IMPROVING ADVERSARIAL ROBUSTNESS VIA CHANNEL-WISE ACTIVATION SUPPRESSING

Anonymous authors
Paper under double-blind review

ABSTRACT

The study of adversarial examples and their activations have attracted significant attention for secure and robust learning with deep neural networks (DNNs). Different from existing works, in this paper, we highlight two new characteristics of adversarial examples from the channel-wise activation perspective: 1) the activation magnitudes of adversarial examples are higher than that of natural examples; and 2) the channels are activated more uniformly by adversarial examples than natural examples. We find that, while the state-of-the-art defense adversarial training has addressed the first issue of high activation magnitude via training on adversarial examples, the second issue of uniform activation remains. This motivates us to suppress redundant activations from being activated by adversarial perturbations during the adversarial training process, via a Channel-wise Activation Suppressing (CAS) strategy. We show that CAS can train a model that inherently suppresses adversarial activations, and can be easily applied to existing defense methods to further improve their robustness. Our work provides a simple but generic training strategy for robustifying the intermediate layer activations of DNNs.

1 INTRODUCTION

Deep neural networks (DNNs) have become standard models for solving real-world complex problems, such as image classification (He et al., 2016), speech recognition (Wang et al., 2017), and natural language processing (Devlin et al., 2019). DNNs can approximate extremely complex functions through a series of linear (*e.g.*, convolution) and non-linear (*e.g.*, ReLU activation) operations. Despite their superb learning capabilities, DNNs have been found to be vulnerable to adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2015). Small adversarial perturbations on the input can cause significant changes on DNN output, leading to completely failed predictions.

A body of works have been proposed as defense techniques against adversarial examples, such as input denoising (Liao et al., 2018; Bai et al., 2019), defensive distillation (Papernot et al., 2016), gradient regularization (Gu & Rigazio, 2014), model compression (Das et al., 2018) and adversarial training (Goodfellow et al., 2015; Madry et al., 2018). Among these methods, adversarial training is arguably one of the most effective approaches (Athalye et al., 2019). Adversarial training is a data augmentation technique that trains DNNs on adversarial rather than natural examples. In adversarial training, natural examples are augmented (or perturbed) with the worst-case perturbations found within a small L_p-norm ball around them. This augmentation has been shown to effectively smooth out the loss landscape around the natural examples, and force the network to focus more on the pixels that are most relevant to the class. Apart from these interpretations, it is still unclear, from the intermediate activation perspective, why small input perturbations can accumulate across intermediate layers to flip the output, and how adversarial training can help mitigate such accumulations. The study of intermediate layer activations has thus become crucial for developing more in-depth understanding and robust DNNs.

In this paper, we show that, if studied from a channel-wise perspective, strong connections between certain characteristics of intermediate activations and adversarial robustness can be established. Our channel-wise analysis is motivated by the fact that different convolution filters (or channels) learn different patterns, which when combined together, describe a specific type of object. We study the adversarial examples from a new perspective of channels in activations. Different from the existing activation works assuming each channel is of equal importance, we focus on the relationship between channels. Intuitively, different channels of an intermediate layer contribute differently to the class prediction, thus have different levels of vulnerabilities (or robustness) to adversarial perturbations. Given an intermediate DNN layer, we first apply global average pooling to obtain the

channel-wise activations, based on which, we show that the activation magnitudes of adversarial examples are higher than that of natural examples. This means that adversarial perturbations generally have the signal-boosting effect on the channels. We also find that the channels are activated more uniformly by adversarial examples than that by natural examples. In other words, some redundant (or low contributing) channels that are not activated by natural examples, yet are activated by adversarial examples. We show that adversarial training can effectively address the high magnitude problem, yet fails to address the uniform channel activation problem, that is, some redundant and low contributing channels are still activated. To a certain extent, this explains why adversarial training works but its performance is not satisfactory. In this work, we propose a new training strategy named Channel-wise Activation Suppressing (CAS), which adaptively learns (with an auxiliary classifier) the importance of different channels to class prediction, and leverages the learned channel importance to adjust the channels dynamically in the training phase. The robustness of existing state-of-the-art adversarial training methods can be consistently improved if applied with our CAS training strategy. Our key contributions are summarized as follows:

- We identify, from a channel-wise activation perspective, two connections between DNN activations and adversarial robustness: 1) the activations of adversarial examples are of higher magnitude than that of natural examples; and 2) the channels are activated more uniformly by adversarial examples than that by natural examples. Adversarial training only addresses the first issue of high activation magnitude, yet fails to address the second issue of uniform channel activation.
- We propose a novel training strategy to train robust DNN intermediate layers via Channel-wise Activation Suppressing (CAS). In the training phase, CAS suppresses redundant channels dynamically by reweighting the channels based on their contributions to the class prediction. CAS is a generic intermediate-layer robustification technique that can be applied to any DNNs along with existing defense methods.
- We empirically show that our CAS training strategy can consistently improve the robustness of current state-of-the-art adversarial training methods. It is generic, effective, and can be easily incorporated into many existing defense methods. We also provide a complete analysis on the benefit of channel-wise activation suppressing to adversarial robustness.

2 Related Work

Adversarial Defense. Many adversarial defense techniques have been proposed since the discovery of adversarial examples (Liao et al., 2018; Bai et al., 2019; Papernot et al., 2016; Das et al., 2018). Among them, many are found to have caused obfuscated gradients and can be circumvented by Back Pass Differentiable Approximation (BPDA), Expectation over Transformation (EOT) or Reparameterization (Athalye et al., 2019). Adversarial training (AT) has been demonstrated to be the most effective defense (Athalye et al., 2019), which solves the following min-max problem:

$$\min_{\theta} \max_{\boldsymbol{x}' \in \mathcal{B}_\epsilon(\boldsymbol{x})} \mathcal{L}(\mathcal{F}(\boldsymbol{x}', \theta), y), \tag{1}$$

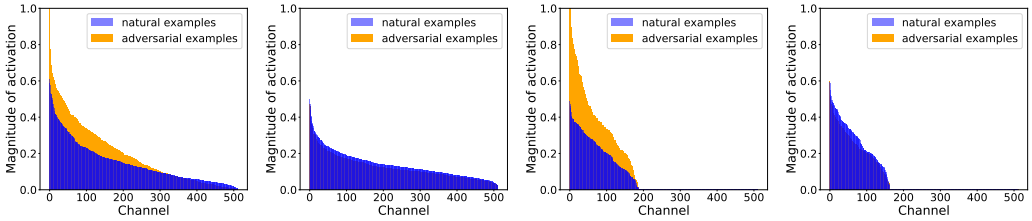
where, $\mathcal{F}$ is a DNN model with parameters θ, $\boldsymbol{x}$ is a natural example with class label y, $\boldsymbol{x}'$ is the adversarial example within the L_p-norm ball $\mathcal{B}_\epsilon(\boldsymbol{x}) = \{\boldsymbol{x}' : \| \boldsymbol{x}' - \boldsymbol{x} \|_p \leq \epsilon\}$ centered at $\boldsymbol{x}$, $\mathcal{F}(\boldsymbol{x}', \theta)$ is the output of the network, and $\mathcal{L}$ is the classification loss (*e.g.*, the cross-entropy loss). The inner maximization problem is dependent on the adversarial examples $\boldsymbol{x}'$ generated within the ϵ-ball, while the outer minimization problem optimizes model parameters under the worst-case perturbations found by the inner maximization. There are other variants of adversarial training with new objective functions or regularizations. For example, TRADES (Zhang et al., 2019) optimizes a trade-off objective of adversarial robustness and accuracy. MART (Wang et al., 2020b) applies a distinctive emphasis on misclassified versus correctly classified examples. However, apart from these improvements, it is still unclear how adversarial training can help produce state-of-the-art robustness from the activation perspective.

