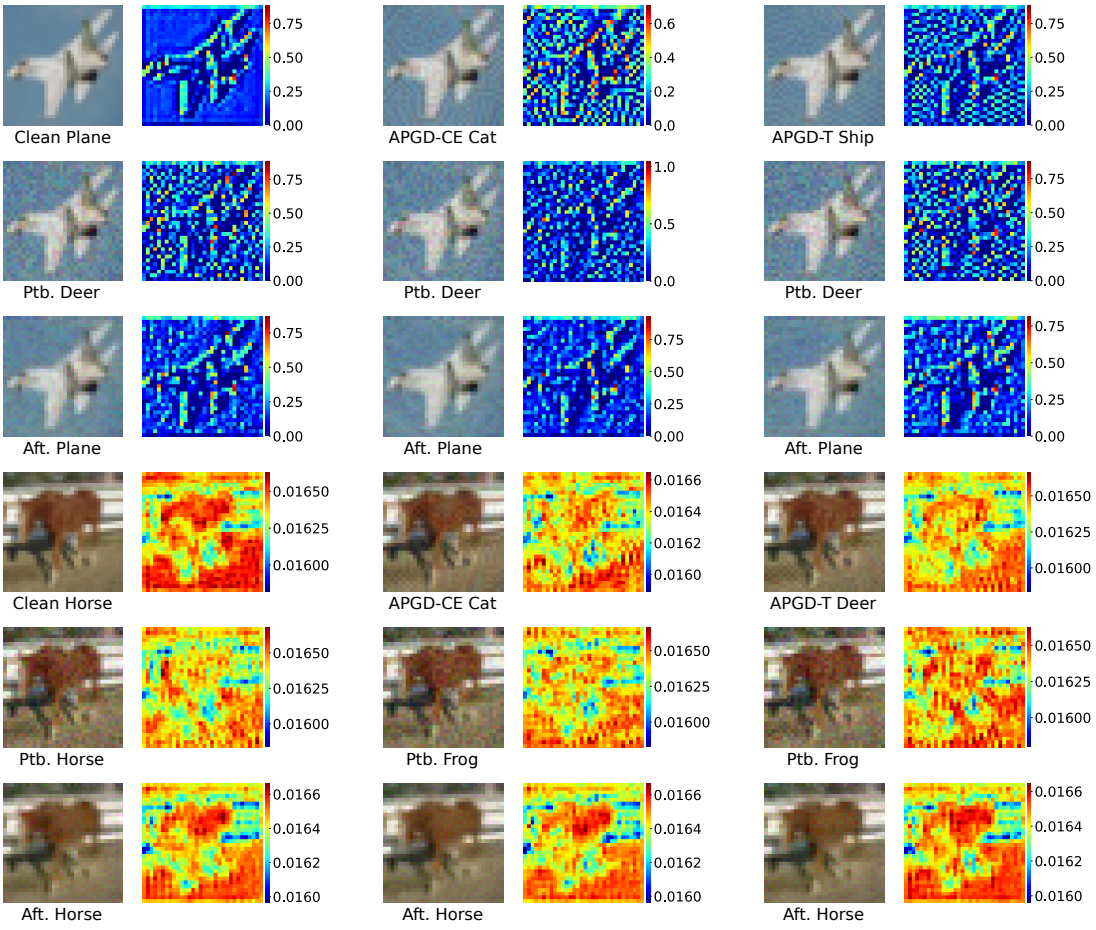
5.3 Adaptive Adversaries

While it is always not easy to propose adaptive attacks, we try our best to defeat our defense scheme. Since the APGD_{CE} seems to be the best attack as listed in Table 1, we adapted it to generate possibly stronger adversarial examples. Specifically, the loss function is modified to

$$L = L_{ce} + \lambda \times max\left(L_{F^l}, \delta\right). \tag{4}$$

In other words, we try to understand how the feature loss term influences the efficiency of attacks with λ spaced uniformly (on a log scale) from 0.1 to 100, both positive and negative. Positive λ means more distortion in feature space, which pressures our smoothing process via Active Defense. While for negative λ, the iterations of the while loop in Algorithm 1 should be small, which results in more effective attacks. These efforts are useless, manifested in robust accuracy of Table 3, almost the same as $\lambda = 0$.

Figure 4: The top half is a plane from CIFAR10. In the first row, from left to right, we demonstrate the clean and adversarial images generated via APGD_{CE} and $\text{APGD}^{\text{T}}_{\text{DLR}}$, and the corresponding feature map in channel 7. The second and third rows are the noise-perturbed and after Active Defense ones, respectively. The second half is arranged similarly except that the feature channel is 26. Our standard training model assigns each image a label shown underneath it.



