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

Zhi-Yuan Zhang[a,*], Hao Ren[b,*], Zhenli He[a], Wei Zhou[a] and Di Liu[c,**]

[a] *School of Software, Yunnan University, Kunming, 650500, China*

[b] *Department of Information, Medical Supplies Center of PLA General Hospital, Beijing, 100853, China*

[c] *Department of Computer Science, Norwegian University of Science and Technology, Trondheim, 7491, Norway*

ARTICLE INFO

Keywords:
Adversarial attack
Robustness
Convolutional Nerual Networks
Element-Wise Scaling

ABSTRACT

Recent works reveal that re-calibrating intermediate activation of adversarial examples can improve the adversarial robustness of CNN models. The state of the arts exploit this feature at the channel level to help CNN models defend adversarial attacks, where each intermediate activation is uniformly scaled by a factor. However, we conduct a more fine-grained analysis on intermediate activation and observe that adversarial examples only change a portion of elements within an activation. This observation motivates us to investigate a new method to re-calibrate intermediate activation of CNNs to improve robustness. Instead of uniformly scaling each activation, we individually adjust each element within an activation and thus propose Element-Wise Activation Scaling, dubbed EWAS, to improve CNNs' adversarial robustness. EWAS is a simple yet very effective method in enhancing robustness. Experimental results on ResNet-18 and WideResNet with CIFAR10 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. The code and trained models are available at `https://github.com/ieslab-ynu/EWAS`.

1. Introduction

Convolutional neural networks (CNNs) have demonstrated its superiority in various applications, especially for computer vision tasks, like classification, object detection, object tracking, and segmentation [1, 2, 3, 4, 5]. However, CNNs are found to be vulnerable to adversarial samples that are perturbed by unperceptive noises [6]. Attacks of adversarial samples significantly undermine CNN models' robustness and threaten the applicability of CNNs to some safety-critical and security-critical contexts, e.g. self-driving [7] and person identification. A plenty of efforts have been made to understand adversarial attacks and to improve CNNs' adversarial robustness [8]. These efforts can be generally divided into two categories: *adversarial attacks* and *adversarial defense*. From the perspective of attack, various methods are proposed to generate diverse adversarial samples to attack CNN models so that we can understand the fundamentals of adversarial attacks and lay theoretical and empirical foundation to defend them [6, 9, 10, 11, 12, 13, 14, 15].

On the other hand, many works aim to defend adversarial attacks, thereby improving adversarial robustness of CNNs, i.e., a model's accuracy evaluated with adversarial samples. A number of defensive methods have been proposed, such as defensive distillation [16, 17], feature denoising [18, 19], Generative Adversarial Network (GAN)-based method [20], model compression [21, 22, 23], authentication defense [24], and adversarial training (AT) [10] and its variants [25, 26, 27]. Recently, some works investigate the difference between natural models and AT-trained counterparts in terms of intermediate activation and propose to adjust intermediate activation to improve adversarial robustness. Kanna et al. [28] proposed to make the logits (i.e., the classifier) of adversarial samples close to natural samples. The adversarial perturbations of input images are usually deemed as noises, and hence Xie et al. [18] suggested to denoise the distorted features using non-local means or other filters to improve robustness. Liao et al. [19] proposed to deploy high-level representations to guide the denoising procedure. Bai et al. [29] observed that adversarial examples wrongly activate '*negative*' features which lead to the final misclassification and thus proposed Channel-wise Activation Suppressing (CAS) strategy to suppress those '*negative*' features to improve a model's robustness. In parallel, Yan et al. [30] had similar observations and proposed a channel-wise activation method, namely CIFS, to enhance the adversarial robustness of CNN models. Besides suppressing the negative activation, they also promoted the positive activation to pursue higher accuracy.

These two methods apply to the channel/activation level, i.e., the whole channel or activation is suppressed or promoted by a uniform scaling. Such uniform activation scaling (suppression or promotion) methods do improve robustness as seen from [29, 30]. However, we conduct a fine-grained, element-wise analysis to compare the differences between

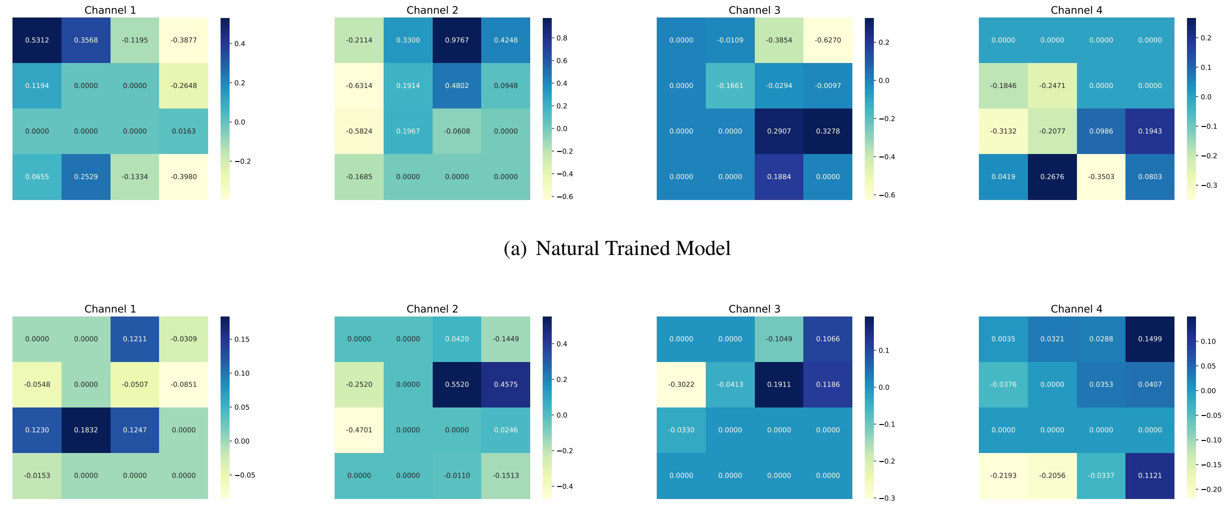
[⋆] This work was supported by NSFC under Grant 62262070 and 61902341 and funds from Yunnan province under Grant 202101AT070182 and 202201AT070156.

[*] Equal Contribution

[**] Corresponding author

✉ `di.liu@ntnu.no` (D. Liu)

ORCID(s):



(a) Natural Trained Model

(b) Adversarial Trained Model

Fig. 1: The penultimate layer's element differences of an adversarial sample comparing with its natural "cat" sample to ResNet-18 on CIFAR-10. ResNet-18 uses natural training (a), adversarial training (b). This figure shows the first four channels. The adversarial sample are generated using PGD-20 based on the corresponding natural sample.

natural samples and their corresponding adversarial samples of the same model and find that pixels/elements' differences within the same channel/activation are not uniform as shown in Fig. 1. Wherein, some elements exhibit no difference, while others show significant disparities. This implies that adversary only affects a portion of elements within a channel. Based on this, we can infer that uniformly scaling activation like [29, 30] is not the best method to manipulate the intermediate activation for robustness improvement, where such uniform scaling may lead to the information loss of the scaled activation, thus losing the opportunity to further improve the robustness.