Activation Perspective of Adversarial Robustness. Some previous works have investigated the adversarial robustness from the architecture perspective, such as skip connection (Wu et al., 2020) and batch normalization (Galloway et al., 2019). As for intermediate activations, Zhang et al. (2018) certifies the robustness of neural networks with different activation functions. Xu et al. (2019) explores the influence of adversarial perturbations on activations from suppression, promotion and balance perspectives. Other works develop new activation operations with the manifold-interpolating data-dependent function in Wang et al. (2020a) and adaptive quantization techniques in Rakin et al.

(2018). While these works all directly modify the activation functions, there are also works focusing on the activation outputs. For instance, k-Winner-Takes-All (kWTA) (Xiao et al., 2020) takes the largest k feature values in each activation layer to enhance adversarial robustness. However, this has recently been shown not robust against adaptive attacks (Tramer et al., 2020). Stochastic Activation Pruning (SAP) (Dhillon et al., 2018) takes the randomness and the value of features into consideration. Each activation is chosen with a probability proportional to its absolute value. Adversarial Neural Pruning (ANP) (Madaan & Hwang, 2020) prunes out the features that are vulnerable to adversarial inputs using Bayesian method. Prototype Conformity Loss (PCL) (Mustafa et al., 2019) is used to cluster class-wise features and pushes class centers away from each other. Feature Denoising (FD) (Xie et al., 2019) adds denoising layers to the network for sample-wise denoising on feature maps. However, these methods are developed based on observations on the full output (*e.g.*, the entire feature or activation map that does not distinguish different channels) of DNN intermediate layers. In contrast, our CAS explores both channel importance and channel correlations, and the suppressing is done with the guidance of the labels.

3 Channel-wise Activation and Adversarial Robustness

In this part, we investigate two characteristics of DNN intermediate activations from a channel-wise perspective, and show two empirical connections between channel-wise activations and adversarial robustness. Specifically, we train ResNet18 (He et al., 2016) and VGG16 (Simonyan & Zisserman, 2014) on CIFAR-10 (Krizhevsky et al., 2009) using both standard training and adversarial training under typical settings. We then apply global average pooling to extract the channel-wise activations from the penultimate layer. We investigate the extracted channel-wise activations of both natural and adversarial examples from two perspectives: 1) the magnitude of the activations, and 2) the activation frequency of the channels.

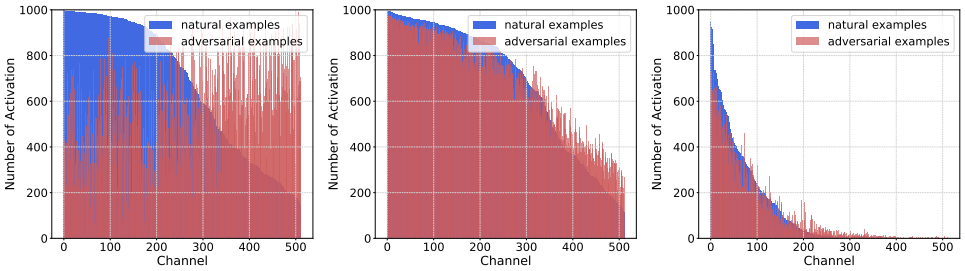


(a) ResNet18(STD) (b) ResNet18(ADV) (c) VGG16(STD) (d) VGG16(ADV)

Figure 1: The magnitudes (y-axis) of channel-wise activations at the penultimate layer (512 at x-axis) for both standard ('STD') and adversarially trained ('ADV') models. In each plot, the magnitudes are averaged and displayed separately for natural and adversarial test examples. The 512 channels are sorted in descending order of the magnitude.

Channel-wise Activation Magnitude. Figure 1 illustrates the averaged activation magnitudes for both natural test examples and the corresponding adversarial examples crafted by PGD-20 attack (Madry et al., 2018). For standard models (trained on natural examples), the activation magnitudes of adversarial examples are generally higher than that of natural examples as shown in Figure 1(a)/1(c). Adversarial perturbation exhibits a clear signal-boosting effect on the channels, which leads to the accumulation of adversarial distortions from the input to the out layer of the network. As shown in Figure 1(b)/1(d), adversarial training can effectively narrow the magnitude gap between natural and adversarial examples, interestingly by decreasing the activation magnitudes of adversarial examples. This is because adversarial training can restrict the Lipschiz constant of the model at deeper layers (*i.e.*, layers close to the output), which reduces the magnitude gaps caused by adversarial perturbations (Finlay et al., 2018; Sinha et al., 2019). Note that network architecture also influences activation magnitudes. Figure 1 show that magnitudes in VGG have much more zero values than those in ResNet, *i.e.*, VGG produces more sparse channels than ResNet.

Channel-wise Activation Frequency. Given a specific class, different convolution filters learn different patterns associated with the class. Similar to the robust *vs.* non-robust feature differentiation in adversarial training (Ilyas et al., 2019), the intermediate filters (or channels) can also be robust or non-robust. Intuitively, for natural examples in the same class, robust channels produce more generic patterns and should be activated more frequently, yet the non-robust ones should be activated less frequently. As such, non-robust channels can cause more variations to the next layer if activated by adversarial perturbations, increasing the vulnerability to adversarial examples. To investigate this, we visualize the activation frequency of the channel-wise activations in Figure 2. Here, we take one specific class (*e.g.*, class 0) of CIFAR-10 as an example. A channel is determined as activated



(a) ResNet18 (STD) (b) ResNet18 (ADV) (c) ResNet18 (our CAS)

Figure 2: The activation frequency (y-axis) of channel-wise activations at the penultimate layer (512 channels at x-axis) of ResNet18 trained using (a) standard training ('STD'), (b) adversarial training ('ADV'), and (c) our CAS-based adversarial training ('CAS'). The activation frequencies are counted separately for natural test examples and their PGD-20 adversarial examples. Channels are sorted in a descending order of activation frequency of natural examples.

if its activation value is larger than a threshold (*e.g.*, 1% of the maximum activation value over all 512 channels). We count the activation frequency for each channel by natural and adversarial examples separately on standard and adversarially trained ResNet18 models, and sort the channels in descending order according to the activation frequency by natural examples. As can be observed in Figure 2(a), adversarial examples activate the channels more uniformly, and they tend to frequently activate those that are rarely activated by natural examples (*e.g.*, the right part in Figure 2(a)). This observation is consistent across different classes. The low frequency channels are non-robust channels, and correspond to those redundant activations that are less important for the class prediction. It can also be observed that adversarial perturbation also inhibits those high frequency channels of natural examples. As shown in Figure 2(b), we find that, by training on adversarial examples, adversarial training can force the channels to be activated in a similar frequency by both natural and adversarial examples. However, there are still a certain proportion of the redundant channels (*e.g.*, channels #350 - #500) that are activated by adversarial examples. This motivates us to propose a Channel-wise Activation Suppressing (CAS) training strategy to avoid those redundant channels from being activated by adversarial examples. Figure 2(c) shows the effectiveness of our CAS strategy applied with adversarial training, that is, our CAS can suppress all channels, especially those low frequency ones on natural examples. More visualizations of channel-wise activation frequency for other adversarial training methods can be found in Appendix A.