$\log_{10}(\lvert\lambda\rvert)$	-1	-0.25	0.5	1.25	2	**-1**	**-0.25**	**0.5**	**1.25**	**2**
CIFAR10	72.95	73.06	72.62	72.95	72.79	**72.87**	**72.91**	**72.7**	**72.76**	**72.43**
CIFAR100	37.72	38.06	38.02	37.32	37.72	**37.79**	**37.69**	**37.66**	**37.64**	**37.32**

Table 3: Robust accuracy on adaptive attacks with different λ through a single run of Active Defense. The left half table is for negative λ, while the remaining one is for positive λ in bold.

5.4 Discussion

All above attacks are one-time in nature, as attackers may take an arbitrarily long time and possibly get many options but can only shoot one. Since our method introduces random noise in the Active Defense phase, the results demonstrate the average success rate for 10K test samples. One may come up with a simple attack that sends the same sample many times, and there is always a chance to defeat. But this brute force attack can be resolved by our enhanced version of defense, that is, to run a lot of times of Algorithm 1, and aggregate the probability of each run. However, this will increase the computation load. On the other hand, it makes sense since there is no free lunch. Also, very fortunately, our capacity of defense can scale up much more conveniently than perhaps retraining the model from scratch as conventional adversarial training methods.

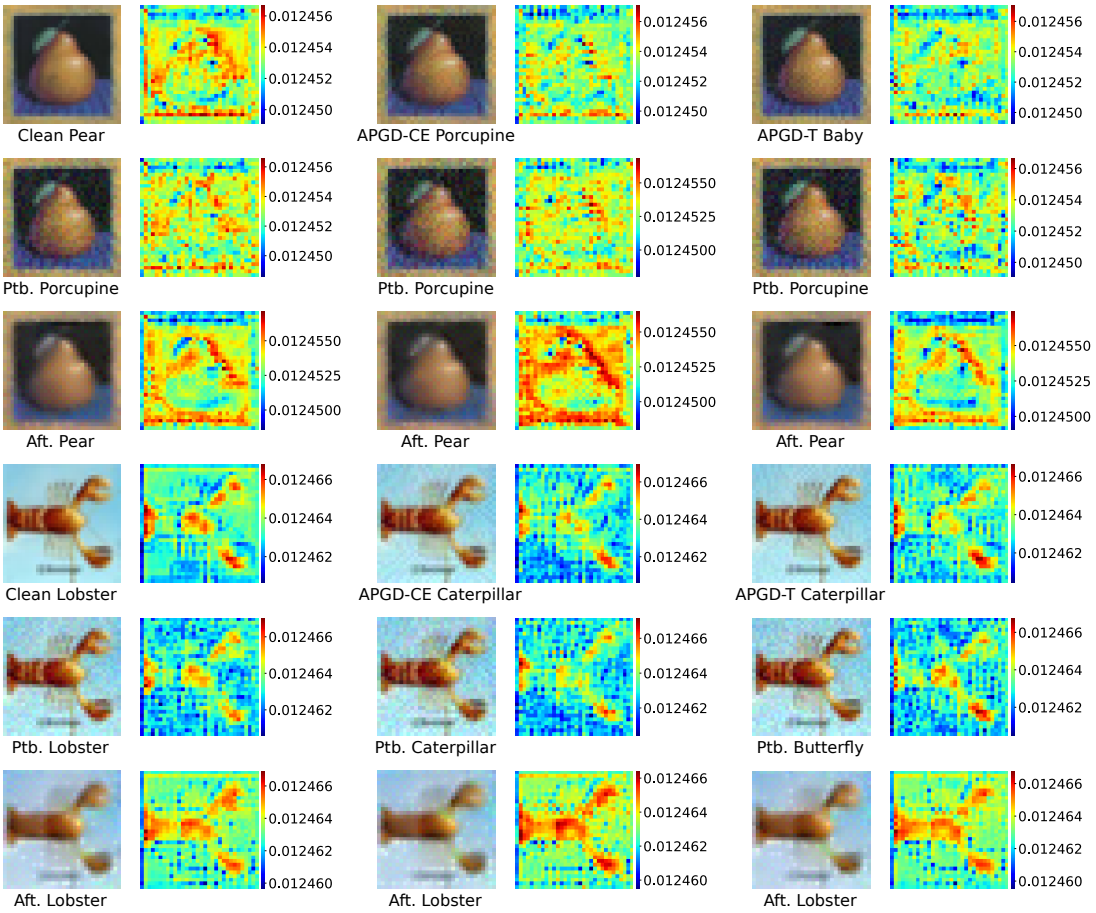


Figure 5: The top half is a plane from CIFAR100. In the first row, from left to right, we demonstrate the clean and adversarial images generated via APGD_{CE} and $\text{APGD}^{\text{T}}_{\text{DLR}}$, and the corresponding feature map in channel 43. The second and third rows are the noise-perturbed and after Active Defense ones, respectively. The second half is arranged similarly except that the feature channel is 33. Our standard training model assigns each image a label shown underneath it.

6 Conclusion

Adversarial learning is of great interest to the deep learning community. Most of the previous works focus on the efficient generation of malicious examples. However, in this paper, we shed some light on a question: Is it possible that a network can be robust without being taught with malicious examples? We propose a standard training model via an additional feature smoothing loss term, which is very different from all existing ones in that there is NO ADVERSARIAL INPUT involved. The standard cross-entropy and feature smoothing loss can collaborate to some extent in training. At test time, we adopt Active Defense to distill feature maps of adversarial inputs. The experimental results demonstrate that this simple method can enhance the robustness of networks greatly. In future work, we will do some theoretical analysis and exploit other forms of cooperative loss that may be more beneficial than the feature smoothing one.

References

Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of*

Machine Learning Research, pp. 274–283. PMLR, 2018. URL http://proceedings.mlr.press/v80/athalye18a.html.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pp. 39–57. IEEE Computer Society, 2017. doi: 10.1109/SP.2017.49.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2206–2216. PMLR, 2020. URL http://proceedings.mlr.press/v119/croce20b.html.

