Improving Robustness of Convolutional Neural Networks Using Element-Wise Activation Scaling

Abstract

Recent works reveal that re-calibrating the intermediate activation of adversarial examples can improve the adversarial robustness of a CNN model. The state of the arts [Bai et al., 2021] and [Yan et al., 2021] explores this feature at the channel level, i.e. the activation of a channel is uniformly scaled by a factor. In this paper, we investigate the intermediate activation manipulation at a more fine-grained level. Instead of uniformly scaling the activation, we individually adjust each element within an activation and thus propose Element-Wise Activation Scaling, dubbed EWAS, to improve CNNs' adversarial robustness. Experimental results on ResNet-18 and WideResNet with CI-FAR10 and SVHN show that EWAS significantly improves the robustness accuracy. Especially for ResNet18 on CIFAR10, EWAS increases the adversarial accuracy by 37.65% to 82.35% against C&W attack. EWAS is simple yet very effective in terms of improving robustness. The codes are anonymously available at https://anonymous. 4open.science/r/EWAS-DD64.

1 Introduction

Convolutional neural networks (CNNs) have demonstrated its superiority in various applications, especially for computer vision tasks, like classification, object detection and segmentation [Krizhevsky et al., 2017; Dosovitskiy et al., 2021]. However, CNNs are found to be vulnerable to adversarial samples that are perturbed by unperceptive noises [Szegedy et al., 2014]. Adversarial attacks significantly undermine the model's robustness and threat the applicability of CNNs to some safety-critical and security-critical contexts, e.g. self-driving, person identification, etc. A plenty of efforts have been made to improve CNNs' adversarial robustness and these efforts can be generally divided into two categories: adversarial attacks and adversarial defence. Various methods are proposed to generate diverse adversarial samples [Szegedy et al., 2014; Carlini and Wagner, 2017; Madry et al., 2018; Moosavi-Dezfooli et al., 2016; Su et al., 2019; Xiao et al., 2018; Jandial et al., 2019; Croce and Hein, 2020].

On the other hand, many works aim to defend the adversarial attacks. A number of defense methods have been proposed, such as defensive distillation [Papernot *et al.*, 2016; Goldblum *et al.*, 2020], feature denoising [Xie *et al.*, 2019; Liao et al., 2018], GAN [Liu and Hsieh, 2019], model compression [Madaan et al., 2020; Ye et al., 2019; Gui et al., 2019], authentication defense [Chen et al., 2019], and adversarial training (AT) and its variants [Madry et al., 2018; Zhang et al., 2019; Wang et al., 2020; Wong et al., 2020]. Recently, some works investigate the difference between natural models and AT-trained counterparts in terms of intermediate activation and propose to adjust intermediate activation for better adversarial robustness. [Kannan et al., 2018] proposes to make the logit of natural samples and adversarial samples similar. The adversarial perturbations of input images are deemed as noises and hence [Xie et al., 2019] suggests to denoise the distorted features using non-local means or other filters to improve robustness. [Liao *et al.*, 2018] proposes to deploy high-level representations to guide the denoising procedure. [Bai et al., 2021] observes that adversarial examples wrongly activate 'negative' features which lead to the final misclassification and thus proposes Channel-wise Activation Suppressing (CAS) strategy to suppress those 'negative' features to improve a model's robustness. In parallel, [Yan et al., 2021] has similar observations and proposes a channelwise activation method, namely CIFS, to enhance the robustness. Besides suppressing the negative activation, they also promote the positive activation for higher accuracy.

All these methods apply to the channel/activation level, i.e., the whole channel or activation will be suppressed or promoted by a uniform scaling. Although such uniform activation scaling (suppression or promotion) methods do improve robustness as seen from [Bai *et al.*, 2021; Yan *et al.*, 2021], scaling uniformly, especially suppression, may lead to the information loss of the scaled activation. This inspires us to think about *can we robustly scale/calibrate activation without losing their information which may help the model to further improve its robustness*?