4 Proposed Channel-wise Activation Suppressing

In this section, we introduce our Channel-wise Activation Suppressing (CAS) training strategy, which dynamically learns and incorporates the channel importance (to the class prediction) into the training phase to train a DNN model that inherently suppresses those less important channels.

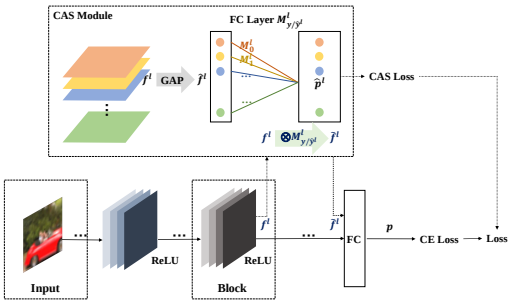


Figure 3: Framework of our proposed Channel-wise Activation Suppressing (CAS).

Overview. Figure 3 illustrates our CAS training strategy. The CAS module consists of a global average pooling operation (*i.e.*, GAP in the CAS module) to obtain the channel-wise activations, and an auxiliary classifier (*i.e.*, FC in the CAS module) to learn the channel importance. The learned channel importance is then multiplied back to the original activations for adjustment, and the adjusted activations are then passed into the next layer for model training. The entire network and the auxiliary classifier are trained simultaneously using a combination of the CAS loss and the CE loss. The CAS module can be attached to any intermediate layer of a DNN.

4.1 CAS MODULE

Denote the l-th activation layer output of network $\mathcal{F}$ as $\boldsymbol{f}^l \in \mathbb{R}^{H\times W\times K}$, where H, W, K represent the height, width, channel of the activation map, respectively. In CAS module, we first apply the GAP operation on the raw activation $\boldsymbol{f}^l$ to obtain the channel-wise activation $\hat{\boldsymbol{f}}^l \in \mathbb{R}^K$. Formally, for the k-th channel,

$$\hat{\boldsymbol{f}}_k^l = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\boldsymbol{f}_k^l(i,j). \tag{2}$$

The channel-wise activation $\hat{\boldsymbol{f}}^l$ is then passed into the auxiliary classifier to perform multi-class classification with a fully-connected (FC) layer. For C classes, the parameters of the auxiliary classifier can be written as $M^l = [M_1^l, M_2^l, .., M_C^l] \in \mathbb{R}^{K\times C}$, which can identify the importance of each channel to a specific class, and will be applied to reweight the original activations $\boldsymbol{f}^l$ in a channel-wise manner. In the training phase, the ground-truth label y is utilized as the index to determine the channel importance, *i.e.*, $M_y^l \in \mathbb{R}^K$. While for the test phase, since the ground-truth label is not available, we simply take the weight component $M_{\hat{y}^l}^l \in \mathbb{R}^K$ that is associated to the predicted class $\hat{y}^l$ as the channel importance (detailed analysis can be found in Section 5.1). The computed channel importance is then applied to reweight the original activation map $\boldsymbol{f}^l$ as follows:

$$\tilde{\boldsymbol{f}}^l = \begin{cases} \boldsymbol{f}^l \otimes M_y^l, & \text{(training phase)} \\ \boldsymbol{f}^l \otimes M_{\hat{y}^l}^l, & \text{(test phase)} \end{cases}, \tag{3}$$

where $\otimes$ represents the channel-wise multiplication. The adjusted $\tilde{\boldsymbol{f}}^l$ will be passed into the next layer via forward propagation. Note that, so far, neither the auxiliary nor the network is trained, just computing the channel importance and adjusting the activations in a channel-wise manner.

4.2 MODEL TRAINING

We can insert S Channel-wise Activation Suppressing (CAS) modules into S different intermediate layers of DNNs. The CAS modules can be considered as auxiliary components of the network, and can be trained using standard training or different types of adversarial training. Here, we take the original adversarial training (Madry et al., 2018) as an example, and define the loss functions to simultaneously train the network and our CAS modules. Each of our CAS modules has a FC layer. Taking one inserted CAS module after the l-th activation layer of network $\mathcal{F}$ for examples, the CAS loss function can then be defined as,

$$\mathcal{L}_{\text{CAS}}(\hat{\boldsymbol{p}}^l(\boldsymbol{x}', \theta, M), y) = -\sum_{c=1}^{C}\mathbb{1}\{c=y\}\cdot\log\hat{\boldsymbol{p}}_c^l(\boldsymbol{x}), \tag{4}$$

where $\hat{\boldsymbol{p}}^l = \text{softmax}(\hat{\boldsymbol{f}}^l M^l) \in \mathbb{R}^C$ is the prediction score of the classifier in CAS module. Similarly, it can also be extented to multiple CAS modules. The overall objective function for adversarial training with our CAS strategy is:

$$\mathcal{L}(\boldsymbol{x}', y; \theta, M) = \mathcal{L}_{\text{CE}}(\boldsymbol{p}(\boldsymbol{x}', \theta), y) + \frac{\beta}{S}\cdot\sum_{s=1}^{S}\mathcal{L}_{\text{CAS}}^s(\hat{\boldsymbol{p}}^s(\boldsymbol{x}', \theta, M), y) \tag{5}$$

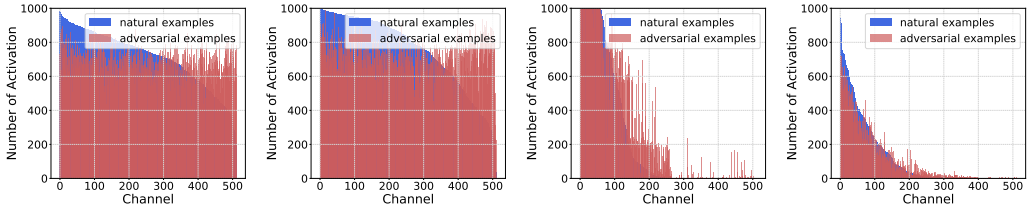
where β is a tunable parameter balancing the strength of CAS. Besides the original adversarial training (AT) (Madry et al., 2018), we can also combine CAS with other defense techniques such as TRADES (Zhang et al., 2019) and MART (Wang et al., 2020b). In Appendix B, we summarize the loss functions of the original AT, TRADES, MART, and their combined versions with our CAS. The complete training procedure of our CAS is described in Algorithm 1 in Appendix C.

5 EXPERIMENTS

5.1 EMPIRICAL UNDERSTANDING OF CAS

In this part, we first show the channel-suppressing effect and robustness of our CAS, then analyze the effectiveness of our CAS when applied at different layers of DNN. Other ablations or parameter analysis of CAS can be found in Appendix D and E.

Experimental Settings. We adversarially train ResNet18 for 200 epochs on CIFAR-10 using SGD with momentum 0.9, weight decay 2e-4, and initial learning rate 0.1 which is divided by 10 at 75-th and 90-th epoch. We use PGD-10 ($\epsilon = 8/255$ and step size $2/255$) with random start for training. The robustness (accuracy on adversarial examples) is evaluated under attacks: FGSM (Goodfellow et al., 2015), PGD-20 (Madry et al., 2018), and CW_∞ (Carlini & Wagner, 2017) optimized by PGD.



(a) kWTA (b) SAP (c) PCL (d) CAS

Figure 4: Comparisons of activation frequency distribution between adversarial and natural examples on different activation or feature oriented defense methods (kWTA, SAP, PCL and our CAS).

Table 1: Robustness (%) of ResNet18 trained by different defense (kWTA, SAP, PCL and our CAS) on CIFAR-10. Avg-PGD-100 denotes 100-step averaged PGD attack (Tramer et al., 2020).

