AGAIN: Adversarial Training with Attribution Span Enlargement and Hybrid Feature Fusion

Shenglin Yin[1], Kelu Yao[2, 3, *], Sheng Shi[4, 5], Yangzhou Du[5], Zhen Xiao[1, *]
[1]School of Computer Science, Peking University, China
[2]Zhejiang Laboratory, Hangzhou 311100, China
[3]Institute of Computing Technology, Chinese Academy of Sciences, China
[4]Northwest University, Xi'an 710127, P. R. China
[5]AI Lab, Lenovo Research, Beijing 100094, P. R. China
yinsl@stu.pku.edu.cn
yaokelu@ict.ac.cn
{shisheng2, duyz1}@lenovo.com
xiaozhen@pku.edu.cn

Abstract

The deep neural networks (DNNs) trained by adversarial training (AT) usually suffered from significant robust generalization gap, i.e., DNNs achieve high training robustness but low test robustness. In this paper, we propose a generic method to boost the robust generalization of AT methods from the novel perspective of attribution span. To this end, compared with standard DNNs, we discover that the generalization gap of adversarially trained DNNs is caused by the smaller attribution span on the input image. In other words, adversarially trained DNNs tend to focus on specific visual concepts on training images, causing its limitation on test robustness. In this way, to enhance the robustness, we propose an effective method to enlarge the learned attribution span. Besides, we use hybrid feature statistics for feature fusion to enrich the diversity of features. Extensive experiments show that our method can effectively improves robustness of adversarially trained DNNs, outperforming previous SOTA methods. Furthermore, we provide a theoretical analysis of our method to prove its effectiveness.

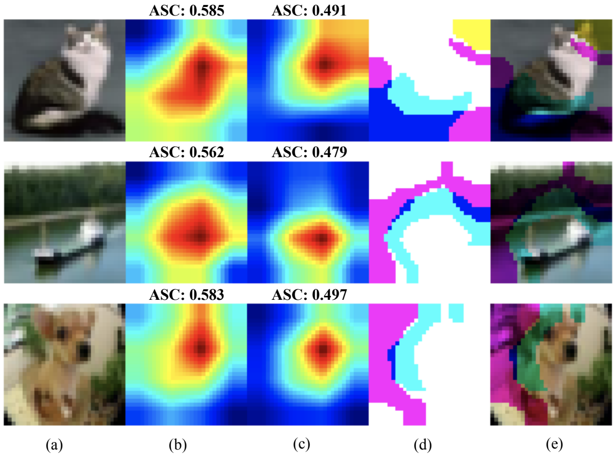


Figure 1. A visual illustration of attribution span under ResNet-18. (a) is the original image; (b) and (c) are attribution spans of the standard model and robust model in the inference phase, respectively. ASC is **A**ttribution **S**pan **C**overage; (d) is the difference between the standard model and the robust model in terms of attribution span; (e) is the result after partial feature erasure of the original image using (d).

1. Introduction

Deep neural networks (DNNs) have shown remarkable success in solving complex prediction tasks. However, recent studies have shown that they are particularly vulnerable to adversarial attacks [22], which take the form of small perturbations to the input that cause DNNs to predict incorrect outputs. The defense of adversarial examples has been intensively studied in recent years and several defenses against adversarial attacks have been proposed in a great deal of work [17, 18].

Among the various existing defense strategies, adversarial training (AT) [10, 15] has been shown to be one of the most effective defenses [16] and has received a lot of attention from the research community. However, adversarially

[*] Zhen Xiao and Kelu Yao are the corresponding authors.

trained DNNs typically show a significant robust generalization gap [27]. Intuitively, there is a large gap between the training robustness and test robustness of the adversarially trained model on the adversarial examples. Some existing methods [23, 25, 27] narrow the robust generalization gap from the perspective of weight loss landscapes. Other existing methods [13,24,32] enhance robust generalization from the perspective of training strategies. However, this work ignores a critical factor affecting generalization robustness, which is the learned knowledgeable representation.