In this paper, we propose a new and fine-grained activation scaling method to improve the robustness of CNN models, i.e., instead of scaling each activation using a uniform scaling, we conduct an Element-Wise Activation Scaling, dubbed EWAS. By means of EWAS, the distorted activation are not completely suppressed or promoted, but are re-calibrated in a fine-grained manner. Our key contributions are summarized as follows:

- We propose the EWAS module, which can be easily added to the existing CNN models. EWAS performs activation adjustment in an element-wise fashion to improve the CNNs' robustness. The core component of EWAS is an auxiliary and class-aware classifier which is used to generate the element scaling factor.
- We conduct extensive experiments to evaluate the effectiveness of EWAS in terms of adversarial robustness, where different CNN models, datasets, AT methods and adversarial attacks are deployed. The experimental results show that our EWAS-based models can greatly improve the robustness of the evaluated models over SOTA [Bai *et al.*, 2021][Yan *et al.*, 2021]]. In the best case against C&W attack, EWAS can improve the robustness by 37.65% to 82.35% and makes its adversarial accuracy comparable to its nature accuracy, 84.73%.

Remark: [Bai *et al.*, 2021] and [Yan *et al.*, 2021] strive to minimize the activation difference between nature examples and adversarial counterparts. However, the activation analysis shows that EWAS does not follow this objective, where EWAS-modified CNNs demonstrate different activation distributions for natural and adversarial pairs. This may provide a new thought in improving the CNNs' robustness.

2 Related Work

In this section, we briefly review adversarial training methods and the adversarial defending methods relevant to EWAS.

Adversarial Training: AT [Madry *et al.*, 2018] is the most widely used method to improve CNNs' robustness. AT which is a data augmentation technique for adversarial defence aims to solve the following min-max optimization problem:

$$\min_{\boldsymbol{\theta}} E_{(x,y)\sim D}[\max_{\boldsymbol{\delta}}(L(y,F(x+\boldsymbol{\delta},\boldsymbol{\theta})))]$$
(1)

where F represents a CNN model with weight parameters θ and L is the loss function, e.g., cross-entropy loss. x and y are a natural example and its corresponding label from dataset D. $x + \delta$ represents the adversary of x with adversarial perturbation δ which is within l_p -norm distance and satisfies $\|\delta\|_p < \varepsilon$. Here, similar to previous methods, [Yan *et al.*, 2021; Bai *et al.*, 2021] we set $p = \infty$. The inner maximization problem aims to generate the strong adversary, while the outer minimization problem is the model training procedure to learn model weights θ with adversarial examples.

Different adversarial attacks can be applied to AT, such as Projected Gradient Descent (PGD) [Madry *et al.*, 2018] and fast gradient sign method (FGSM) [Wong *et al.*, 2020]. Since the emergence of AT, diverse methods have been proposed to improve the effectiveness and efficiency of AT. [Wong *et al.*, 2020] combined FGSM [Szegedy *et al.*, 2014] with random initialization to make FGSM applicable to AT with lower cost. [Wang *et al.*, 2020] observed the impact of misclassified samples on models' robustness and thus proposed a misclassification-aware AT (MART) to improve the adversarial robustness. Although AT can improve adversarial robustness, it also sacrifices the accuracy for natural examples. Two improvements, TRADES [Zhang *et al.*, 2019] and FAT [Zhang *et al.*, 2020], are proposed to address the accuracy drop for natural examples.

Robust Activation Manipulation: Some works strive to understand the difference between adversarial examples and their nature counterparts from the lens of intermediate activation. Then, some works propose to diminish such difference to improve the robustness, e.g., adversarial logit pairing [Kannan et al., 2018]. Two concurrent works, CAS [Bai et al., 2021] and CIFS [Yan et al., 2021], adopt the robust activation scaling. [Bai et al., 2021] proposed Channel-wise Activation Suppressing (CAS) strategy to suppress redundant activation that are 'negatively' activated by adversarial examples. Similarly, [Yan et al., 2021] observed that some channels, which are over-activated by adversarial examples but are not important to correct prediction, undermine the adversarial robustness. Thus, they proposed CIFS which identifies those channels and suppresses them to improve the robustness. The two above-mentioned methods both feature a channel-level scaling, i.e., the whole activation is uniformly scaled as shown in Fig. 1(a). We conjecture that these uniformly scaled channels carry some useful information which can contribute to the robust prediction, so individually adjusting each element within an activation would help improve a model's robustness. The idea is simple but effective as we can see from our extensive evaluation in Section 4 which justifies our conjecture.



(a) Channel-wise activation scaling, CAS and CIFS



(b) Element-wise activation scaling, EWAS

Figure 1: Channel-wise scaling vs element-wise scaling. Elementwise scaling conducts a more fine-grained scaling to the intermediate activation.