Motivated by our fine-grained analysis shown in Fig 1, in this paper, we propose a novel 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 is not completely suppressed or promoted, but is re-calibrated in a fine-grained manner. Our key contributions in this paper are summarized as follows:

- We conduct an element-level analysis on the intermediate activation of the AT model and its natural counterpart and obtain a new observation regarding adversarial examples' intermediate activation. *Regardless of AT models or natural models, when a natural sample and its corresponding adversarial sample are fed to a same model, we observe that adversarial sample only changes a portion of elements of each activation*. This analysis is visualized in Fig. 1 and detailed in Sec 4.

- The new observation motivates us to design a new activation manipulation method to improve models' robustness. We propose the EWAS module, which can be easily added to existing CNN models. EWAS performs activation adjustment in an element-wise fashion to defend adversarial attack, and then the distorted activation is re-calibrated in a fine-grained manner. 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 [29][30]. 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%.

The remainder of this paper is organized as follows: Section 2 discusses related work. Section 3 presents preliminaries which are critical to understand our contribution. Section 4 presents the fine-grained analysis and EWAS. Section 5 shows experimental results and Section 6 concludes this paper.

2. Related Work

Since adversarial samples of CNNs were first found by Szegedy et al in [6], numerous methods are proposed to investigate the adversarial vulnerability of CNNs and to defend adversarial attacks. In this section, we review the related work from these two categories: *attack* and *defence*.

2.1. Adversarial Attack

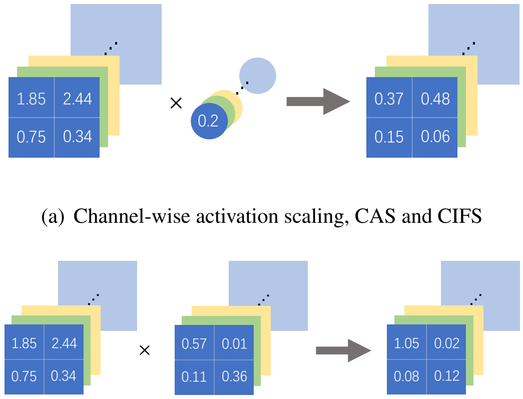
We classify adversarial attack into two categories, *black-box attacks* and *white-box attacks*. A black-box attack knows only the inputs and corresponding outputs from a model but the model's structure is unknown to attackers. The Pixel Attack proposed in [12] uses differential evolution algorithms that modify only one pixel of the image to misclassify the model. Jandial et al. [13] suggested using GAN to generate adversarial noise to implement black-box adversarial attack with high attack success rate. Papernot et al. [31] used a local substitute to craft adversarial examples so that the target CNN misclassifies its input. Chen et al. [32] proposed zeroth order optimization (ZOO) based attacks to directly estimate the gradients of the target DNN for generating adversarial examples without substitute models to avoid the loss in attack transferability. In general, black-box attacks are more difficult to successfully implement but practical in real world scenario.

In contrast, white-box attack is performed when the attacker knows all details of the target model, including parameters, gradients, structures, and data, so it is more challenging to defend against white-box attacks. Knowing all details of the target model is impossible in practice, but white-box attacks can facilitate the understanding of how adversarial examples realize attack on CNNs. Szegedy et al. [6] proposed a fast gradient sign method (FGSM) to generate adversarial examples. Madry et al. [10] proposed an iterative FGSM algorithm combined with random initialization to attack CNN models. Carlini et al. [9] designed a novel loss function to measure the difference between inputs and outputs to generate adversarial samples. To overcome the improper tuning of hyperparameters, Croce et al. [15] proposed a parameter-free, computationally affordable, and user-independent ensemble of attacks. Jandial et al. [14] also used GAN to generate adversarial examples where they use the feature map as the input of the generator. Wang et al. [33] observed existing transferable adversarial attacks ignore the intrinsic features of objects in images, so they proposed Feature Importance-aware Attack (FIA) that enhances the transferability of adversarial examples by disrupting the critically object-aware features which play a pivotal role in the predictive decision of different models. In this paper, we use the state-of-the-art attacks, Auto-Attack[15], FGSM [6], PGD [10], C&W [9], to evaluate the robustness of models.

2.2. Adversarial Defense

Besides various attacking methods, many efforts are made towards improving a model's defensive ability against adversarial attacks, i.e. , the model's robustness [10, 22, 25, 34, 26, 17, 35, 36, 20, 18, 28, 37, 21, 38, 39, 40]. Among them, adversarial training (AT) is widely used, because it can greatly improve robustness without modifying the model's structure. AT and its variants can be deemed as a data augmentation technique, where they generate adversarial samples during the training procedure and use adversaries to train the model. Resource-constrained devices are sensitive to models' size, so Ye et al. [22] proposed a framework to combine AT and weight pruning so that AT can improve the robustness while reducing the models' complexity. In addition of improving the robustness of CNN models, AT is also used for other purpose. Recently, Liu et al. [20] proposed to add adversarial samples into GAN training to improve the convergence speed and output quality (generated images). Due to the superiority of AT, it has been used as the de-facto training method for various adversarial methods. We also use AT and its variants to train EWAS-enabled models, and more details about AT is presented in Section 3.

Besides AT, diverse defense methods were proposed. Papernot el al. [16] proposed to use knowledge distillation to improve the robustness against FGSM attacks. Goldblum et al. [17] introduced Adversarially Robust Distillation (ARD) to transfer the superior robustness of large networks to the student model. Zi et al. [39] proposed a novel adversarial robustness distillation method, called Robust Soft Label Adversarial Distillation (RSLAD), to train a small robust student model.



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

(b) Element-wise activation scaling, EWAS

Fig. 2: Channel-wise scaling vs element-wise scaling. Element-wise scaling conducts a more fine-grained scaling to the intermediate activation.

Robust Activation Manipulation: The rationale behind adversarial attacks is *error amplification effect* [19], i.e., a small adversarial noise on an input image will be progressively amplified along a model's forward propagation, finally leading to misclassification. Thus, manipulating intermediate activations to eliminate such effects can enhance models' robustness. Some works strive to understand how activation features are modified by adversarial samples, thereby affecting the robustness of CNNs. Xie et al. [18] considered that features from adversarial examples are disturbed by noises, so they suggested using non-local means or other filters block to denoise the features. Madaan et al. [21] argued that the distortion in the latent feature space leads to the adversarial vulnerability, so they formally defined the vulnerability of the latent feature and proposed a vulnerability suppression loss to minimize the feature-level vulnerability during training. Some works observe the difference between adversarial examples and their natural counterparts from the lens of intermediate activation and strive to diminish such difference, e.g., adversarial logit pairing [28]. Since the robust local features can generalize well for unseen shape variation, Song el al. [37] developed a Random Block Shuffle (RBS) transformation to break up the global structure features on adversarial examples and learn the robust local features. Dhillon et al. [41] proposed stochastic activation pruning to prune those smaller magnitude activation on adversarially pre-trained model to improve the robustness. Mustafa el al. [42] proposed Prototype Conformity Loss to force the intermediate features of the same class to be clustered and different classes to be apart.