Training DNNs with robust generalization is particularly difficult, typically possessing significantly higher sample complexity [8, 29, 31] and requiring more knowledgeable [4, 19]. Compared with standard DNNs, we discover that the generalization gap of adversarially trained DNNs is caused by the smaller attribution span on the input image. In other words, adversarially trained DNNs tend to focus on specific visual concepts on training images [8], causing its limitation on test robustness. Specifically, we explore the difference between the standard model (training w/o AT) and the robust model (training w/ AT) in the inference phase through empirical experiments. As shown in Figure 1 (b) and Figure 1 (c), the standard model and the robust model have different attribution span for the same image in the inference phase, and the attribution span of the standard model is larger than that of the robust model in general. Through our further exploration, we find that these different spans (see Figure 1 (d)) affect the model's decision on clean data and hardly affect the model's decision on adversarial examples. This indicates that AT enables the model to learn robust features, but ignores the features of generalization. This motivates us to design a method to enlarge the attribution span to ensure that the model focuses on robust features while enhancing the focus on other features to improve the generalization ability of the robust model.

To this end , we propose a generic method to boost the robust generalization of AT from the novel perspective of attribution span. Specifically, we use the class activation mapping to obtain the attribution span of the model under real and fake labels, and mix these two spans proportionally to complete the enlargement of the attribution span and make the model focus on the features within this span during the training process. In addition, in order to increase the diversity of features and ensure the stable training of the model under the enlarged attribution span, we adopt the feature fusion implemented by hybrid feature statistics to further improve the generalization ability of the model. Compared to other methods, our method can further improve the accuracy of the model on clean data and adversarial examples. Meanwhile, our work provides new insights into the lack of good generalization of robust models.

Our main contributions are summarized as follows.

- We find that adversarially trained DNNs focus on a smaller span of features in the inference phase and ignores some other spans of features. These spans are generally associated with generalization ability and have little impact on robustness.

- We propose a method to boost AT, called AGAIN, which is short for **A**ttribution Span Enlar**G**ement and Hybrid Fe**A**ture Fus**I**o**N**. During model training, we expand the region where the model focuses its features while ensuring that it learns robust features, and combine feature fusion to enhance the generalization of the model over clean data and adversarial examples.

- Extensive experiments have shown that our proposed method can better improve the accuracy of the model on clean data and adversarial examples compared to state-of-the-art AT methods. Particularly, it can be easily combined with other methods to further enhance the effectiveness of the method.

2. Related Work

2.1. Adversarial Attack Methods

Sezgedy et al. [22] first introduced the concept of adversarial examples, i.e., adding noise that is imperceptible to the human eye to the original clean examples, so that the perturbed examples cause DNNs prediction errors. After this concept was introduced, many works investigated the robustness of the model and proposed a series of attack methods. Goodfellow et al. [10] proposed a classical adversarial attack method called Fast Gradient Sign Method (FGSM), which finds the most aggressive perturbation within a fixed range of perturbations by exploiting the gradient information of the model. Madry et al. [15] further improved FGSM by proposing a multi-step version of FGSM called projected gradient descent (PGD). PGD generates stronger adversarial examples by means of multi-step iterative projection. Carlini Wagner et al. [3] proposed an optimization-based method to generate adversarial examples that can be widely used to evaluate the robustness of deep learning models. Croce et al. [6] proposed a parameter-free combination of attacks to evaluate the robustness of the model. First they proposed two extensions of the PGD attack to overcome failures due to suboptimal step size and objective function problems. Then they combined the new attack with two complementary existing attacks, which is called AutoAttack.

2.2. AT Defense Methods

As a series of attack methods have been proposed, a large number of defense strategies have been developed to defend against adversarial attacks. Athalye et al. [1] showed that: most defense methods are ineffective against gradient mask-based adaptive adversarial attacks, and only the