3 Element-Wise Activation Scaling

Fig. 2 demonstrates the overview of EWAS, where EWAS is a plug-in module being added to the existing CNN models. The EWAS module is trained with the backbone network by means of an auxiliary loss function. Each layer of a CNN can be equipped with an EWAS module, but we empirically find that for a CNN model, simply adding one EWAS module demonstrates the best adversarial robustness. Next, we proceed to the module and how to train it.

3.1 EWAS Module

Let $z^l \in \mathbb{R}^{C \times H \times W}$ denote the activation of layer l which has an EWAS module, where C denotes the number of channels

and H and W are the height and width, respectively. Each element in z^l is expected to have an individual scaling factor and thus we have $m \in \mathbb{R}^{C \times H \times W}$ to denote the scaling factor vector. As seen in [Bai *et al.*, 2021][Yan *et al.*, 2021], class-related activation modification is instrumental in improving the robustness. Hence, we also deploy an auxiliary classifier to have the class-related feature and determine the element-wise scaling factor m.

Auxiliary Linear classifier (ALC)

The core of EWAS is the scaling factor m. A good scaling factor m will suppress redundant and negative elements while retaining or promoting robust and positive elements. Inspired by CAS [Bai *et al.*, 2021], we add an auxiliary linear classifier (ALC) to the original model and use ALC to derive m. The overview of EWAS can be seen in Fig. 2. ALC takes activation z^l as the input and outputs classification scores of K classes.

Let $\theta^{ALC} \in \mathbb{R}^{C \cdot H \cdot W \times K}$ denote the parameters of ALC. ALC parameters θ^{ALC} are deployed to generate the scaling mask m. ALC is a class-related scaling classifier, i.e., we have $\theta_k^{ALC} \in \mathbb{R}^{C \cdot H \cdot W}$ to represent the parameters related to class k and θ_k^{ALC} will be converted to the scaling factor m. In the training stage, the ground truth label y serves as the class index to select which class' parameters to update. In the inference stage, since there is no label information provided, the maximum value of \hat{s} predicted by ALC is used as the class index. The scaling factor m is formulated as follows:

$$\boldsymbol{m} = \begin{cases} \operatorname{reformat}(\theta_y^{\operatorname{ALC}}), & (\operatorname{training stage})\\ \operatorname{reformat}(\theta_{\arg\max(\hat{s})}^{\operatorname{ALC}}), & (\operatorname{inference stage}) \end{cases}$$
(2)

Note that the scaling factor m is reformated into size $\mathbb{R}^{C \times H \times W}$. After obtaining the scaling factor m, we perform element-wise multiplication on z^l to obtain the adjusted activation \hat{z}^l .

$$\tilde{z}^l = z^l \otimes \boldsymbol{m} \tag{3}$$

where \otimes represents the element-wise multiplication. The modified activation \hat{z}^l is forward-propagated to the next layer.

3.2 Model Training

EWAS module should be adversarially trained with the backbone network. We can add multiple EWAS modules to a CNN model, but we empirically find that adding one module shows the best robustness. We conjecture the rational behind is that fine-grained modification effectively identifies the error or negative elements. As soon as the negative elements are adjusted accordingly, more EWAS modules are not helpful. However, the position of EWAS is critical for the robustness and we evaluate this in Section 4.2. Following the min-max optimization in Eq. (1), the EWAS-modified optimization problem can be written as:

$$\min_{\theta} E_{(x,y)\sim D}[\max_{\delta}(L(y, F(x+\delta, \theta)) + \lambda \cdot L_{\text{EWAS}}(y, \hat{s}))] \quad (4)$$

where $\hat{s} = \text{ALC}(f^l(x+\delta), \theta^{\text{ALC}})$, and f^l indicates the output of layer l. λ is a trade-off coefficient to balance the contribu-



Figure 2: Three steps of EWAS: 1) Flatten z^l and input it into ALC, and the output score of ALC \hat{s} calculate EWAS loss. 2) Class Related Scaling (CRS): m from ALC's weight element-wise multiply with z^l to scaling the z^l to get \tilde{z}^l . 3) Forward \tilde{z}^l into the model's next layer.

tion of ALC loss. L_{EWAS} here is the same loss function as the maximization problem in Eq. (1), which for AT is:

$$L_{\rm EWAS} = L_{\rm CE}^{\rm ALC}(\hat{p}(x+\delta), y)$$
(5)

EWAS can combine with diverse adversarial training methods such as TRADES [Zhang *et al.*, 2019], MART ([Wang *et al.*, 2020]), and the EWAS loss function needs to be modified accordingly. More details of different loss functions can be seen in Appendix B. The training algorithm is given in Appendix A.