Defense	Natural	FGSM	PGD-20	CW_∞	Avg-PGD-100	EOT
kWTA	76.48	59.56	**50.72**	46.84	16.72	–
SAP	79.13	59.04	46.35	46.65	–	19.98
PCL	**88.15**	46.47	24.68	37.50	–	–
CAS	86.79	**61.23**	48.88	**53.33**	**53.20**	**56.47**

Table 2: Effectiveness of channel suppressing in CAS (inserted at Block4) on CIFAR-10 with ResNet18.

Defense	Natural	FGSM	PGD-20	CW_∞
AT	84.27	60.46	46.50	48.97
AT+**CAS** (**without** channel suppressing)	83.42	59.81	44.20	46.27
AT+**CAS** (**with** channel suppressing)	**86.79**	**61.23**	**48.88**	**53.33**

Channel Suppressing Effect. We compare CAS with three activation- or feature-based defense methods: kWTA (Xiao et al., 2020), SAP (Dhillon et al., 2018) and PCL (Mustafa et al., 2019). Here, we train kWTA with 20% sparse largest values in each activation layer, SAP with typical random pruning and PCL with warm-up training by CE loss and then fine-tuning with the added PCL loss. Figure 4 shows the activation frequencies at the penultimate layer of ResNet18 trained by different methods. While kWTA, SAP and PCL demonstrate a certain level of channel suppressing, their effects are not as significant as our CAS training. kWTA and SAP hardly have the channel suppressing effect (*e.g.*, channel #350 - #500 in Figure 4(a) and channel #380 - #500 in Figure 4(b)), for the reason that they improve robustness mainly through the introduced randomness in activations and thus also could be easily attacked by some adaptive attacks (Tramer et al., 2020; Athalye et al., 2019). PCL still frequently activates many redundant channels (*e.g.*, channel #150 - #250 in Figure 4(c)). This is because PCL does not directly enforce channel suppression. Different to these methods, our CAS demonstrates the most effective channel suppression. It is also important to note that the comparison of frequency distributions across models is infeasible, yet the comparison of frequency distributions within one model is reasonable. For example, PCL has a quite skewed distribution on natural examples, yet it has the highest natural accuracy. Nevertheless, within the same model, the more similar the frequency distribution of adversarial examples is to that of natural examples, the better the adversarial robustness (*i.e*, the adversarial accuracy tends to approach the natural one).

From this view, our CAS effectively reduces the gap between natural and adversarial activation frequencies by suppressing the abundant channel activations, leading to superior robustness. The natural accuracies and robustness of these methods are reported in Table 1. Due to the randomness introduced in kWTA and SAP, they are not robust against average PGD (Avg-PGD) (Tramer et al., 2020) or Expectation Over Transformation (EOT) (Athalye et al., 2019) attacks. Our CAS training strategy does not rely on randomness, thus is robust even against Avg-PGD or EOT attacks.

We further verify that explicit Channel Suppressing (CS) is indeed essential to the improved robustness of CAS. Specifically, we remove CS defined in Eq. 3 from CAS, then retrain ResNet18 using

adversarial training with the CAS loss defined in Eq. 5. Table 2 shows that the robustness can not be improved without the explicit channel suppressing operation.

CAS at Different Layers. We insert the CAS module into different blocks of ResNet18, and show the different robustness improvements in Table 3. Intuitively, deeper layer activations are more correlated to the class prediction, thus should benefit more from our CAS training. Shallow layers, however, may suffer from inaccurate channel importance estimations. As demonstrated in Table 3, it is indeed the case: the largest improvement is obtained when applying CAS at Block4 (*e.g.*, after the ReLU output of Block4). The robustness can also be improved when inserting CAS into Block3 or both the Block3 and Block4 ('Block3+4'), though not as significant as that at Block4.

Table 3: Effectiveness of our CAS module at different blocks of ResNet18 on CIFAR-10.

Defense	Block	Natural	FGSM	PGD-20	CW_∞
AT+**CAS**	Block2	71.89	49.69	40.26	46.46
	Block3	83.05	59.20	47.84	48.19
	Block4	**86.79**	**61.23**	**48.88**	53.33
	Block3+4	83.77	58.32	48.27	**54.62**

Robustness of CAS Module. As our CAS module suppresses some channel-wise activations according to the label, it might raise some concerns on whether the CAS module itself is robust or how the misclassification in CAS module would affect the final results. First, as our CAS module is applied near the final output of the neural network, it shares very similar prediction label with the final one. Besides, we evaluate the robustness of CAS module against PGD-20 on the adaptive CAS loss in Table 4. The accuracy results show that our CAS module is robust itself, returning with a higher accuracy than the corresponding baseline model. More evaluations are in the Appendix F.5.

Table 4: Robustness of our CAS module of ResNet18 on CIFAR-10.

Defense	Adaptive Loss	Natural	FGSM	PGD-20	CW_∞
AT / **+CAS**	– / CAS	84.27/**84.95**	60.46/**61.40**	46.50/**47.99**	48.97/**57.79**
TRADES / **+CAS**	– / CAS	83.50/**83.84**	63.68/**64.31**	52.80/**54.11**	50.90/**64.01**
MART / **+CAS**	– / CAS	82.16/**84.89**	63.91/**65.14**	52.67/**54.48**	49.44/**66.92**

5.2 Robustness Evaluation

In this section, we evaluate our CAS on CIFAR-10 (Krizhevsky et al., 2009) and SVHN (Netzer et al., 2011) datasets with ResNet18 (He et al., 2016). We apply our CAS training strategy to several state-of-art adversarial training approaches: 1) AT (Adversarial Training) (Madry et al., 2018), 2) TRADES (Zhang et al., 2019), and 3) MART (Wang et al., 2020b). We follow the default setting as their original papers. More results on WideResNet-34-10 (Zagoruyko & Komodakis, 2016) and VGG16 (Simonyan & Zisserman, 2014) can be found in Appendix F.1 and F.2.

Experimental Settings. The training settings for CIFAR-10 are the same as Section 5.1. For SVHN, we adversarially train ResNet18 using SGD with momentum 0.9, weight decay 5e-4, initial learning rate 0.01 which is divided by 10 at 75-th and 90-th epoch, and training attack PGD-10 ($\epsilon = 8/255$ and step size $1/255$) with random start.

White-box Robustness. We evaluate the robustness of all defense models against three types of white-box attacks: FGSM, PGD-20 (step size $\epsilon/10$) and CW_∞ (optimized by PGD). To fairly compare our method with baselines, we use *adaptive white-box attack* for our CAS models, *i.e.*, the attacks are performed on the joint loss of CE and CAS. Here, we report the robustness of the models obtained at the last training epoch in Table 5. As shown in Table 5, our CAS can improve both the natural accuracy and the robustness of all baseline methods, resulting in noticeably better robustness. The improvement against CW_∞ attack is more significant than against FGSM or PGD-20 attacks. This is because the prediction margins are enlarged by our CAS training with channel suppression. As shown in Figure 9 Appendix E, the deep representations learned by CAS-trained models are more compact in each class, while are more separated across different classes. This makes margin-based attacks like CW_∞ much more difficult to succeed. Results with respect to the best model obtained throughout the entire training process and the learning curves are shown in Appendix F.3. It is worth mentioning that our CAS can consistently improve the robustness on the best models compared with the baseline methods, thus the improvement of our CAS is reliable, which is not caused by the ease of overfitting (Rice et al., 2020). Recently, several new attacks such as AutoAttack (Croce &

Hein, 2020b) and Fast Adaptive Boundary (FAB) attack (Croce & Hein, 2020a) have been proposed for more accurate evaluation of DNN robustness to adversarial examples. Our experiment results against the AutoAttack are provided in Appendix F.4, where our CAS consistently improves the robustness of baseline methods. This confirms that our CAS training strategy can bring real robustness rather than obfuscated gradients (Athalye et al., 2019). Furthermore, we evaluate our proposed CAS method under various attack perturbation constrains ϵ in Appendix F.5.