AT defense strategy is the only proven effective defense. AT defends against adversarial attacks by using adversarially generated data in model training [10] and is formulated as a minimal optimization problem. Madry et al. [15] proposed the main AT framework to improve the robustness of the model. The channel-wise activation suppressing (CAS) [2] strategy suppresses redundant activations from adversarial perturbations during AT. Wang et al. [25] improved the process of generating adversarial examples by simultaneously applying misclassified clean examples, as well as adversarial examples for model training (MART). Zhang et al. [31] explored the tradeoff between standard accuracy and adversarial robustness. To achieve a better tradeoff, they decomposed the adversarial prediction error into natural error and boundary error and proposed TRADES to control these two terms simultaneously (TRADES). Zhang et al. [32] proposed Friendly Adversarial Training (FAT), instead of using loss-maximizing mostly adversarial examples, they searched for the least lossy adversarial examples among the adversarial examples with confident misclassification. The solution proposed by Cui et al. [7] is to constrain the logits from a robust model that takes adversarial examples as input and makes them similar to the logits of a clean model with corresponding natural examples as input (LBGAT). Jia et al. [13] proposed the concept of ”learnable attack policy”, called LAS-AT, which learns to automatically generate attack policies to improve the robustness of the model. All these methods improve the robustness of the model, but still suffer from insufficient generalization.

2.3. Class Activation Mapping

The class activation mapping [33] is one of the methods to visualize the attribution span, which highlights the distinguishing object fraction detected by the convolutional neural network. Given a classification network, the class activation mapping uses the parameters of the final fully connected layer obtained from training as class weights are projected onto the final convolutional features and the weighted features are linearly summed to identify the importance of the image regions. The equation is as follows:

$$\mathbf{M}_c = \sum_k \mathbf{w}_k^c \mathbf{A}_k, \tag{1}$$

where $\mathbf{A}_k$ denotes the value of the k-th feature map; $\mathbf{w}_k^c$ denotes the class weight of the k-th feature map corresponding to class c; and $\mathbf{M}_c$ denotes the sum of the weights of the different activation feature maps for identifying a certain class c, which is the attribution span that the model focuses on during inference.

In addition, Selvaraju et al. [20] proposed a method called Grad-CAM for improving the interpretability and transparency of deep neural networks. The Grad-CAM method is based on gradient information and can be applied to any type of convolutional neural network and does not require modifying the structure of the model. Chattopadhay et al [5] improved on Grad-CAM by proposing Grad-CAM++ and providing a mathematical interpretation. The method uses the weighted combination of the positive and negative derivatives of the last convolutional layer feature maps with respect to the specific category scores as weights to produce visual interpretations for the considered category labels.

3. Attribution Span of The Model

In this section, we investigate the difference between the standard model and the robust model from the perspective of attribution span and show the correlation between attribution span and generalization ability. Specifically, we train ResNet-18 [11] and VGG-16 [21] on CIFAR-10 [14] using standard training and PGD-AT [15]. Then, we visualize the attribution span in the second last layer of the model using class activation mapping. We explore two main questions:

Q1: **What differences exist between the standard model and the robust model in terms of attribution span?**

To begin with, in order to quantify the attribution span and to better analyze the difference between the standard model and the robust model in terms of attribution span, we define the **A**ttribution **S**pan **C**overage (ASC). The larger the ASC, the wider the attribution span of the model.

$$ASC = \frac{\sum_{i=1}^{N} \mathbb{I}(M_i > \lambda \cdot M_{max})}{N}, \tag{2}$$

where $\mathbb{I}$ is the indicator function; $\mathbf{M}$ is the attribution span calculated according to Equation (1); N is the total number of elements in $\mathbf{M}$; M_i is the i-th element in $\mathbf{M}$ and M_{max} is the maximum value in $\mathbf{M}$; λ is the hyperparameter, which is set to 0.5 here, i.e., the model is considered to be concerned with this span when the elements in $\mathbf{M}$ are greater than 50% of the maximum value.

Taking the CIFAR-10 dataset as an example, the experimental results are shown in Figure 1(b) and Figure 1(c). As can be seen from the results, the ASC of the standard model is larger than that of the robust model, which is common throughout the dataset. The ResNet-18 model has this phenomenon in 79.38% of the data in the entire CIFAR-10, of which the average ASC of the standard model is 54.86%, and the average ASC of the robust model is 51.80%; VGG-16 model in the entire CIFAR-10 dataset, this phenomenon exists in 74.15% of the data, where the average ASC of the standard model is 46.84%, and the average ASC of the robust model is 43.70%. This indicates that training the model using AT makes the model focus on some of the more robust features and ignores the learning of other features, which results in a smaller ASC value for the robust model.