4 Experiments

In this section, we extensively evaluate the effectiveness of EWAS in terms of adversarial robustness in comparison with the state of the arts [Bai et al., 2021] [Yan et al., 2021]. We use WideResNet-32-10 (we call it WideResNet), WideResNet-28-10 and ResNet-18 as in CAS and CIFS, and train models using CIFAR10 [Krizhevsky et al., 2009] and SVHN [Netzer et al., 2011] datasets. We empirically determine the best layer to add the EWAS module, i.e., the 15th layer for ResNet-18, the 19th layer of WideResNet-28-10 and 25th layer for WideResNet, respectively. AT [Madry et al., 2018] and its variants MART [Wang et al., 2020] and TRADES [Zhang et al., 2019] are used to train the models with EWAS-modified models, and three white-box attack methods are considered, FGSM [Szegedy et al., 2014], PGD-20 [Madry et al., 2018], C&W [Carlini and Wagner, 2017]. All attacks are perturbed by l_{∞} -norm with bound $\epsilon = 8/255$ and step size $\epsilon/4$. Note that to train the model with MART and TRADES, we need to modify the loss function accordingly. Models are trained for 120 epochs under AT, and the setting for other AT variants can be seen in Appendix D. We also visualize how the EWAS module affects the intermediate activation in Appendix F and investigate how λ effect attack evaluation results in Appendix G.

4.1 Robustness Analysis and Evaluation

In this section, we first evaluate EWAS against CAS and CIFS, which are closest to our work.

Robustness Evaluation

 λ in Eq. (4) is a critical parameter for EWAS module training, and the two datasets have different values, 0.01 for CIFAR10 and 0.05 for SVHN. Later, in the ablation study, we further evaluate the impact of λ . The adversarial accuracy of the last epoch is reported for each model.

ResNet-18	Natural	FGSM	PGD-20	C&W
AT	84.47	61.09	44.33	44.70
AT+CAS	85.89	61.17	50.55	52.56
AT+CIFS	82.70	58.10	49.49	50.24
AT+EWAS	84.73	65.78	64.84	82.35
TRADES	79.57	62.26	52.29	49.18
TRADES+CAS	83.05	63.81	56.63	60.03
TRADES+EWAS	80.35	61.85	61.29	74.92
MART	78.86	61.87	51.61	46.97
MART+CAS	86.40	62.61	54.33	61.49
MART+EWAS	81.80	65.31	64.01	79.67
WideResNet-28-10	Natural	FGSM	PGD-20	C&W
AT	87.29	58.50	49.17	48.68
AT+CAS	88.05	57.94	49.03	49.97
AT+CIFS	85.56	61.34	53.74	53.20
AT+EWAS	85.29	62.23	55.66	67.07
WideResNet	Natural	FGSM	PGD-20	C&W
AT	86.65	63.71	47.06	45.75
AT+EWAS	87.12	64.05	59.90	73.01
TRADES	84.16	65.34	52.92	51.61
TRADES+EWAS	83.96	64.50	62.39	74.88
MART	84.39	65.10	50.39	48.77
MART+EWAS	80.84	63.19	65.40	76.72

Table 1: Robustness (accuracy (%) on various white-box attacks) comparison of defense methods on CIFAR10. The best results are marked with an underline.

ResNet-18	Natural	FGSM	PGD-20	C&W
AT	93.72	65.87	50.35	47.89
AT+CAS	94.08	65.24	48.47	46.15
AT+CIFS	93.94	66.24	52.02	50.13
AT+EWAS	92.18	71.57	59.01	69.67

Table 2: Experimental results for SVHN.

Table 1 shows the experimental results for CIFAR10. As see from Table 1, EWAS greatly improves the robustness of models, especially the robustness against PGD and C&W attacks. The robust accuracy of ResNet-18 against C&W increases by 37.65% under AT, and such huge improvement makes its robust accuracy comparable to its natural accuracy, where the difference is only 2.38%. Also for PGD attack, EWAS significantly improves the adversarial accuracy by up to 20.51%. Although MART and TRADES can improve the robustness, the vanilla AT achieves the best robustness for ResNet-18 under CIFAR10. For WideResNet-28-10, EWAS outperforms CAS and CIFS in terms of robust accuracy. For WideResNet, MART and TRADES demonstrate better

performance than the vanilla AT, where we obtain the best robust accuracy under MART. Table 2 summarizes the results for SVHN, where EWAS performs superiority over CAS and CIFS in terms of the adversarial accuracy, and the improvement against C&W is up to 19.54%.