Table 5: White-box robustness (accuracy (%) on various white-box attacks) on CIFAR-10 and SVHN, based on the last checkpoint of ResNet18. '**+CAS**' indicates applying our CAS training strategy to existing defense methods. The best results are **boldfaced**.

Defense	SVHN				CIFAR-10			
	Natural	FGSM	PGD-20	CW_∞	Natural	FGSM	PGD-20	CW_∞
AT	89.62	65.09	42.55	50.96	84.27	60.46	46.50	48.97
AT+**CAS**	**90.39**	**67.51**	**51.98**	**53.53**	**86.79**	**61.23**	**48.88**	**53.33**
TRADES	91.16	69.85	50.90	50.85	83.50	63.68	52.80	50.90
TRADES+**CAS**	**91.69**	**70.97**	**55.26**	**60.10**	**85.83**	**65.21**	**55.99**	**67.17**
MART	91.16	67.31	48.72	50.52	82.16	**63.91**	52.67	49.44
MART+**CAS**	**93.05**	**70.30**	**51.57**	**53.38**	**86.95**	63.64	**54.37**	**63.16**

Black-box Robustness. We evaluate the black-box robustness of the defense models against both transfer- and query-based attack. For transfer attack, the adversarial examples are generated on CIFAR-10/SVHN test images by applying PGD-20 and CW_∞ attacks on a naturally trained ResNet50. For query-based attack, we adopt $\mathcal{N}$Attack (Li et al., 2019). Since $\mathcal{N}$Attack requires a lot of queries, we randomly sample 1,000 images from CIFAR-10/SVHN test set and limit the maximum query to 20,000. We test both black-box attacks on the models obtained at the last training epoch. The results are reported in Table 6. For both transfer and query attacks, our CAS can improve the robustness of all defense models by a considerable margin. Especially against the $\mathcal{N}$Attack, our CAS training strategy can improve AT, TRADES and MART by $\sim 30\% - 40\%$. One reason why our CAS is particularly more effective against $\mathcal{N}$Attack is that $\mathcal{N}$Attack utilizes a similar margin objective function as CW_∞, which can be effectively blocked by channel suppressing.

Table 6: Black-box robustness (accuracy (%) on various black-box attacks) of ResNet18 on SVHN and CIFAR-10. '**+CAS**' indicates applying our CAS training strategy to existing defense methods. The best results are **boldfaced**.

Defense	SVHN			CIFAR-10		
	PGD-20	CW_∞	$\mathcal{N}$Attack	PGD-20	CW_∞	$\mathcal{N}$Attack
AT	64.42	70.60	40.36	79.13	79.87	45.05
AT+**CAS**	**65.50**	**72.35**	**75.72**	**85.80**	**86.54**	**83.32**
TRADES	67.81	74.43	44.16	78.18	78.95	49.25
TRADES+**CAS**	**68.48**	**75.66**	**81.82**	**84.77**	**85.54**	**79.42**
MART	66.85	73.90	41.66	76.93	77.49	49.05
MART+**CAS**	**68.45**	**75.45**	**80.12**	**85.68**	**86.59**	**81.93**

6 Conclusion

In this paper, we investigated intermediate activations of deep neural networks (DNN) from a novel channel-wise perspective, in the context of adversarial robustness and adversarial training. We highlight two new characteristics of the channels of adversarial activations: 1) higher magnitude, and 2) more uniform activation frequency. We find that standard adversarial training improves robustness by addressing the first issue of higher magnitude, yet it fails to address the second issue of uniform activation frequency. To address the second issue, we proposed the Channel-wise Activation Suppressing (CAS), which dynamically learns the channel importance and leverages the learned channel importance to suppress the channel activations during the training process. When combined with adversarial training, we show that, CAS can train DNNs that inherently suppress redundant channels from being activated by adversarial examples. Our CAS is a simple but generic training strategy that can be easily plugged into different defense methods to further improve their robustness.

REFERENCES

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2019.

Yang Bai, Yan Feng, Yisen Wang, Tao Dai, Shu-Tao Xia, and Yong Jiang. Hilbert-based generative defense for adversarial examples. In *ICCV*, 2019.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *S&P*, 2017.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. *ICML*, 2020a.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *ICML*, 2020b.

Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E Kounavis, and Duen Horng Chau. Compression to the rescue: Defending from adversarial attacks across modalities. In *KDD*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

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

Chris Finlay, Adam Oberman, and Bilal Abbasi. Improved robustness to adversarial examples using lipschitz regularization of the loss. *arXiv preprint arXiv:1810.00953*, 2018.

Angus Galloway, Anna Golubeva, Thomas Tanay, Medhat Moussa, and Graham W Taylor. Batch normalization is a cause of adversarial vulnerability. *ICML Workshop*, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *Computer Science*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019.

Linxi Jiang, Xingjun Ma, Zejia Weng, James Bailey, and Yu-Gang Jiang. Imbalanced gradients: A new cause of overestimated adversarial robustness. *arXiv preprint arXiv:2006.13726*, 2020.

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

Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. *ICML*, 2019.

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*, 2018.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Divyam Madaan and Sung Ju Hwang. Adversarial neural pruning with latent vulnerability suppression. *ICML*, 2020.

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

Aamir Mustafa, Salman Khan, Munawar Hayat, Roland Goecke, Jianbing Shen, and Ling Shao. Adversarial defense by restricting the hidden space of deep neural networks. In *ICCV*, 2019.

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

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *S&P*, 2016.

Adnan Siraj Rakin, Jinfeng Yi, Boqing Gong, and Deliang Fan. Defend deep neural networks against adversarial examples via fixed anddynamic quantized activation functions. *arXiv preprint arXiv:1807.06714*, 2018.

Leslie Rice, Eric Wong, and J Zico Kolter. Overfitting in adversarially robust deep learning. *ICML*, 2020.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014.

Abhishek Sinha, Vineeth N Balasubramanian, Harshitha Machiraju, et al. Harnessing the vulnerability of latent layers in adversarially trained models. *AAAI*, 2019.

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

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

Bao Wang, Alex T Lin, Zuoqiang Shi, Wei Zhu, Penghang Yin, Andrea L Bertozzi, and Stanley J Osher. Adversarial defense via data dependent activation function and total variation minimization. *Inverse Problems and Imaging*, 2020a.

Yisen Wang, Xuejiao Deng, Songbai Pu, and Zhiheng Huang. Residual convolutional ctc networks for automatic speech recognition. *arXiv preprint arXiv:1702.07793*, 2017.

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

Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Rethinking the security of skip connections in resnet-like neural networks. In *ICLR*, 2020.

Chang Xiao, Peilin Zhong, and Changxi Zheng. Resisting adversarial attacks by k-winners-take-all. *ICLR*, 2020.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 501–509, 2019.

Kaidi Xu, Sijia Liu, Gaoyuan Zhang, Mengshu Sun, Pu Zhao, Quanfu Fan, Chuang Gan, and Xue Lin. Interpreting adversarial examples by activation promotion and suppression. *arXiv preprint arXiv:1904.02057*, 2019.