Recently, two concurrent works, CAS [29] and CIFS [30], adopt the robust activation scaling. Bai et al. [29] proposed Channel-wise Activation Suppressing (CAS) strategy to suppress redundant activations that are '*negatively*' activated by adversarial examples. Similarly, Yan et al. [30] observed that some channels, which are not pivotal to correct prediction but over-activated by adversarial examples, undermine the adversarial robustness. Thus, they proposed CIFS which identifies those channels and suppresses them to improve the robustness. These two methods both feature a channel-level scaling, i.e., the whole activation is uniformly scaled as shown in Fig. 2(a). However, from our fine-grained analysis, only some elements within an activation from an adversary differ from its natural counterpart. Then, uniform scaling may not be the optimal solution to manipulate intermediate activation. Thus, we conjecture that individually adjusting each element within an activation would help improve a model's robustness. This motivates our EWAS method. The idea behind EWAS is simple but effective as we can see from our extensive evaluation in Section 5 which justifies our conjecture.

3. Preliminaries

EWAS is a module to re-calibrate the intermediate activation of CNN models, so that the distorted elements caused by adversarial noise/perturbation can be corrected, thereby improving the robustness. AT is the most common technique to improve the model's robustness, where it trains models with adversarial data augmentation, i.e., adding adversarial examples to the training procedure. The EWAS-enhanced models also deploy AT. Therefore, in this section, we briefly review AT and its variants.

Adversarial Training: AT [10] is the most widely used method to improve CNNs' robustness. AT is a training method including data augmentation technique for adversarial defence, where it aims to solve the following min-max optimization problem:

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

where F represents a CNN model with weight parameters $\boldsymbol{\theta}$, 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 $\|\delta\|_p < \epsilon$. Here, similar to previous methods [30, 29], 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 minimize the loss by learning model weights $\boldsymbol{\theta}$ with generated adversarial examples.

Different adversarial attacks can be applied to AT, such as Projected Gradient Descent (PGD) [10] and fast gradient sign method (FGSM) [27]. Moreover, since the emergence of AT, diverse methods have been proposed to improve the effectiveness and efficiency of AT. Wong et al. [27] combined FGSM [6] with random initialization to make FGSM applicable to AT with lower cost. TRADES [25] is proposed to strike a balance between robustness and accuracy. Wang et al. [26] 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. FAT [34] is proposed to use early stop PGD to address the accuracy drop for natural examples. In our evaluation, we use AT, TRADES, and MART to train our models.

4. Element-Wise Activation Scaling

In this section, we first conduct an element-wise analysis to evaluate the activation difference between natural samples and adversarial counterparts and then present the details of EWAS and the training methods for the EWAS-enhanced models.

4.1. Activation Analysis

We investigate how activations are changed by adversarial samples in terms of activation elements. This analysis serves as a foundation for our EWAS module. In this analysis, we compare the activation difference for both normal model and AT model. This analysis uses a ResNet18 model and data from CIFAR10 dataset. We present a natural sample and its corresponding adversary to the model and observe the activation difference in terms of element.

Some results are shown in Fig. 1, where we select one sample from the 'cat' category and visualize the activation difference of the first 4 channels within the penultimate convolutional layer (the one just before the classifier). *It is worth noting that for other samples and other channels, we have the similar observation, but due to the space limitation we cannot show all of them here.*

The activation size of the penultimate layer of ResNet 18 is 4×4 and the difference is calculated by the adversarial activation minus the natural activation in an element-wise manner. Fig. 1(a) and 1(b) show the results of the natural model and AT model, respectively. From the results, we have the following observations:

- Within an activation, only a portion of activation elements from the adversary are different from its natural counterpart, i.e., adversarial perturbation does not affect all elements of an activation.
- For the elements which demonstrate difference, their disparities are not uniform, i.e., we cannot use a uniform scaling factor to adjust them.
- Both natural and AT models demonstrate the above-mentioned trend in terms of element difference, i.e., element-wise distortion and non-uniform distortion.

These observations imply that if we want to make the adversary similar to its natural counterpart in terms of activation elements, it is better to conduct an element-wise adjustment instead of uniformly scaling up or down the whole activation like [29] and [30]. This inspires to design EWAS. We proceed to the EWAS module and how to train it in the next section.

4.2. EWAS Module

Fig. 4 demonstrates the overview of EWAS, where EWAS is a plug-in module being added to 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. We conjecture the rationale behind is that fine-grained modification effectively identifies the distorted elements. As soon as the distorted elements are adjusted accurately and correctly, errors will not be further propagated afterwards and more EWAS modules are thus not helpful. Moreover, by doing this, we can also reduce the extra computation caused by EWAS. On the other hand, the position of EWAS is critical for the robustness, and we evaluate this in Section 5.4.

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 $\boldsymbol{m} \in \mathbb{R}^{C\times H\times W}$ to denote the scaling factor vector. As seen in [29][30], class-related activation modification is instrumental in improving the robustness. Hence, we also deploy an auxiliary classifier to derive the class-related features and to determine the element-wise scaling factor $\boldsymbol{m}$.

4.2.1. *Auxiliary Linear classifier (ALC)*

The core of EWAS is scaling factor $\boldsymbol{m}$. A good scaling factor $\boldsymbol{m}$ suppresses redundant and negative elements while retaining or promoting robust and positive elements. Inspired by CAS [29], we add an auxiliary linear classifier (ALC) to the original model and use ALC to derive $\boldsymbol{m}$. The overview of EWAS can be seen in Fig. 4, where Fig. 4(a) and 4(b) show the training procedure and inference procedure, respectively.

ALC is a simple but effective linear classification layer. ALC takes activation z^l and flats it as the input. It outputs classification scores of K classes, where K is the number of classes the dataset has. Let $\theta^{\text{ALC}} \in \mathbb{R}^{(C\cdot H\cdot W)\times K}$ denote the parameters of ALC. ALC parameters $\theta^{\text{ALC}} \in \mathbb{R}^{(C\cdot H\cdot W)\times K}$ are deployed to generate scaling mask $\boldsymbol{m}$, where $\theta^{\text{ALC}}_k \in \mathbb{R}^{C\cdot H\cdot W}$ represents the parameters related to class k and θ^{ALC}_k is reformatted to scaling factor $\boldsymbol{m} \in \mathbb{R}^{C\times H\times W}$. To facilitate the understanding of the reformat function, we deploy a simple example to visualize the procedure, which is shown in Fig. 3.