ResNet-18	Vanilla	CAS	EWAS
Robust Accuracy	39.35	65.31	63.22

Table 3: Robustness accuracy against AutoAttack on CIFAR10. CIFS does not report this.

We also evaluate the robust accuracy against AutoAttack [Croce and Hein, 2020] as [Bai *et al.*, 2021], which is a parameter-free attacks framework consist of both white-box and black-box attack. We use the AutoAttack including one white-box attack (APGD-DLR [Croce and Hein, 2020]) and one black-box attack (Square Attack [Andriushchenko *et al.*, 2020]). As shown in Table 3, EWAS can improve the robustness of DNN but 2.19% lower than CAS.

Feature Analysis

We visualize the activation of the penultimate layer (the last convolutional layer) of ResNet-18 w.r.t the activation magnitude and frequency in Fig. 3, and the visualization details are shown in Appendix C. As observed from the figure, the 4 methods demonstrate significantly different results. AT, CAS and CIFS aim to make adversarial examples similar to natural examples, whereas EWAS presents more difference between natural examples and adversarial examples. EWAS tends to have different activation distributions for two types of examples. This may provide a new direction to improve CNNs' robustness. The visualization of WideResNet activation is shown in Appendix H.

4.2 Ablation Study

The Impact of λ

In this part, we evaluate the impact of λ in Eq. (4). We train EWAS-modified ResNet-18 with 6 different values $\lambda = [0.01, 0.05, 0.1, 0.5, 1, 2]$ under AT on CIFAR10 and SVHN. λ serves two roles in the model training: 1) it balances the contributions of the backbone classifier and the auxiliary classifier; 2) it controls the strength of element scaling. The results are reported in Table 4 and Table 5.

For CIFAR10, the natural and robust accuracies decrease with the increase of λ over different attacks. When λ (i.e. $\lambda = 2$) is large, the model training cannot be converged, thereby leading to low accuracy for both natural and adversarial accuracy. However, for SVHN, there is no winning λ for diverse attacks. For PGD and C&W, the best λ is 0.05, where $\lambda = 2$ is the best for FGSM. The best λ for CIFAR10 is the worst selection for SVHN. Therefore, we choose $\lambda = 0.01$ for CIFAR10, and $\lambda = 0.05$ for SVHN.

The Impact of EWAS position

In this part, we evaluate the effect of EWAS' position on models' robustness, where we insert the EWAS module to different layers. The natural and robust accuracies against PGD-20 at different positions are shown in Fig. 4, and more different



Figure 3: Comparison of activation magnitude and frequency between adversarial and natural samples on different defense methods. Natural samples are from CIFAR-10 "airplane" class.

λ	Natural	FGSM	PGD-20	C&W
0.01	84.73	65.78	64.84	82.35
0.05	84.79	63.54	58.58	72.64
0.1	84.67	62.09	53.77	60.83
0.5	83.96	61.34	48.73	52.59
1	83.61	61.77	47.45	49.3
2	10.00	10.00	10.00	10.00

Table 4: Robust comparison of different λ on CIFAR10 for ResNet-18. The accuracies(%) for natural and adversarial data are reported.

λ	Natural	FGSM	PGD-20	C&W
0.01	19.58	19.58	19.58	19.58
0.05	92.18	71.57	59.01	69.67
0.1	92.72	72.42	58.38	63.36
0.5	93.20	74.03	57.07	55.37
1	93.02	74.23	57.30	54.57
2	93.34	75.42	58.85	55.23

Table 5: Robust comparison of different λ on SVHN for ResNet-18.

layers robust evaluation shown in Appendix E. The experimental results show that the best position is the first conv layer of the last block within a model.

We think there are two reasons behind. Since the adversarial perturbation is gradually amplified along its forward propagation [Liao *et al.*, 2018], adding EWAS module to early layers cannot effectively discern perturbations. In addition, features in early layers are more class-agnostic, so the auxiliary classifier may not take effect in this case. Therefore, we empirically choose to insert the EWAS module after the 15th layer of ResNet-18, the 19th layer of WideResNet-28-10 and the 25th layer of WideResNet.