Jiequan Cui, Shu Liu, Liwei Wang, and Jiaya Jia. Learnable boundary guided adversarial training. *CoRR*, abs/2011.11164, 2020. URL https://arxiv.org/abs/2011.11164.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy A. Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *CoRR*, abs/2010.03593, 2020. URL https://arxiv.org/abs/2010.03593.

Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=SyJ7ClWCb.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2712–2721. PMLR, 2019. URL http://proceedings.mlr.press/v97/hendrycks19a.html.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 125–136, 2019. URL http://papers.nips.cc/paper/8307-adversarial-examples-are-not-bugs-they-are-features.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Saehyung Lee, Hyungyu Lee, and Sungroh Yoon. Adversarial vertex mixup: Toward better adversarially robust generalization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 269–278. IEEE, 2020. doi: 10.1109/CVPR42600.2020.00035.

Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4970–4979. PMLR, 2019. URL http://proceedings.mlr.press/v97/pang19a.html.

Tianyu Pang, Kun Xu, and Jun Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=ByxtC2VtPB.

Rahul Rade and Seyed-Mohsen Moosavi-Dezfooli. Helper-based adversarial training: Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021. URL https://openreview.net/forum?id=BuD2LmNaU3a.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A. Calian, Florian Stimberg, Olivia Wiles, and Timothy A. Mann. Fixing data augmentation to improve adversarial robustness. *CoRR*, abs/2103.01946, 2021. URL https://arxiv.org/abs/2103.01946.

Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8093–8104. PMLR, 2020. URL http://proceedings.mlr.press/v119/rice20a.html.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=BkJ3ibb0-.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pp. 5019–5031, 2018. URL http://papers.nips.cc/paper/7749-adversarially-robust-generalization-requires-more-data.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=SyxAb30cY7.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1ef91c212e30e14bf125e9374262401f-Abstract.html.

Chang Xiao, Peilin Zhong, and Changxi Zheng. Enhancing adversarial defense by k-winners-take-all. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=Skgvy64tvr.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 501–509. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00059. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Xie_Feature_Denoising_for_Improving_Adversarial_Robustness_CVPR_2019_paper.html.

Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society, 2018. URL http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018_03A-4_Xu_paper.pdf.

Yuzhe Yang, Guo Zhang, Zhi Xu, and Dina Katabi. Me-net: Towards effective adversarial robustness with matrix estimation. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long*

Beach, California, USA, volume 97 of *Proceedings of Machine Learning Research*, pp. 7025–7034. PMLR, 2019. URL http://proceedings.mlr.press/v97/yang19e.html.

Haichao Zhang and Jianyu Wang. Defense against adversarial attacks using feature scattering-based adversarial training. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 1829–1839, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/d8700cbd38cc9f30cecb34f0c195b137-Abstract.html.