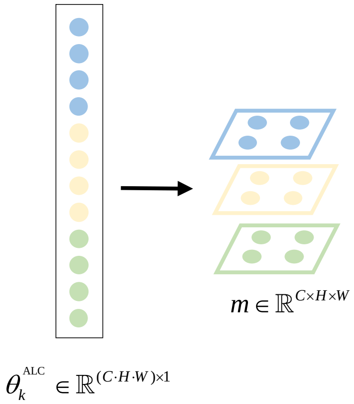

$m \in \mathbb{R}^{C\times H\times W}$
$\theta_k^{\text{ALC}} \in \mathbb{R}^{(C\cdot H\cdot W)\times 1}$

Fig. 3: A example of reformat() function. In this diagram, $C = 3, W = H = 2$.

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 $\boldsymbol{m}$ is formulated as follows:

$$\boldsymbol{m} = \begin{cases} \text{reformat}(\theta^{\text{ALC}}_y), & \text{(training stage)} \\ \text{reformat}(\theta^{\text{ALC}}_{\arg\max(\hat{s})}), & \text{(inference stage)} \end{cases} \tag{2}$$

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

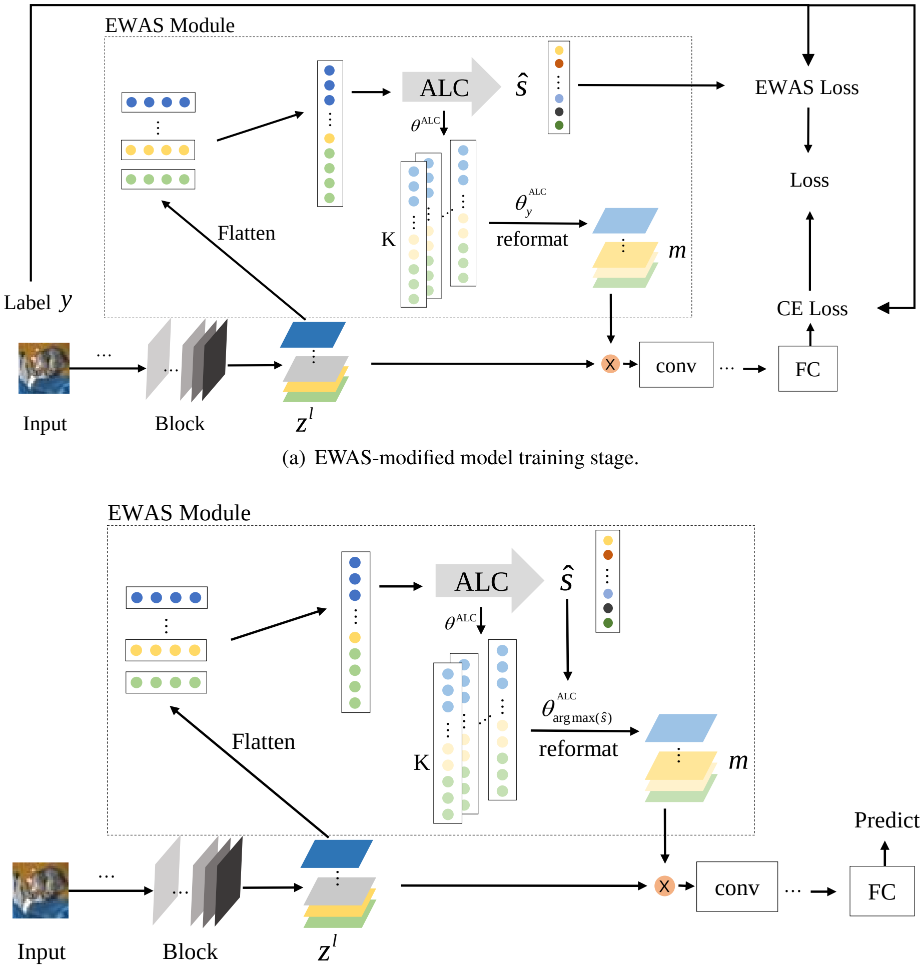
$$\hat{z}^l = z^l \otimes \boldsymbol{m} \tag{3}$$

where $\otimes$ represents the element-wise multiplication. The modified activation $\hat{z}^l$ is forward-propagated to the next layer. Fig 4 visualizes the procedure.

4.3. Model Training

EWAS module should be adversarially trained with its backbone network. Following the min-max optimization introduced in Eq. (1), the EWAS-modified optimization problem can be written as:

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



(a) EWAS-modified model training stage.

(b) EWAS-modified model inference stage.

Fig. 4: Three steps of EWAS: 1) Flatten z^l and input it into ALC, and the output score of ALC $\hat{s}$ calculates EWAS loss. 2) Class Related Scaling (CRS): $\boldsymbol{m}$ from ALC's weight element-wise multiplies with z^l to scaling the z^l to get $\tilde{z}^l$. 3) Forward $\tilde{z}^l$ into the model's next layer.

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

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

Note that similar to many other works which have the trade-off coefficient in their loss functions [30, 29], the coefficient λ of Eq. (5) is determined empirically in EWAS. Also, we observe that different datasets may have different empirically optimal λ.

Algorithm 1 shows the training process of EWAS-added models. EWAS can also combine with other AT methods such as TRADES [25], MART ([26]), but the EWAS loss function needs to be modified accordingly. More details of different loss functions can be seen in Table 1.

Algorithm 1 Adversarial training with EWAS

```
Require: Dataset $D = (x_i, y_i)_{i=1}^n$, CNN $F(\theta)$ with EWAS module, training epoch $T$
Ensure: A robust CNN $F$
for $t = 1, 2, ..., T$ do
    for $(x_i, y_i)$ in $D$ do
        Generate adversarial example using PGD by solving inner-max problem in Eq. (4)
        $\hat{s} = \text{ALC}(f^l(x_i+\delta), \theta^{\text{ALC}})$
        Generate $\boldsymbol{m}$ by Eq. (2)
        $\tilde{z}^l = z^l \otimes \boldsymbol{m}$
        Feed $\tilde{z}^l$ to the next layer, complete the forward-propagation and compute the overall loss
    end for
    Optimize all the parameter of model and EWAS by solving outer-min problem in Eq. (4) using gradient descent
end for
```

Table 1
The loss function used for AT, TRADES, MART with EWAS module.

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

5. Experiments

In this section, we extensively evaluate the effectiveness of EWAS in terms of adversarial robustness in comparison with the state of the arts [29][30]. We also report other robust activation manipulation methods, SAP (Stochastic Activation Pruning [41]), which prunes those smaller magnitude activation from pre-trained adversarial model, and PCL (Prototype Conformity Loss [42]), which separates the features of each class from others. We use WideResNet-32-10 (we call it WideResNet or WRN), WideResNet-28-10 (we call it WRN-28-10) and ResNet-18 as in CAS [29] and CIFS [30], and train models using CIFAR10 [43] and SVHN [44] datasets. AT [10] and its variants, MART [26] and TRADES [25], are used to train the models, and three white-box attack methods are considered, FGSM [6], PGD-20 [10], and C&W [9]. To train models using MART or TRADES, we use different loss functions accordingly as shown in Table 1. The training is conducted on one Nvidia RTX 2080ti.