5 Conclusion

In this paper, we propose a new element-wise activation scaling (EWAS) method to improve CNNs' adversarial robust-



Figure 4: The impact of EWAS position on CIFAR10.

ness. EWAS is a form of activation robustification techniques which can conduct a more fine-grained activation scaling. EWAS is a simple but very effective method to improve CNNs' robustness. It can be easily added to existing CNN models and be trained with the backbone network using an auxiliary loss function. The experimental results demonstrate that EWAS outperforms other two latest activation robustificiation techniques in terms of adverserial accuracy.

References

- [Andriushchenko *et al.*, 2020] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *ECCV*, pages 484–501, 2020.
- [Bai et al., 2021] Yang Bai, Yuyuan Zeng, Yong Jiang, Shu-Tao Xia, Xingjun Ma, and Yisen Wang. Improving adver-

sarial robustness via channel-wise activation suppressing. In *ICLR*, 2021.

- [Carlini and Wagner, 2017] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In S&P, pages 39–57, 2017.
- [Chen et al., 2019] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *IJCAI*, pages 4658–4664, 2019.
- [Croce and Hein, 2020] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, volume 119, pages 2206–2216, 2020.
- [Dosovitskiy *et al.*, 2021] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
- [Goldblum *et al.*, 2020] Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *AAAI*, pages 3996–4003, 2020.
- [Gui et al., 2019] Shupeng Gui, Haotao Wang, Haichuan Yang, Chen Yu, Zhangyang Wang, and Ji Liu. Model compression with adversarial robustness: A unified optimization framework. In *NeurIPS*, pages 1283–1294, 2019.
- [Jandial et al., 2019] Surgan Jandial, Puneet Mangla, Sakshi Varshney, and Vineeth Balasubramanian. Advgan++: Harnessing latent layers for adversary generation. In ICCV Workshops, pages 2045–2048, 2019.
- [Kannan *et al.*, 2018] Harini Kannan, Alexey Kurakin, and Ian J. Goodfellow. Adversarial logit pairing. *CoRR*, abs/1803.06373, 2018.
- [Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [Krizhevsky et al., 2017] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84– 90, 2017.
- [Liao et al., 2018] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In CVPR, pages 1778–1787, 2018.
- [Liu and Hsieh, 2019] Xuanqing Liu and Cho-Jui Hsieh. Rob-gan: Generator, discriminator, and adversarial attacker. In *CVPR*, pages 11234–11243, 2019.
- [Madaan *et al.*, 2020] Divyam Madaan, Jinwoo Shin, and Sung Ju Hwang. Adversarial neural pruning with latent vulnerability suppression. In *ICML*, pages 6575–6585, 2020.
- [Madry *et al.*, 2018] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

- [Moosavi-Dezfooli *et al.*, 2016] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *CVPR*, 2016.
- [Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.
- [Papernot et al., 2016] Nicolas Papernot, Patrick D. Mc-Daniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In S&P, pages 582–597, 2016.
- [Su et al., 2019] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE TEVC*, 23(5):828–841, 2019.
- [Szegedy *et al.*, 2014] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.
- [Wang *et al.*, 2020] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020.
- [Wong *et al.*, 2020] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.
- [Xiao et al., 2018] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. In *IJCAI*, 2018.
- [Xie et al., 2019] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In CVPR, pages 501–509, 2019.
- [Yan et al., 2021] Hanshu Yan, Jingfeng Zhang, Gang Niu, Jiashi Feng, Vincent Y. F. Tan, and Masashi Sugiyama. CIFS: improving adversarial robustness of cnns via channel-wise importance-based feature selection. In *ICML*, 2021.
- [Ye et al., 2019] Shaokai Ye, Xue Lin, Kaidi Xu, Sijia Liu, Hao Cheng, Jan-Henrik Lambrechts, Huan Zhang, Aojun Zhou, Kaisheng Ma, and Yanzhi Wang. Adversarial robustness vs. model compression, or both? In *ICCV*, pages 111–120, 2019.
- [Zhang *et al.*, 2019] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.
- [Zhang et al., 2020] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan S. Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *ICML*, 2020.