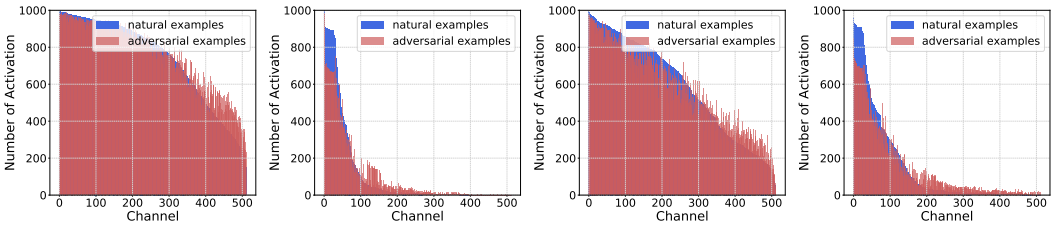
Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *BMVC*, 2016.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *ICML*, 2019.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *NeurIPS*, 2018.

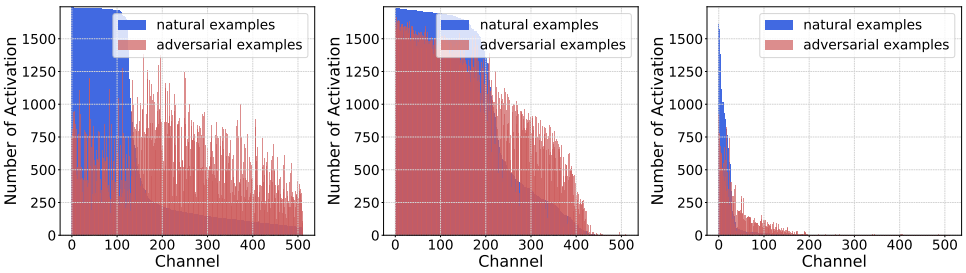
A CHANNEL-WISE ACTIVATION SUPPRESSING ON MORE DATA AND MORE DEFENSE MODELS

Here, we demonstrate the channel suppressing effect of our CAS training strategy on more defense models TRADES (Zhang et al., 2019) and MART (Wang et al., 2020b). We train ResNet18 (He et al., 2016) on CIFAR-10 (Krizhevsky et al., 2009). The CAS modules are attached to the final block of ResNet18. Figure 5 illustrates the channel activation frequencies of the original defense models and their improved versions with our CAS. As can be observed, our CAS can suppress the redundant channels consistently in both defense models. This restricts the channel-wise activations to be more class-correlated. More importantly, the channel activations are suppressed to similar frequency distributions between natural versus adversarial examples. Moreover, we compare the channel-wise activation of natural and adversarial examples on SVHN and ImageNet in Figure 6 and Figure 7. For SVHN, we choose examples of class 0 and construct the adversarial examples by PGD-20 with $\epsilon = 8/255$, we count the channel activations frequency of standard trained, adversarial trained and CAS trained models. As shown in Figure 6, neurons #200-#500 in Figure 6(a) and neurons #250-#400 in Figure 6(b), are frequently activated by adversarial examples while in Figure 6(c), our CAS suppresses those redundant activations. For ImageNet, as it is time consuming to perform adversarial training by ourselves, we adopt the standard pre-trained ResNet152 from torchvision.models [1] and adversarial trained ResNet152 from a Github repository[2]. We choose total 5,900 dog images in the validation set of ImageNet (class #151-#269) and generate the adversarial examples by PGD-30 (ϵ=16/255, α=1/255). We count the channel activations of the penultimate layer of ResNet152. It can be observed in Figure 7(a) that adversarial examples activate more uniformly than natural examples. While in Figure 7(b), adversarial training models align the activation distribution of natural examples and adversarial examples in some extent, however, the activation of adversarial examples still outstrip which indicates the non-robust features.



(a) TRADES (b) TRADES+**CAS** (c) MART (d) MART+**CAS**

Figure 5: The distributions of channel activation frequency of both natural and adversarial examples in different defense models (*e.g.*, TRADES and MART). The frequency distribution gap between natural and adversarial examples is effectively narrowed down by our CAS training, and the redundant channels (channel #200 - #512) are significantly suppressed by CAS.

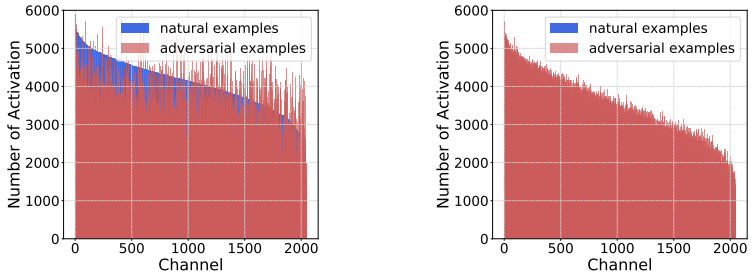


(a) SVHN ResNet18 (STD) (b) SVHN ResNet18 (ADV) (c) SVHN ResNet18 (our CAS)

Figure 6: The distributions of channel activation frequency of both natural and PGD-20 adversarial examples of ResNet18 on SVHN.

[1] https://pytorch.org/docs/stable/torchvision/models.html

[2] https://github.com/facebookresearch/ImageNet-Adversarial-Training



(a) ImageNet ResNet152 (STD) (b) ImageNet ResNet152 (ADV)

Figure 7: The distributions of channel activation frequency of both natural and PGD-30 adversarial examples of ResNet152 on ImageNet.

B Loss Functions for Training Different Defenses with our CAS

When combined with our CAS, the training loss is a combination of the original loss and our CAS loss. Table 7 defines the exact loss functions used for our CAS training of AT (Madry et al., 2018), TRADES (Zhang et al., 2019) and MART (Wang et al., 2020b). Here, we also assume attaching S number of CAS modules to the network. $\hat{\boldsymbol{p}}^s$ and M^s denote the prediction score and weights of the auxiliary classifier in the s-th CAS module, respectively.

Table 7: A summary of the loss functions used for standard adversarial training (AT), TRADES, MART, and their corresponding versions with CAS strategy ('**+CAS**').

Defense	Loss Function
AT **+CAS**	$\mathrm{CE}(\boldsymbol{p}(\boldsymbol{x}', \theta), y)$ $+\frac{\beta}{S}\sum_{s=1}^{S}\mathrm{CE}(\hat{\boldsymbol{p}}^s(\boldsymbol{x}', \theta, M^s), y)$
TRADES **+CAS**	$\mathrm{CE}(\boldsymbol{p}(\boldsymbol{x}, \theta), y) + \lambda \cdot \mathrm{KL}(\boldsymbol{p}(\boldsymbol{x}, \theta)\|\|\boldsymbol{p}(\boldsymbol{x}', \theta))$ $+\frac{\beta}{S}\sum_{s=1}^{S}\mathrm{CE}(\hat{\boldsymbol{p}}^s(\boldsymbol{x}, \theta, M^s), y) + \beta \cdot (\frac{\lambda}{S} \cdot \sum_{s=1}^{S}\mathrm{KL}(\hat{\boldsymbol{p}}^s(\boldsymbol{x}, \theta, M^s)\|\|\hat{\boldsymbol{p}}^s(\boldsymbol{x}', \theta, M^s)))$
MART **+CAS**	$\mathrm{BCE}(\boldsymbol{p}(\boldsymbol{x}', \theta), y) + \lambda \cdot \mathrm{KL}(\boldsymbol{p}(\boldsymbol{x}, \theta)\|\|\boldsymbol{p}(\boldsymbol{x}', y)) \cdot (1 - \boldsymbol{p}_y(\boldsymbol{x}, \theta))$ $+\frac{\beta}{S}\sum_{s=1}^{S}\mathrm{BCE}(\hat{\boldsymbol{p}}^s(\boldsymbol{x}', \theta, M^s), y) + \beta \cdot (\frac{\lambda}{S} \cdot \sum_{s=1}^{S}\mathrm{KL}(\hat{\boldsymbol{p}}^s(\boldsymbol{x}, \theta, M^s)\|\|\hat{\boldsymbol{p}}^s(\boldsymbol{x}', \theta, M^s)) \cdot (1 - \hat{\boldsymbol{p}}_y^s(\boldsymbol{x}, \theta, M^s))$

C Algorithm of CAS method

Algorithm 1 Robust Training with CAS.