We implement EWAS using PyTorch and have open-sourced our code to reproduce the experimental results[1]. In order to efficiently reproduce our results, we first present the experimental settings. Then, we discuss and analyze the experimental results using numerical comparisons and visualization. Ablation study evaluates the importance of the different parts in our EWAS, the impact of the newly introduced hyperparameter λ, and the position of EWAS module.

5.1. Experimental Settings

5.1.1. *Experimental Details on CIFAR10*

For CIFAR10, we train models with 128 batch size using SGD optimizer (momentum 0.9 and weight decay 2e−4), and the 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 2. During AT, we set $\epsilon = 8/255$ and step size $\epsilon/4$ for PGD-10 to generate adversarial samples. For CIFAR10 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 is $\epsilon/4$.

Table 2
Training epochs and learning rate adjust milestones for CIFAR10 data set.

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

5.1.2. *Experimental Details on SVHN*

For SVHN, we train models with 128 batch size using SGD optimizer (momentum 0.9 and weight decay 5e−4), and the initial learning rate is 0.01. With different training methods, we set the same training epoch 120 and divide the learning rate 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 SVHN evaluation, adversarial data are generated by FGSM, PGD-20, and C&W. ϵ is $8/255$ and step size is $\epsilon/10$.

For all models, the adversarial accuracy of the last epoch is reported for each model. The training settings of CIFS and CAS follow the original paper [29, 30].

5.2. Robustness Evaluation and Analysis

In this section, we present the evaluation mainly comparing EWAS with CAS and CIFS, which are closest to our work, but we also add two representative methods as baseline [41] and [42]. Additionally, to facilitate the understanding and analysis of the results, we visualize the experimental results using t-SNE [45].

5.2.1. *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.

Table 3 shows the experimental results for CIFAR10. As seen from Table 3, 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

[1] https://github.com/ieslab-ynu/EWAS

Table 3
Robustness comparison of defense methods on CIFAR10, where accuracy (%) on various white-box attacks is reported. The best results are marked with an underline.

ResNet-18	Natural	FGSM	PGD-20	C&W
SAP [41]	79.13	59.04	46.35	46.65
AT	84.47	61.09	44.33	44.70
PCL [42]	**88.15**	46.47	24.68	37.50
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**

WRN-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**

WRN	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**

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 under 3 attacks, but CAS achieves the best natural accuracy. For WideResNet, MART and TRADES demonstrate better performance than the vanilla AT, where we obtain the best robust accuracy under MART.

Compared with other methods, EWAS may sacrifice a model's nature accuracy for its robust accuracy. We conjecture the rational behind this is that there is a large difference between features of the adversarial and natural samples, and EWAS learns to scale features in a fine-grained fashion in order to improve its robustness. Nevertheless, such scaling may unnecessarily over-scale the features of natural examples and in turns affects its natural accuracy. We may consider to improve the natural accuracy in our future work. However, we can see that EWAS achieves a good balance between accuracy and robustness (as we can see in Table 10 and Table 11), where EWAS significantly improves the robust accuracy with a slight natural accuracy drop.

Table 4 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%.

Table 4
Robustness comparison of defense methods on SVHN, where accuracy (%) on various white-box attacks is reported. 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 5
Robustness accuracy against AutoAttack on CIFAR10.

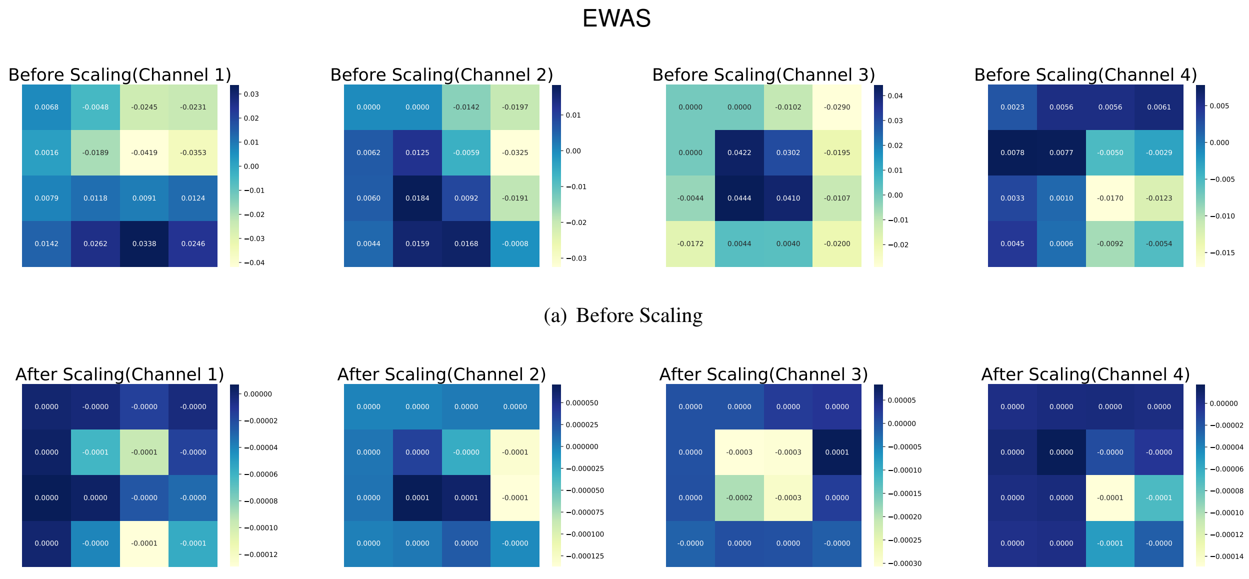
ResNet-18	Vanilla	CAS	CIFS	EWAS
Robust Accuracy	37.02	42.07	43.17	**55.60**

We also evaluate the robustness accuracy against AutoAttack [15] as [29]. AutoAttack is a parameter-free attacks framework, consisting of both white-box and black-box attacks. AutoAttack simultaneously applies multiple selected attack methods to all inputs and only reports one overall accuracy. In our experiment, we select one white-box attack (APGD [15]) and one black-box attack (Square Attack [46]) same as [29]. As shown in Table 5, EWAS can improve the robustness of ResNet-18 over the two reference approaches.

5.2.2. *Feature Analysis*

Similar to Fig. 1, we use a "cat" natural sample and its corresponding adversarial sample to compare the activation differences of the first 4 channels at the penultimate layer, where we show the results before scaling and after scaling. As shown in the Fig. 5 , after EWAS scaling, the activation differences of the model are reduced, which proves that EWAS can perform fine-grained activation scaling.