Q2: **What decisions of the model are affected by these differences.**

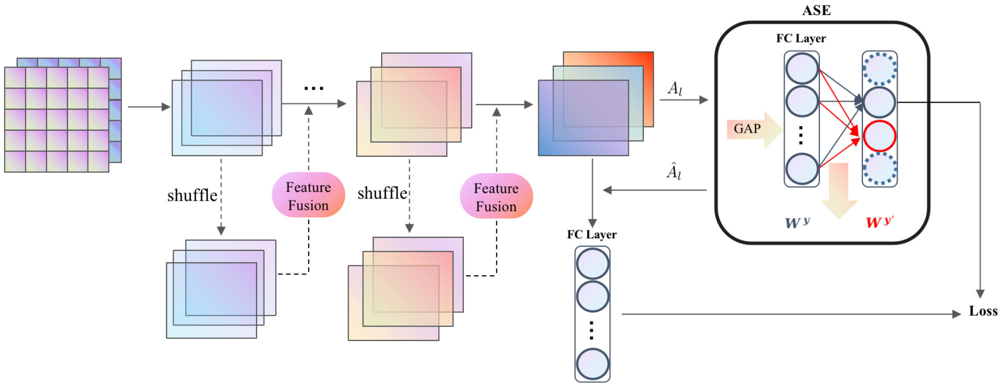


Figure 2. A visual illustration of our proposed method. It consists of two parts: attribution span enlargement (ASE) and hybrid feature fusion (HFF). ASE is used in the second last layer of the model; HFF is used before ASE.

Table 1. Accuracy of the model on original data and corrupted data. These features mainly affect clean data and essentially have no effect on the adversarial examples.

Network	Training method	Original clean	Corrupted clean	Original adv	Corrupted adv
ResNet-18	Std training	100%	79.4%	00.7%	03.6%
	Adv training	100%	84.4%	76.9%	72.3%
VGG-16	Std training	100%	65.3%	00.3%	04.3%
	Adv training	100%	85.5%	76.6%	71.7%

In the previous section, we found that the attribution span of the robust model is generally smaller than that of the standard model. Next, we explore $Q2$.

We first obtain the different spans in the attribution span for the standard model and the robust model. Then we binarize it to generate a mask (see Figure 1(d), where white represents the feature span to be saved and the other colors represent the feature span to be deleted). Then we perform a Hadamard product of the mask with the original data to get the corrupted data (see Figure 1(e)). We select data from the original dataset that can be classified correctly by the model and generate adversarial examples using PGD attack [15].

The mask is applied to the clean data and the adversarial examples to generate corrupted clean data (Corrupted clean) and corrupted adversarial examples (Corrupted adv), respectively, and the model is used to classify these corrupted data. The results are shown in the Table 1. Analyses are as follows. Taking ResNet-18 as an example, the accuracy of the model under standard training decreased by 20.6% on the corrupted clean data compared with the original clean data; the accuracy of the model trained under AT decreased by 15.6%. In the same case, the accuracy of the model on the adversarial examples do not change significantly.

Through empirical experiments, we find that these attribution spans that are ignored by the robust model are highly susceptible to influence the model's decisions of clean data. In addition, the adversarial perturbation has an inherent structure, which is broken by some methods of random masking thus reducing the aggressiveness [28]. But these attribution spans ignored by the robust model hardly affect the aggressiveness of the adversarial examples, indicating that no special attention is paid to these spans when generating the adversarial examples. Through the study of the above two problems, we propose a hypothesis:

In the process of AT, ensuring that the model learns robust features while enlarging the attribution span of its learning can improve the generalization ability of the robust model.

To examine this hypothesis, we propose a novel AT strategy. We describe this method in detail in the next section. Subsequently, our proposed hypothesis is confirmed by a large number of experiments, and the experimental results show that our proposed method significantly improves the accuracy of the model on clean data and adversarial examples.

4. The Proposed Method

In this section, we present the proposed method in detail. Its overall framework is shown in Figure 2. It consists of two parts: Attribution Span Enlargement and Hybrid Feature Fusion.