Input: Training data $\{\boldsymbol{x}_i, y_i\}_{i=1,2,\ldots,n}$, DNN $\mathcal{F}(\theta)$, CAS modules with parameters M, maximum training epochs T

Output: Robust Network $\mathcal{F}$

```
for t in [1, 2, ··· , T] do
    for minibatch {x_1, ··· , x_b} do
        Generate adversarial examples using PGD attack on Eq. 5
        Compute the CAS loss in Eq. 4 using the channel-wise activations f̂^l
        Reweight the original features f̃^l = f^l ⊗ M_y^l using the parameters M_y^l in CAS
        Forward with the adjusted f̃^l and compute the CE loss at the output layer
    end for
    Optimize all parameters (θ, M) by Eq. 5 using gradient descent
end for
```

D Sensitivity of CAS to Parameter β

As mentioned in Section 4, the parameter β on the CAS loss controls the suppressing strength. To test the sensitivity of CAS training under different β, we insert a CAS module to the 4-th block of ResNet18, and train the network on CIFAR-10 using AT with CAS under $\beta \in [0, 0.5, 1, 2, 5, 10, 20]$. Note that $\beta = 0$ indicates the standard adversarial training (AT). We test the robustness of the models

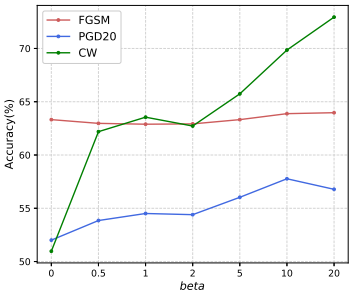


Figure 8: Robustness of AT+CAS against white-box attacks FGSM, PGD-20 and CW_∞ under different β. As β increases, the robustness is also improved, especially against the CW_∞ attack.

in a white-box setting against FGSM, PGD-20 and CW_∞ as shown in Figure 8. As can be observed, the models trained with larger β are generally more robust, especially against the CW_∞ attack. This is because larger β increases the strength of channel suppressing, leading to larger inter-class margins. As we further show in Figure 9, the representations learned with CAS are more separated between different classes, and are more compact within the same class. This tends to increase the difficulty of margin-based attacks like CW_∞.

E CAS IMPROVES REPRESENTATION LEARNING

The representations learned by natural or adversarial training with or without our CAS strategy are illustrated in Figure 9. The t-SNE (Maaten & Hinton, 2008) 2D embeddings are computed on deep features extracted at the penultimate layer of ResNet18 on CIFAR-10. As can be observed, our CAS training improves representation learning for both natural training and adversarial training. This is largely attributed to the strong channel suppressing capability of our CAS training. Channel suppressing helps learn high-quality representations with high inter-class separation and intra-class compactness. Interestingly, our CAS training can even improve the performance of natural training from 92.75% to 94.56%. This implies that our CAS is a generic training strategy that can benefit both model training and representation learning. Although CAS is not a direct regularization technique, it can achieve a similar representation regularization effect as existing representation enhancing techniques like the center loss.

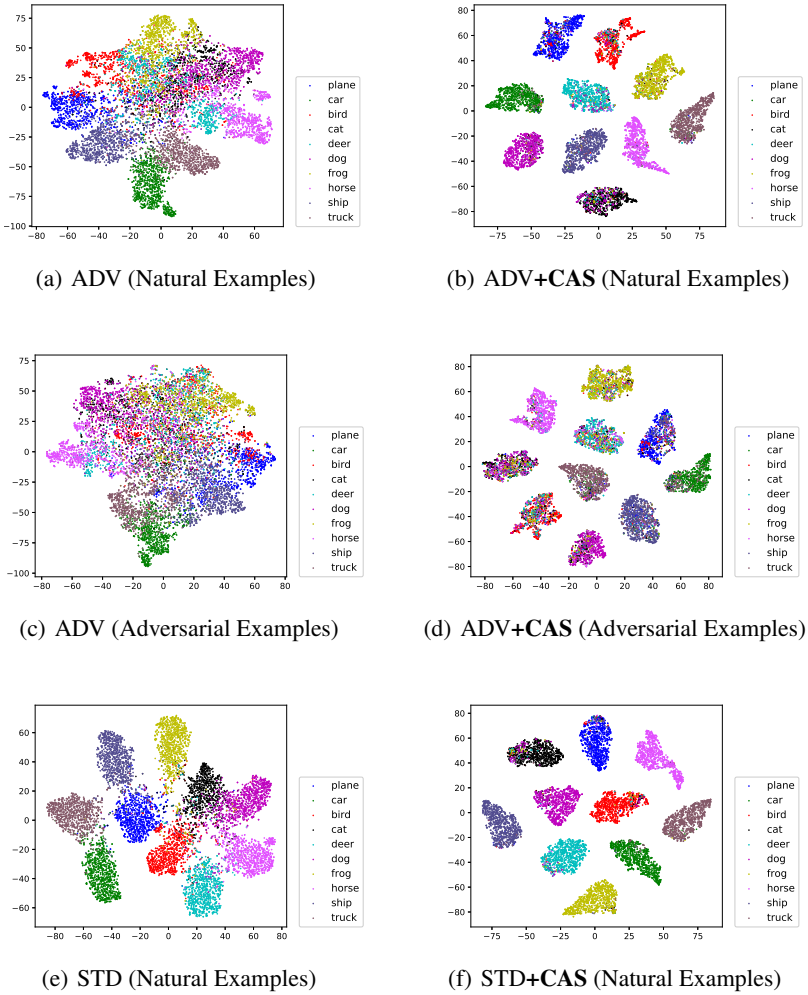
F MORE EXPERIMENTAL RESULTS

F.1 WIDERESNET RESULTS ON CIFAR-10

The white-box robustness of WideResNet-34-10 (Zagoruyko & Komodakis, 2016) models trained using AT, AT+CAS, TRADES, TRADES+CAS, MART and MART+CAS are reported in Table 8. We attach the CAS module to the last two convolution layers of the network, and set $\beta = 2$. The training settings are the same as that of ResNet18, except that, here we use weight decay 5e-4. The 'best' and 'last' results indicate the best and last checkpoints, respectively. For our CAS, we generate the attacks using the same *adaptive white-box attacks* as used in Section 5.2. The robustness of AT, TRADES and MART can be consistently improved by our CAS training. The most improvements are achieved on AT. These results confirm that our CAS training can lead to consistent improvements to different adversarial training methods on more complex models.