To investigate the differences between the compared methods, we use t-SNE 2D [45] to visualize the output of the last layer of the model, where we show the results for different methods with both natural and adversarial samples. As shown in Fig. 6(a), the features of the samples under natural training are clustered and show clear boundary between different classes, and moreover the clusters can be clearly distinguished with adversarial attacks. On the contrary, the features of the samples under adversarial training, shown in Fig. 6(b), are generally mixed and there are no obvious boundaries between classes. For CAS and CIFS shown in Fig. 6(c) and 6(d), they demonstrate the similar trends, where more clusters are formed and clusters from the same class have a larger variation comparing to the natural model. We see that CIFS and CAS demonstrate similar trends and we guess the channel scaling may be the reason. In contrast, due to the individual scaling of each activation element, the results of EWAS are more like the natural model's, and this may be the rational behind the robust improvement from EWAS.


EWAS

Before Scaling(Channel 1)

Before Scaling(Channel 2)

Before Scaling(Channel 3)

Before Scaling(Channel 4)

(a) Before Scaling

After Scaling(Channel 1)

After Scaling(Channel 2)

After Scaling(Channel 3)

After Scaling(Channel 4)

(b) After Scaling

Fig. 5: The penultimate layer features differences of an adversarial sample comparing with its natural "cat" sample to ResNet-18 on CIFAR-10 before EWAS scaling (a) and after EWAS scaling (b). This figure shows only the feature differences of the first four channels. The adversarial sample are generated using PGD-20 based on the corresponding natural samp

5.3. Channel Activation Analysis

CIFS and CAS conduct a channel-level analysis. In this part, we compare the EWAS-modified model with other channel-level scaling methods in channel level to demonstrate the differences for channel-wise scaling and element-wise scaling. We use the output of the last residual block of ResNet-18 to conduct the analysis, where we show the *activation frequency* and *activation average magnitude*. 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/average magnitude of the natural samples.

We visualize the average activation magnitude and frequency of ResNet-18 before and after EWAS scaling on CIFAR10 under AT. As shown in Fig 7, 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.

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. 8. As observed from the figure, the four methods demonstrate significantly different results. Comparing with CAS and CIFS, EWAS retains more activation features, as expected, with individual scaling for each activation unit, while those unchanging features continue to work. We also visualize the activation magnitude and frequency of the penultimate layer of WideResNet, as shown in Fig 9, The results also prove that EWAS can retain more features to obtain better performance.

Table 6
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	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

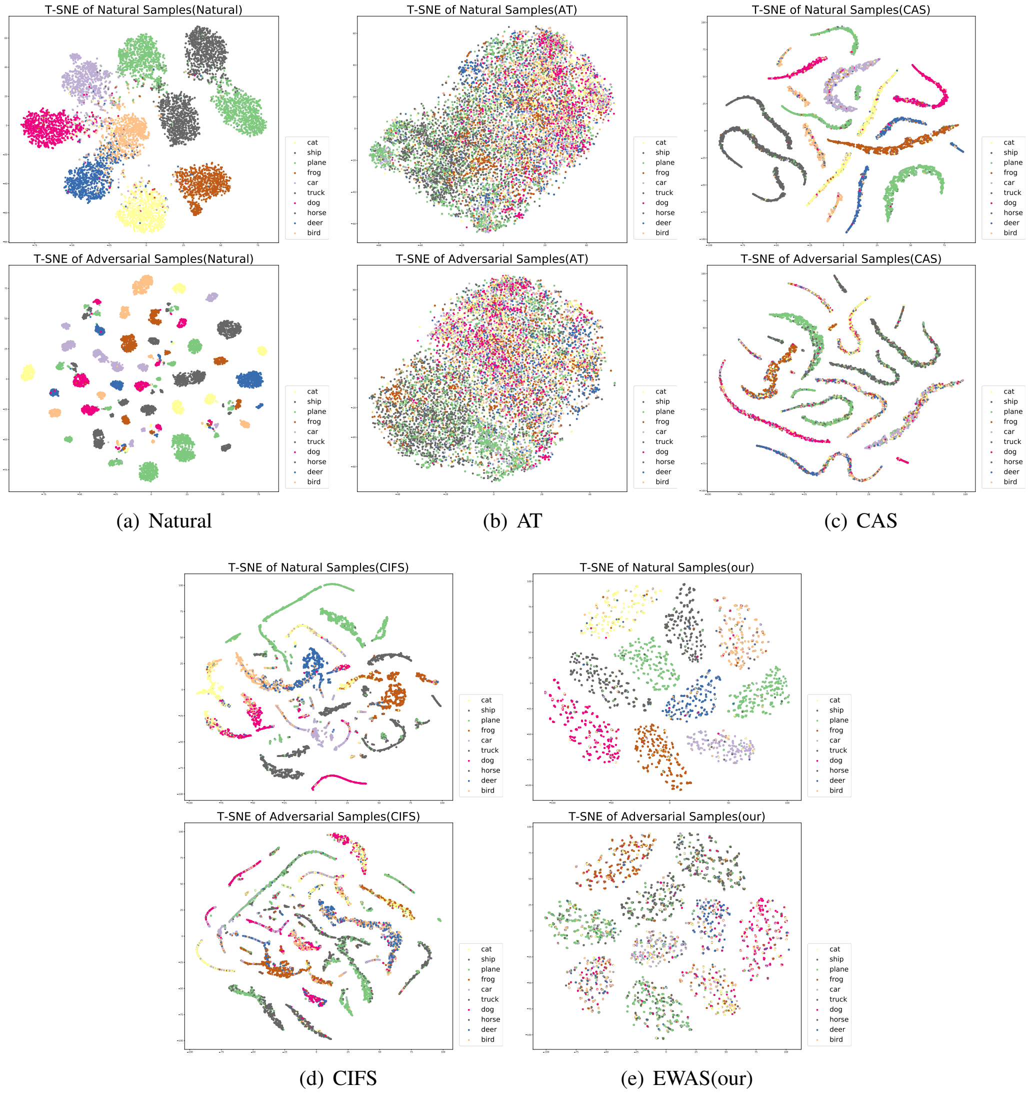
5.4. Ablation Study

5.4.1. *The Impact of λ*

In this part, we evaluate the impact of λ in Eq. (4), where we evaluate it in two ways, i.e., the impact on training and the impact on inference. The training impact indicates that how λ affects the robust accuracy when training a model, while the inference impact denotes how λ affects the generation of adversarial samples when evaluating the models.

Training impact:

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. We empirically 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. We observe that with larger λ, the auxiliary classifier part dominates the loss function, and this will degrade the model's performance, as we see from the experimental results shown in Table 6 and Table 7. Also, we observe different datasets have different optimal λ, so we empirically select proper λ for different datasets.



(a) Natural

(b) AT

(c) CAS

(d) CIFS

(e) EWAS(our)

Fig. 6: T-SNE of Natural Training (a), Adversarial Training (b), CAS (c), CIFS (d), EWAS (e) model for natural and adversarial samples on CIFAR-10. The backbone model is ResNet-18, and adversarial samples are generated by PGD-20.