A Algorithm of EWAS Training

Algorithm 1 Adversarial training with EWAS

Input: Dataset $S = (x_i, y_i)_{i=1}^n$, CNN $F(\theta)$ with EWAS module, training epoch T

Output: A robust CNN F

1: for t = 1, 2, ..., T do

2: for (x_i, y_i) in S do

- 3: Generate adversarial example using PGD by solving inner-max problem in Eq. (4)
- 4: $\hat{s} = \operatorname{ALC}(\hat{f}^l(x_i + \delta), \theta^{\operatorname{ALC}})$

5: Generate m by Eq. (2)

- 7: Give \tilde{z}^l to the next convolution, complete the forward-propagation and compute the overall loss
- 8: end for
- 9: Optimize all the parameter of model and EWAS by solving outer-min problem in Eq. (4) using gradient descent
- 10: end for

B Loss function of EWAS-modified Model

Here we show the loss functions of MART and TRADES, the two AT variants. As shown in Table 6, p denotes the prediction score of network F and \hat{p} denotes the prediction score of EWAS module.

C Details on Activation Visualizing

We show the activation frequency and average activation magnitude.

C.1 Activation Frequency

We respectively performed natural and adversarial training on ResNet-18 under CIFAR-10 data set for 120 epochs with the SGD optimizer (momentum 0.9 and weight decay 0.0002). During adversarial training, We use adversarial data generated by PGD-10 attack ($\epsilon = 8/255$, step size $\epsilon/4$, and random initialization).

We use the output of the last residual block which also is the input of the global average pooling operation as the frequency visualization layer. The activation unit is valid if its activation magnitude is larger than 1% of the maximum of all activation. For visualization, we select all samples of one class as the input samples, and the results are shown in descending order of channel frequencies of the natural samples.

C.2 Activation Average Magnitude

The training details follows those in activation frequency visualizing. We also use the output of the last residual block which also is the input of the global average pooling operation as the activation magnitude visualization layer. For a certain class, we calculate channel's max activation value for each samples and average it over all the same class samples. We also plot it in descending order of average magnitude of the natural samples.

D Experimental Setting Details

The training setting of CIFS and CAS follows those in [Bai *et al.*, 2021; Yan *et al.*, 2021], which are $\beta_{CIFS} = 2$ and $\beta_{CAS} = 2$.

D.1 Experimental Details on CIFAR10

We train models with 128 batch size using SGD optimizer (momentum 0.9 and weight decay 0.0002), and initial learning rate is 0.1. With different training methods, we set different training epoch and milestones with multiplicative factor of learning rate decay 0.1, as shown in Table 7. During AT, we set $\epsilon = 8/255$ and step size $\epsilon/4$ for PGD-10 to generate adversarial samples. For TRADES and MART, β is 6.

	epochs	milestones
AT	120	60, 90
TRADE	85	75
MART	90	60

Table 7: Training epochs and learning rate adjust milestones for CI-FAR10 data set.

D.2 Experimental Details on SVHN

For SVHN dataset, we train model with 128 batch size using SGD optimizer (momentum 0.9 and weight decay 0.0005), and initial learning rate is 0.01, with different t raining method, we set same training epoch 120 and divided by 10 at 75-th and 90-th epoch. For training stage, we set $\epsilon = 8/255$ and step size $\epsilon/4$ for PGD-10 to generate adversarial samples. For TRADES and MART, β is 6.

For SVHN evaluation, adversarial data are generated by FGSM, PGD-20 (20-steps PGD with random start), and C&W (L_{∞} version of C&W optimized by PGD-30), ϵ is 8/255 and step size $\epsilon/10$.

Method	Loss function
AT	$L_{CE}(p(x+\delta, heta),y)$
+EWAS	$+\lambda \cdot L_{CE}^{ALC}(\hat{p}(x+\delta),y)$
TRADES	$L_{CE}(p(x,\theta),y) + \beta \cdot L_{KL}(p(x,\theta),p(x+\delta,\theta))$
+EWAS	$+\lambda \cdot L_{CE}^{ALC}(\hat{p}(x), y) + \lambda \cdot \beta \cdot L_{KL}^{ALC}(\hat{p}(x), \hat{p}(x+\delta))$
MART	$L_{BCE}(p(x+\delta,\theta),y) + \beta \cdot L_{KL}(p(x,\theta),p(x+\delta,\theta)) \cdot (1-p_y(x,\theta))$
+EWAS	$+\lambda \cdot L_{BCE}^{ALC}(\hat{p}(x+\delta), y) + \lambda \cdot \beta \cdot L_{KL}^{ALC}(\hat{p}(x), \hat{p}(x+\delta)) \cdot (1-\hat{p}_y(x))$

Table 6: The loss function used for AT, TRADES, MART with EWAS module.