4.1. Attribution Span Enlargement

Two main tasks need to be accomplished in this part: 1) enable the model to learn the original robust features to ensure the robustness of the model itself; 2) enable the model to learn features from other spans to achieve the enlargement of attribution span and improve the generalization ability of the model. The class activation mapping gives us a solution to this problem.

First, we obtain the robust attribution span that the model focuses on under the true label y of the data. Secondly, we randomly disrupt y in the same batch to generate y'. Then under the fake label, we obtain the other attribution span of the model. Finally, these two attribution spans are fused in a certain ratio to generate a new attribution span to achieve our goal. The point to note is that, unlike the purpose of class activation mapping, we need the weighted features to maintain their original dimensionality for the subsequent process of feature extraction or classification, and thus there is no need to perform the weighted summation operation at the end. In order to achieve the final purpose, we modify Eq. 1. The equation is as follows.

$$\hat{\mathbf{A}}_1 = \alpha \cdot \mathbf{A}_l \cdot \mathbf{W}^y + (1-\alpha) \cdot \mathbf{A}_l \cdot \mathbf{W}^{y'}, \qquad (3)$$

where $\mathbf{A}_l \in \mathbb{R}^{B \times C_l \times W_l \times H_l}$ is the feature at l-th layer of the model, B is the number of samples in a batch, and C_l, W_l, and H_l are the number of channels, width, and height of the feature output at l-th layer, respectively. $\hat{\mathbf{A}}_l$ is the weighted feature; $\mathbf{W}^y \in \mathbb{R}^{B \times C_l \times 1 \times 1}$ and $\mathbf{W}^{y'}$ represent the parameter weights corresponding to the true label and fake label in the fully connected layer, respectively; $\alpha \in [0.5, 1]$ is a hyperparameter to balance the weights between the robust attribution span and other attribution span.

In addition, the class activation mapping is implemented by replacing the fully connected layer of the original model with a global average pooling layer. This is not applicable in our case. An alternative method is to first train a robust model using AT, such as PGD-AT, to provide the desired attribution span. However, this method is wasteful of resources and not flexible enough.

To address this problem, we received inspiration from [2] and designed an auxiliary network (see Figure 2), called Attribution Span Enlargement (ASE), containing only a fully connected layer to implement the above process. We modify Eq. 3 and the final equation is as follows.

$$\hat{\mathbf{A}}_l = \alpha \cdot \mathbf{A}_l \cdot \mathbf{W}^y_{ASE} + (1-\alpha) \cdot \mathbf{A}_l \cdot \mathbf{W}^{y'}_{ASE}, \qquad (4)$$

where $\mathbf{W}^y_{ASE}$ and $\mathbf{W}^{y'}_{ASE}$ represent the parameter weights corresponding to the true and fake labels in the fully connected layer of the ASE, respectively.

In ASE, we first perform a global average pooling operation on the features of the l-layer, and then feed them into the fully connected layer for classification. In the inference phase, we weight the features using only the parameter weights $\mathbf{W}^{ped}_{ASE}$ corresponding to the labels predicted by the model in the fully connected layer of ASE. It is worth noting that during inference, we do not randomly generate fake labels, so there is no gradient confusion [1]. ASE has its own independent output and thus needs to be trained jointly with the original classification network.

4.2. Hybrid Feature Fusion

The purpose of feature fusion is twofold: 1) to increase the diversity of features and provide a richer attribution span for ASE to make the model training more stable. 2) to make the model pay more attention to the structural information of the data and further increase the robustness of the model [9]. Inspired by AdaIN [12], we designed the Hybrid Feature Fusion (HFF). In practice, we randomly shuffle the output features A_l to generate a new set of features $\mathbf{A}'_l$ as the style data. First we use AdaIN to calculate hybrid features $\mathbf{A}^1_l$ and $\mathbf{A}^2_l$: $\mathbf{A}^1_l = AdaIN(\mathbf{A}_l, \mathbf{A}'_l)$, $\mathbf{A}^2_l = AdaIN(\mathbf{A}'_l, \mathbf{A}_l)$. Second, we use the original features and the generated hybrid features to calculate the hybrid feature statistics. The equation is as follows.