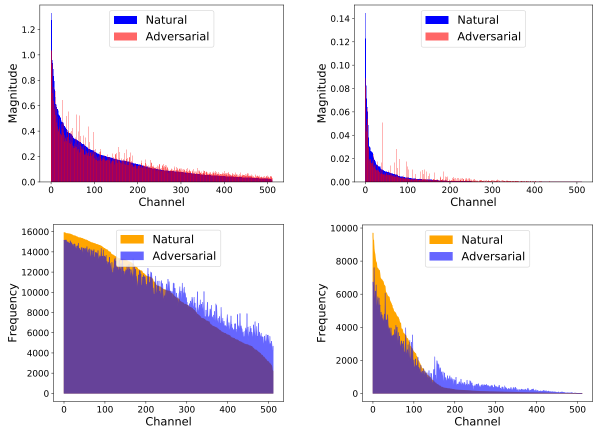
Table 7
Robustness comparison of different λ on SVHN for ResNet-18.

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

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.

Inference Impact: We evaluate the impact of different λ on robustness at inference phase. We use the ResNet18 with the best robust accuracy under AT and changing λ to generate adversarial examples. We set different $\lambda = [0, 0.01, 0.1, 0.5, 1, 2, 3, 5, 10]$. In this evaluation, λ is 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 gradually focuses 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 8 and Table 9). When the adversary



(a) Before EWAS

(b) After EWAS

Fig. 7: Comparison of activation average magnitude and frequency between adversarial and natural examples before and after EWAS scaling. Natural samples are from CIFAR-10 "airplane" class.

Table 8
Robustness comparison of the different λ of ResNet-18 on CIFAR10. We report the robust accuracy (%) at the last epoch. The training λ is marked with underline.

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 9
Robustness comparison of the different λ of WideResNet on CIFAR10. The training λ is marked with underline.

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 10
Robustness comparison of the EWAS module at different layers of ResNet-18 on CIFAR10. We report the robust accuracy (%) at the last epoch. The final selected layer is marked with underline.

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 11
Robustness comparison of the EWAS module at different layers of WideResNet on CIFAR10. 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

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

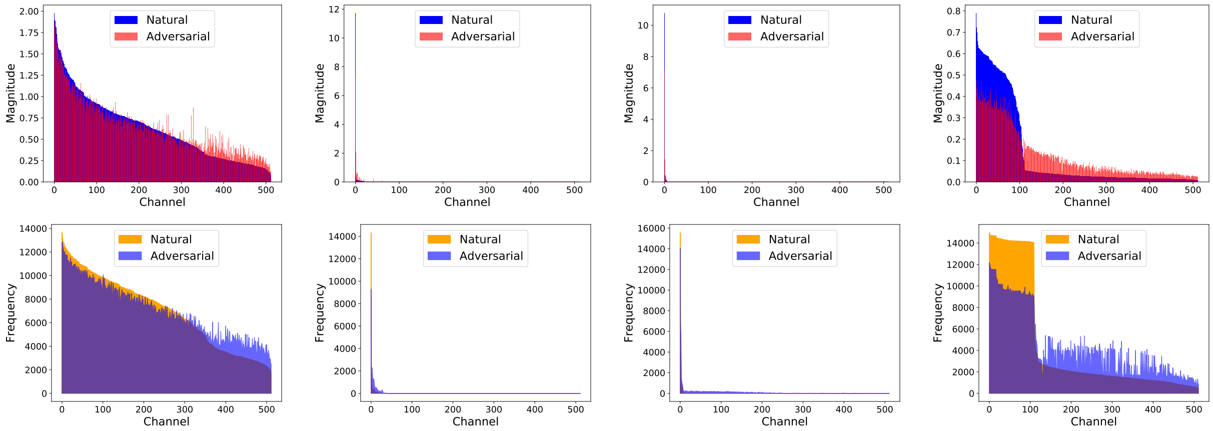
5.4.2. *The Impact of EWAS position*

In this part, we evaluate the effect of EWAS' position on models' robustness, where we insert the EWAS module at different layers. The natural and robust accuracies of EWAS-modified models against adversarial attack are shown in Table 10 and Table 11 for ResNet-18 and WideResNet, respectively. 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. On the one hand, the features at early layers are more class-agnostic. Since ALC is a class-aware classifier, adding ALC to early layers cannot effectively exploit this class-aware feature. On the other hand, the features of later layers are more class-specific, thus later layers are more suitable to add EWAS module. However, if we add the EWAS module to the last layer just before the classifier, it may sacrifice the generality, thereby affecting the accuracy as we can see from the experimental results. Therefore, by means of our empirical evaluation, we choose to insert the EWAS module at the 15th layer of ResNet-18, the 19th layer of WideResNet-28-10 and the 25th layer of WideResNet, which are the first convolutional layer of the last block of the models.

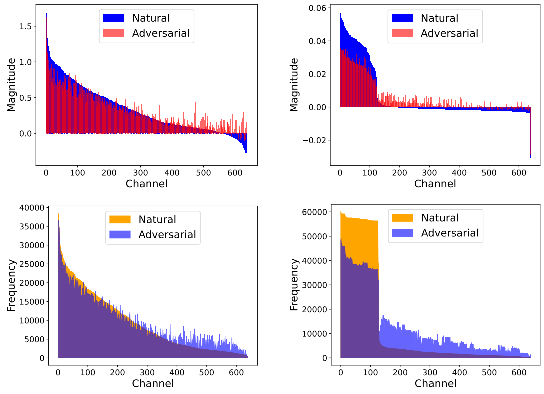
5.4.3. *Overhead Evaluation*

Since EWAS adds a new module to the backbone network, in this section, we evaluate EWAS in terms of training time, inference time and FLOPs. We use the same adversarial training method (AT) and train ResNet-18 on CIFAR-10 and record the training time on one Nvidia RTX 2080Ti. For the inference stage, since the increasing number of CNN models are implemented on edge devices, we use a



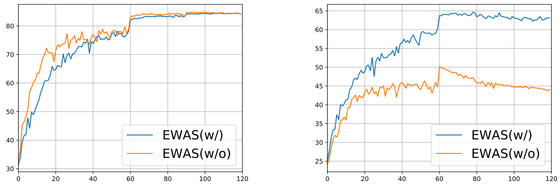
(a) AT (b) CAS (c) CIFS (d) EWAS(our)

Fig. 8: Comparison of activation magnitude and frequency between adversarial and natural samples on different defense methods. Natural samples are from CIFAR-10 "airplane" class.



(a) AT (b) EWAS

Fig. 9: 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.



(a) Natural Test Accuracy (b) PGD-20 Test Accuracy

Fig. 10: Comparison of loss convergence with and without EWAS module on CIFAR-10 use ResNet-18.

representative edge device Nvidia Jetson NANO to evaluate the inference time. We infer 10000 random inputs and take the average as its inference time. As shown in Table 12, the difference between the model with EWAS module and without EWAS module is negligible in terms of FLOPs and inference time and the training time is just increased by 7%. This result demonstrates the efficiency of EWAS.