Figure 5: Comparison of activation average magnitude and frequency between adversarial and natural examples before and after EWAS scaling and the penultimate layer. Natural samples are from CIFAR-10 "airplane" class.

E The Impact of EWAS Position

Here, we further show the experimental results of EWAS position evaluation. The natural and robust accuracies of ResNet-18 and WideResNet with different EWAS position against FGSM, PGD-20, C&W (Table 8 and Table 9). We train the model with $\lambda = 0.01$, and the evaluation settings follow those in Appendix D.

Layer	Natural	FGSM	PGD-20	C&W
11	79.02	58.19	55.71	65.71
13	83.58	62.73	58.77	72.77
15	84.73	65.78	64.84	82.35
17	83.73	62.67	56.63	71.31

Table 8: Robustness comparison of the EWAS module after different layers of ResNet-18 on CIFAR10. We reported the robust accuracy (%) at the last epoch. The final selected layer is marked with underline.

Layer	Natural	FGSM	PGD-20	C&W
21	85.63	61.71	51.68	58.00
23	85.70	62.46	50.52	55.43
25	87.12	64.05	59.90	73.01
27	86.66	62.74	51.19	60.68
29	86.36	61.68	50.51	58.62
31	85.90	65.06	54.52	62.60

Table 9: Robustness comparison of the EWAS module after different layers of WideResNet on CIFAR10. The final selected layer is marked with underline.

F The Performance of EWAS

To visualize how the EWAS module affects the intermediate activation, we visualize the average activation magnitude and frequency of ResNet-18 before and after EWAS scaling on CIFAR10 under AT. As shown in Fig 5, after EWAS scaling, both the magnitude and frequency have dropped drastically. From the figure, we can see that before EWAS, the activation magnitude is high, and after EWAS the activation magnitude is suppressed. Along the forward propagation, the activation of the penultimate layer shows different distributions between natural examples and adversaries.

G Attack Impact of λ

λ	Natural	FGSM	PGD-20	C&W
0	84.73	86.09	85.33	84.60
0.01		65.78	64.84	82.35
0.1		63.29	56.22	60.91
0.5		62.71	47.55	47.90
1		62.71	46.99	46.95
2		62.71	46.78	46.91
3		62.71	46.72	46.87
5		62.71	46.69	46.87
10		62.71	46.66	46.79
Vanilla	84.47	61.09	44.33	44.70

Table 10: Robustness comparison of the different evaluation λ of ResNet-18 on CIFAR10. We reported the robust accuracy (%) at the last epoch. The training λ is marked with underline.

We set different $\lambda = [0, 0.01, 0.1, 0.5, 1, 2, 3, 5, 10]$ to control the attack degree on the EWAS, where the larger the λ , the stronger the attack effect on the EWAS module. In other words, as the λ increases, the attack will focus on the EWAS module until the EWAS module is compromised, which means the model can only rely on its own robustness.

Here, we report the natural and robust accuracies of EWAS-modified ResNet-18 and WideResNet against FGSM, PGD-20, C&W (Table 10 and Table 11). When the adversary only takes the backbone classification loss as the maximization goal ($\lambda = 0$), it is very likely that the attack will fail. As the attack focuses on the EWAS loss, the robustness of the model will gradually decrease, but its robustness is still higher

λ	Natural	FGSM	PGD-20	C&W
0	87.12	83.96	83.61	83.66
0.01		64.05	59.90	73.01
0.1		63.50	48.88	50.42
0.5		63.49	47.27	48.23
1		63.50	47.21	48.08
2		63.50	47.20	48.09
3		63.50	47.19	48.07
5		63.50	47.20	48.06
10		63.50	47.19	48.05
Vanilla	86.65	63.71	47.06	45.75

Table 11: Robustness comparison of the different evaluation λ of WideResNet on CIFAR10. The training λ is marked with underline.

than the vanilla. We can see that EWAS plays an important role in the robustness of the model.

H The Visualization of WideResNet

Here we visualize the activation magnitude and frequency of the penultimate layer of WideResNet, as shown in Fig 6. The results also show EWAS presents more difference between natural examples and adversarial examples.



Figure 6: Comparison of activation magnitude and frequency between adversarial and natural samples on different defense methods on WideResNet. Natural samples are from CIFAR-10 "airplane" class.