$$\begin{aligned} \hat{\mu} &= \gamma_1 \cdot \mu(\mathbf{A}_l) + \gamma_2 \cdot \mu(\mathbf{A}'_l) + \gamma_3 \cdot \mu(\mathbf{A}^1_l) + \gamma_4 \cdot \mu(\mathbf{A}^2_l) \\ \hat{\sigma} &= \gamma_1 \cdot \sigma(\mathbf{A}_l) + \gamma_2 \cdot \sigma(\mathbf{A}'_l) + \gamma_3 \cdot \sigma(\mathbf{A}^1_l) + \gamma_4 \cdot \sigma(\mathbf{A}^2_l), \end{aligned} \qquad (5)$$

where the mean $\mu(\cdot)$ and variance $\sigma(\cdot)$ are calculated independently for each channel and sample; $\gamma_1, \gamma_2, \gamma_3, \gamma_4 \in [0, 1]$ are randomly generated, and $\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 = 1$. Finally, the hybrid feature statistics are applied to the style-normalized A_l.

$$FeatureFusion(\mathbf{A}_l) = \hat{\sigma} \frac{\mathbf{A}_l - \mu(\mathbf{A}_l)}{\sigma(\mathbf{A}_l)} + \hat{\mu}. \qquad (6)$$

To obtain a new hybrid feature, we linearly interpolated the statistics from these four features. That is, unlike previous methods, our approach mixes the statistics from multiple dimensions, which leads to a richer enhancement effect. HFF can be easily applied on each layer of the model, and in this paper, we apply it before ASE.

4.3. Training Strategy

In our proposed method, ASE can be considered as an auxiliary network with its own parameters and outputs. Therefore, it needs to be trained together with the original network during the training process. In view of this, we use a joint loss function [2] to train the model. Moreover, although feature fusion is used in our proposed method, we do not fuse on the labels during the training process, but use the original labels. Under AT, the overall loss function is as follows.

$$\begin{aligned} L &= L_{CE}(F_{ori}(\mathbf{x}_{adv}), y) \\ &+ L_{CE}(F_{ASE}(F^l_{ori}(\mathbf{x}_{adv})), y), \end{aligned} \qquad (7)$$

where $L_{CE}(\cdot)$ is the CrossEntropy Loss function; $F_{ori}(\cdot)$ is the output of the original model after softmax; $F_{ASE}(\cdot)$

is the output of the ASE module after softmax; $F_{ori}^{l}(\cdot)$ is the output of the l-th layer of the original model after global average pooling; $\mathbf{x}_{adv}$ and y are the adversarial example and the true label, respectively.

In addition, since the model does not learn the features of the data in the early stage of training. Therefore, in the early stage of model training, we use original AT to train the model with high probability. As training epochs increase, we gradually increase the probability of using the proposed method. The algorithm of the training process is shown in Appendix.

5. Experiments

In this section, we conduct comprehensive experiments on the CIFAR-10 and CIFAR-100 datasets [14] with ResNet-18 [11] and WideResNet-34-10 [30] to come in and assess the effectiveness of the proposed method, which includes an empirical exploration of the proposed method, comparison experiments, and ablation experiments. More models, datasets, and exploratory experimental results are presented in Appendix. Our code is available at https://github.com/InsLin/AGAIN.

5.1. Competitive Methods

In order to evaluate the effectiveness of proposed method, we compare proposed method with the current mainstream benchmark methods, which mainly include standard AT (PGD-AT) [15] , MART [25], TRADES [31], FAT [31], LBGAT [7], CAS [2], AWP [27] and LAS-AT [13]. The combination of our proposed method and some of these methods are respectively referred to as AGAIN-PGD-AT, AGAIN-MART and AGAIN-AWP