F.2 VGG16 RESULTS ON CIFAR-10

Similar to the above WideResNet experiments, here we report the results of VGG16 (Simonyan & Zisserman, 2014) on CIFAR-10 in Table 9. For VGG16, we attach the CAS module to its last three convolution layers, and set $\beta = 3$. We can see that our CAS training can enhance the robustness of 'best' models by a significant margin of 7%-10%. Note that the accuracy of natural training are also improved by a substantial margin. Compared with complex models such as ResNet18 and WideResNet, our CAS training is even more beneficial for small capacity models (*e.g.*, VGG16).



(a) ADV (Natural Examples)

(b) ADV**+CAS** (Natural Examples)

(c) ADV (Adversarial Examples)

(d) ADV**+CAS** (Adversarial Examples)

(e) STD (Natural Examples)

(f) STD**+CAS** (Natural Examples)

Figure 9: The t-SNE 2D embeddings of deep features extracted at the penultimate layer of ResNet18 models trained using natural ('STD') or adversarial training ('ADV') on CIFAR-10. The embeddings are shown separately for natural versus adversarial examples. Our CAS training can help improve inter-class separation and intra-class compactness for both types of training.

Table 8: White-box robustness (accuracy (%) on various white-box attacks) of WideResNet-34-10 on CIFAR-10 dataset. '**+CAS**' indicates applying our CAS training to existing defense methods. The best results are **boldfaced**.

Defense	CIFAR-10							
	Natural		FGSM		PGD-20		CW_∞	
	Best	Last	Best	Last	Best	Last	Best	Last
AT	84.16	84.23	66.62	61.27	53.75	49.53	50.52	47.68
AT**+CAS**	**88.25**	**89.06**	**67.14**	**66.98**	**56.28**	**53.22**	**58.54**	**54.77**
TRADES	86.23	86.41	66.41	65.63	54.42	52.64	53.45	52.43
TRADES**+CAS**	**87.07**	**87.15**	**66.92**	**66.15**	**55.43**	**53.15**	**61.46**	**57.07**
MART	84.09	85.69	67.24	66.41	57.56	54.49	54.42	52.55
MART**+CAS**	**87.87**	**89.20**	**68.09**	**67.96**	**58.24**	**54.95**	**61.48**	**57.11**

Table 9: White-box robustness (accuracy (%) on various white-box attacks) of VGG16 on CIFAR-10. '**+CAS**' indicates applying our CAS training strategy to existing defense methods. The best results are **boldfaced**.

Defense	CIFAR-10							
	Natural		FGSM		PGD-20		CW_∞	
	Best	Last	Best	Last	Best	Last	Best	Last
AT	70.32	70.38	51.87	51.85	42.23	42.01	43.63	43.81
AT**+CAS**	**82.25**	**82.61**	**59.77**	**57.56**	**49.22**	**42.73**	**53.03**	**49.68**
TRADES	74.98	76.67	52.10	52.99	41.53	41.13	45.50	45.25
TRADES**+CAS**	**83.45**	**83.25**	**61.36**	**61.51**	**49.86**	**49.56**	**52.60**	**52.47**
MART	71.70	72.20	54.72	54.90	46.55	46.53	44.52	44.72
MART**+CAS**	**81.53**	**83.28**	**61.89**	**61.70**	**52.01**	**49.13**	**50.97**	**49.10**

F.3 More Results of ResNet18 on SVHN and CIFAR-10

We report the white-box robustness of the 'best' models of ResNet18 on SVHN and CIFAR-10 in Table 10, as a supplementary to the 'last' results in Table 5. The 'best' represents the best (highest evaluation robustness) checkpoint of the model. Again, our CAS can also improve the robustness of the 'best' models consistently. Therefore, our CAS can constantly improve both the robustness and the natural accuracy throughout the entire training process.

Table 10: White-box robustness (accuracy (%) on various white-box attacks) of ResNet18 on CIFAR-10 and SVHN on the best checkpoint. '**+CAS**' indicates applying our CAS training strategy to existing defense methods. The best results are **boldfaced**.

Defense	SVHN				CIFAR-10			
	Natural	FGSM	PGD-20	CW_∞	Natural	FGSM	PGD-20	CW_∞
AT	91.41	69.52	53.07	50.38	84.20	**63.32**	52.01	50.97
AT**+CAS**	**92.69**	**71.83**	**54.78**	**54.13**	**86.72**	62.92	**54.40**	**62.72**
TRADES	90.40	71.22	57.71	54.49	82.68	63.15	53.05	50.46
TRADES**+CAS**	**91.41**	**71.39**	**59.32**	**66.94**	**84.75**	**63.94**	**57.20**	**67.77**
MART	87.37	68.58	57.72	51.48	78.90	63.29	54.92	50.31
MART**+CAS**	**91.04**	**72.32**	**60.29**	**57.05**	**85.99**	**64.23**	**58.21**	**69.91**

F.4 Robustness against AutoAttack

We report the white-box AutoAttack (Croce & Hein, 2020b) evaluation results of ResNet18 on CIFAR-10 in Table 11. AutoAttack is a recently proposed attack with two proposed Auto-PGD attacks and the other two complementary attacks (Croce & Hein, 2020a). AutoAttack has been shown can produce more accurate robustness evaluations on a wide range of adversarial training defenses. As shown in Table 11, although less significant than against regular attacks like PGD and CW, our CAS training can still improve the robustness of AT, TRADES and MART by a noticeable margin. This confirms that the improvements brought by our CAS training is 'real' and substantial, rather than obfuscated gradients nor improper evaluation.

F.5 More Robustness Evaluations of CAS Module

The robustness of our CAS module is further evaluated in this section. Our CAS module could make the margin loss between different classes become large, which brings some concern whether the mistakes in CAS module would destroy the final results. However, the proposed CAS could make some misclassified boundary adversarial examples in the baseline defense model be correctly classified with the CAS module. As the pulling away of decision boundary margin might cause some imbalanced gradient risks, in Table 12 we also evaluate the robustness against the Margin Decomposition attack (Jiang et al., 2020) to confirm the improvement of CAS robustness does not come from the imbalanced gradient. The mentioned MD attack is set with the default setting in its

Table 11: White-box robustness (accuracy (%) against AutoAttack) of ResNet18 on CIFAR-10. '**+CAS**' indicates applying our CAS training strategy to existing defense methods. The best results are **boldfaced**.

Defense	AutoAttack	
	Best	Last
AT	46.58	41.90
AT**+CAS**	**47.40**	**44.74**
TRADES	48.28	47.46
TRADES**+CAS**	**48.40**	**48.38**
MART	47.06	45.45
MART**+CAS**	**48.45**	**46.39**

paper. Moreover, we evaluate our CAS method against PGD-20 (step size $\epsilon/10$) under various attack strengths. Robustness drops when ϵ increases in Table 13, that is, CAS is not obfuscated gradient.

Table 12: Robustness of our CAS module of ResNet18 on CIFAR-10 against MD attack.

Defense	Loss Type		
	MD(CE)	MD(CAS)	MD(CE+CAS)
AT	43.35	–	–
AT**+CAS**	–	**46.27**	**49.56**
TRADES	48.47	–	–
TRADES**+CAS**	–	**53.51**	**55.53**
MART	46.99	–	–
MART**+CAS**	–	**49.89**	**55.75**

Table 13: Robustness of CIFAR-10 on ResNet18 against PGD-20 with different ϵ.

Attack Strength ϵ	2	4	8	16	32
AT**+CAS**	79.15	70.26	48.88	16.64	2.48
TRADES**+CAS**	80.09	72.69	55.99	27.94	5.10
MART**+CAS**	81.01	73.56	54.37	20.49	2.19