Table 12
Total training time(on Nvidia RTX 2080ti), average inference time(on Nvidia Jetson NANO) and FLOPs with and without EWAS on ResNet-18.

	Training Time (h)	**Inference Time (ms)**	**FLOPs (M)**
EWAS(w/)	4.30	16.27	556.73
EWAS(w/o)	4.04	16.26	556.65

In addition, we evaluate the effect of EWAS on the loss convergence, where we, in Fig. 10, visualize the test accuracy and robust accuracy for each epoch using AT with ResNet-18 on CIFAR-10 to compare the loss convergence of the model with or without EWAS module. The results show that adding EWAS module does not affect the loss convergence.

6. Conclusion

In this paper, we conduct a more fine-grained study of the activation features of the model and obtain a new observation of adversarial examples' features, i.e., the adversary implements attack on CNNs by modifying a portion of elements within activation. The new observation motivates us to propose a new element-wise activation scaling (EWAS) method to improve CNNs' adversarial robustness. EWAS is a simple yet 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 adversarial accuracy.

This is the first work to explore the element-wise scaling to improve CNN models' robustness. However, the element-wise scaling may cause over-scaling problem, thereby affecting the natural accuracy. Therefore, we plan to address this issue in our future work. Moreover, we may further explore the application of element-wise scaling methods in other areas such as object detection, image segmentation, etc.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017.

[2] 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.

[3] Amir Hossein Barshooi and Abdollah Amirkhani. A novel data augmentation based on gabor filter and convolutional deep learning for improving the classification of COVID-19 chest x-ray images. *Biomed. Signal Process. Control.*, 72(Part):103326, 2022.

[4] Amir Ebrahimi Abdollah Amirkhani, Amir Hossein Barshooi. Enhancing the robustness of visual object tracking <i>via</i> style transfer. *Computers, Materials & Continua*, 70(1):981–997, 2022.

[5] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788. IEEE Computer Society, 2016.

[6] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[7] Yao Deng, Xi Zheng, Tianyi Zhang, Chen Chen, Guannan Lou, and Miryung Kim. An analysis of adversarial attacks and defenses on autonomous driving models. In *2020 IEEE international conference on pervasive computing and communications (PerCom)*, pages 1–10. IEEE, 2020.

[8] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

[9] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *S&P*, pages 39–57, 2017.

[10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[11] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

[12] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE TEVC*, 23(5):828–841, 2019.

[13] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. In *IJCAI*, 2018.

[14] Surgan Jandial, Puneet Mangla, Sakshi Varshney, and Vineeth Balasubramanian. Advgan++: Harnessing latent layers for adversary generation. In *ICCV Workshops*, pages 2045–2048, 2019.

[15] 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.

[16] Nicolas Papernot, Patrick D. McDaniel, 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.

[17] Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *AAAI*, pages 3996–4003, 2020.

[18] 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.

[19] 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.

[20] Xuanqing Liu and Cho-Jui Hsieh. Rob-gan: Generator, discriminator, and adversarial attacker. In *CVPR*, pages 11234–11243, 2019.

[21] Divyam Madaan, Jinwoo Shin, and Sung Ju Hwang. Adversarial neural pruning with latent vulnerability suppression. In *ICML*, pages 6575–6585, 2020.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020.

[27] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.

[28] Harini Kannan, Alexey Kurakin, and Ian J. Goodfellow. Adversarial logit pairing. *CoRR*, abs/1803.06373, 2018.

[29] Yang Bai, Yuyuan Zeng, Yong Jiang, Shu-Tao Xia, Xingjun Ma, and Yisen Wang. Improving adversarial robustness via channel-wise activation suppressing. In *ICLR*, 2021.

[30] 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.

[31] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *AsiaCCS*, pages 506–519. ACM, 2017.

[32] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pages 15–26. ACM, 2017.

[33] Zhibo Wang, Hengchang Guo, Zhifei Zhang, Wenxin Liu, Zhan Qin, and Kui Ren. Feature importance-aware transferable adversarial attacks. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 7619–7628. IEEE, 2021.

[34] 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.

[35] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*, pages 5286–5295. PMLR, 2018.

[36] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. Ensemble adversarial training: Attacks and defenses. In *ICLR*, 2018.

[37] Chuanbiao Song, Kun He, Jiadong Lin, Liwei Wang, and John E. Hopcroft. Robust local features for improving the generalization of adversarial training. In *ICLR*, 2020.

[38] Nanyang Ye, Qianxiao Li, Xiao-Yun Zhou, and Zhanxing Zhu. Amata: An annealing mechanism for adversarial training acceleration. In

AAAI, pages 10691–10699, 2021.

[39] Bojia Zi, Shihao Zhao, Xingjun Ma, and Yu-Gang Jiang. Revisiting adversarial robustness distillation: Robust soft labels make student better. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16443–16452, 2021.

[40] Yujing Jiang, Xingjun Ma, Sarah Monazam Erfani, and James Bailey. Dual head adversarial training. In *IJCNN*, pages 1–8. IEEE, 2021.

[41] Guneet S. Dhillon, Kamyar Azizzadenesheli, Zachary C. Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *ICLR*, 2018.

[42] Aamir Mustafa, Salman H. Khan, Munawar Hayat, Roland Goecke, Jianbing Shen, and Ling Shao. Adversarial defense by restricting the hidden space of deep neural networks. In *ICCV*, pages 3384–3393. IEEE, 2019.

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

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

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

[46] 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.

Zhi-Yuan Zhang received his BS degree from Yunnan University, Kunming, China, in 2019. He is currently pursuing his master degree at Yunnan University. His research interests include adversarial attacks and edge intelligence.

Hao Ren received her BEng degree from Northwestern Polytechnical University, Xi'an, China, in 2007 and MSc degree from National University of Defence and Technology, Changsha, China, in 2009. She is currently working as a researcher in Medical Supplies Center of PLA General Hospital. Her research interests include adversarial attack, network security, medical information and software engineering.

Zhenli He received his Ph.D. degree in Systems Analysis and Integration from the Yunnan University, Kunming, China, in 2015. He is a Lecturer with the School of Software of Yunnan University. His current research interests include edge computing, energy-efficient computing, heterogeneous computing, and machine learning.

Wei Zhou received his PhD degree from the Chinese Academy of Science. Now he is a full professor at School of Software, Yunnan University. His current research interests include distributed data intensive computing and bioinformatics.

Di Liu received his BEng and MEng degrees from Northwestern Polytechnical University, Xi'an, China, in 2007 and 2011, respectively, and the PhD degree from Leiden University, Leiden, The Netherlands, in 2017. He was an assitant professor at Yunnan University, China and a research fellow at Nanyang Technological University, Singapore. He is currently an associate professor with the Department of Computer Science, Norwegian University of Science and Technology, Norway. His research interests include the fields of edge systems, machine learning on embedded systems and cyber-physical systems.