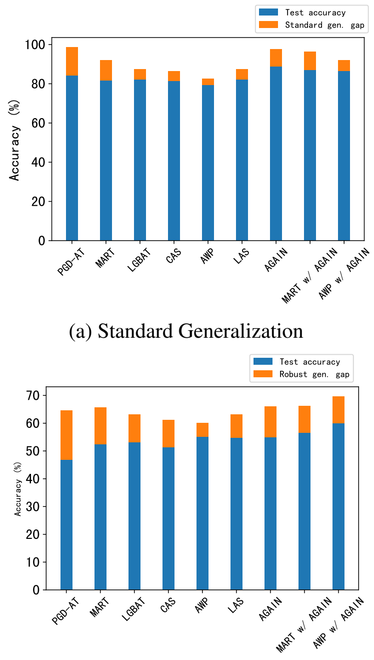
5.2. Implementation Details

5.2.1 Training Phase.

In the training phase, except for LAS-AT, we use a fixed training strategy: throughout the training, we use an SGD optimizer with an initialized learning rate of 0.1 and a learning rate variation of [0.1, 0.01, 0.001], adjusting the learning rate at the 75th, 90th and 100th epoch of training respectively, for a total of 120 epochs of training; the maximum perturbation intensity of the attack is 8/255, the step size is 4/255, and the number of iterations is 10. For LAS-AT, all settings are consistent with those in [13]. The hyperparameter α in our method is set to 0.6.

5.2.2 Evaluation Phase.

In the evaluation phase, the robustness of the model is evaluated by measuring the correct accuracy of the model under different adversarial attacks and approximating the upper bound of the robustness on the test set. We choose several adversarial attack methods to attack the trained model, including PGD (PGD-10, PGD-20, PGD-50, and PGD-100) [15], C&W [3], and AutoAttack (AA) [6]. The maximum perturbation strength of all attack methods under L_∞ is set to 8/255. Meanwhile, in order to fairly compare other defenses with proposed method, we use adaptive white-box attacks for the proposed method, i.e., the attacks are performed on the loss function of the original network and the loss function of the auxiliary network.



(a) Standard Generalization

(b) Robust Generalization

Figure 3. Standard generalization (on clean data) and robust generalization (on adversarial examples) of different methods.

5.3. Experiments on ResNet-18

Our proposed method is a plug-and-play and thus can be easily combined with other methods to further improve its robustness. We evaluate the robustness of all defense methods against several types of white-box attacks. The results for CIFAR-10 and CIFAR-100 are shown in Table 2 and 3.

From the experimental results, we can see that our proposed method outperforms other methods in most attack scenarios, not only improving the accuracy of the model on clean data, but also further improving the accuracy on adversarial examples. On CIFAR-10, our method achieves 87.88% accuracy on clean data, and the robustness under different attacks is significantly improved. Our method also improves the robustness of different defense methods under various attacks. The proposed method improves the perfor-

Table 2. Test robustness on the CIFAR-10 database. The best results are **boldfaced**, and the second best results are underlined.

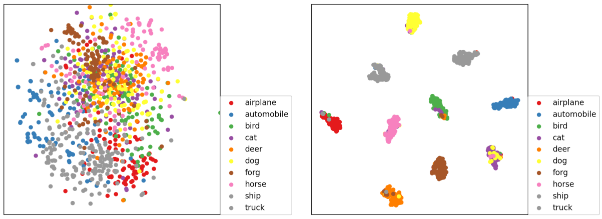
Method	Clean	PGD-10	PGD-20	PGD-50	PGD-100	C&W	AA
PGD-AT	84.25%	46.88%	46.56%	44.85%	44.76%	45.75%	41.69%
MART	81.61%	52.38%	51.28%	50.93%	50.80%	47.77%	46.09%
TRADES	83.64%	52.05%	50.67%	50.38%	50.20%	49.68%	48.41%
FAT	87.32%	45.80%	43.53%	43.11%	42.98%	43.50%	40.76%
LBGAT	85.73%	53.12%	52.05%	51.78%	51.68%	50.63%	49.04%
CAS	86.24%	51.38%	51.49%	51.77%	51.04%	53.66%	46.69%
AWP	79.45%	55.04%	54.47%	54.36%	54.30%	51.17%	49.40%
LAS-AT	82.39%	54.74%	53.70%	53.70%	53.72%	51.96%	49.94%
AGAIN-PGD-AT	**87.88%**	54.87%	54.43%	53.62%	53.13%	55.80%	49.31%
AGAIN-MART	87.13%	56.63%	56.00%	55.71%	55.67%	58.56%	50.77%
AGAIN-AWP	86.52%	**59.99%**	**59.35%**	**59.11%**	**58.85%**	**61.19%**	**51.89%**

Table 3. Test robustness on the CIFAR-100 database. The best results are **boldfaced**, and the second best results are underlined.

Method	Clean	PGD-10	PGD-20	PGD-50	PGD-100	C&W	AA
PGD-AT	62.34%	21.24%	21.38%	21.05%	21.01%	22.15%	19.76%
MART	55.14%	28.52%	28.08%	27.79%	27.91%	25.65%	24.04%
TRADES	58.18%	28.71%	28.25%	28.10%	27.99%	24.22%	24.03%
FAT	61.61%	19.33%	18.35%	18.08%	17.98%	19.31%	17.38%
LBGAT	56.78%	32.84%	32.21%	32.11%	32.07%	27.46%	26.39%
CAS	64.04%	31.66%	31.55%	31.26%	31.02%	34.82%	24.40%
AWP	54.00%	31.78%	31.49%	31.44%	31.74%	28.20%	26.19%
LAS-AT	58.38%	32.32%	31.89%	31.82%	31.77%	28.48%	26.84%
AGAIN-PGD-AT	**66.92%**	32.97%	32.88%	32.54%	32.15%	35.59%	26.21%
AGAIN-MART	63.61%	33.75%	33.69%	33.46%	33.28%	37.99%	27.22%
AGAIN-AWP	64.51%	**35.58%**	**35.44%**	**35.39%**	**35.08%**	**40.02%**	**28.69%**

mance of MART by 2.33% and 3.22% under PGD-100 and AA attacks, respectively. It also improves the performance of AWP by 4.55% and 3.49% under the same attacks, respectively. In addition, our method is able to achieve the highest accuracy of 61.19% under C&W attack. At CIFAR-100, the proposed method achieves not only the highest accuracy on clean data, but also the best robust performance under all attack scenarios. In detail, our proposed method helps MART and AWP to improve the accuracy on clean data by 8.47% and 10.51%, respectively. In terms of attacks, when our proposed method is combined with AWP, the highest accuracy is achieved under PGD-100 and AA attacks, reaching 35.08% and 28.69%, respectively.

In addition, it can also be seen that the improvement for C&W attack is more obvious than that for other attacks. As shown in Figure 4, the deep feature distribution learned by the trained model of our method is similar to Center Loss [26], i.e., it is more compact in the same class and more separated in different classes. This makes it more difficult for margin-based attacks such as C&W to succeed [2].



(a) AT w/o AGAIN

(b) AT w/ AGAIN

Figure 4. The t-SNE two-dimensional embedding of the depth features extracted from the penultimate layer of the ResNet-18 model trained on CIFAR-10 using our proposed training method (AGAIN) and PGD-AT.

5.4. Experiments on WideResNet-34-10

We use the WideResNet-34-10 [30] for experiments on the CIFAR-10 dataset. To evaluate the model, we use PGD [15] (PGD-20, PGD-50), C&W [3] and A&A [6]. The methods of comparison use MART [25], TRADES [31], FAT [31], AWP [27] and LAS-AT [13]. The experimental results are shown in the Table 4. It can be seen from the experimental results that when our proposed method is combined with other AT methods, it is still effective in improving the robustness of the model on large complex models with clean data and adversarial examples.

Table 4. WideResNet-34-10 result on CIFAR-10. The best results are **boldfaced**.

Method	Clean	PGD-20	PGD-50	C&W	AA
MART	83.63%	56.74%	56.44%	53.16%	51.23%
TRADES	84.91%	55.78%	55.10%	54.29%	52.95%
FAT	84.91%	49.91%	49.69%	49.13%	48.01%
AWP	84.12%	58.09%	57.84%	56.08%	53.19%
LAS-AT	86.16%	56.28%	56.07%	55.67%	53.08%
AGAIN-AWP	**90.31%**	**62.43%**	**62.29%**	**68.13%**	**53.59%**

Table 5. The effect of our proposed method at different layers on the CIFAR10 dataset.

Layer	Clean	PGD-20	PGD-100	C&W
Layer1	59.52%	29.04%	28.72%	33.47%
Layer2	78.09%	34.52%	34.05%	38.95%
Layer3	86.76%	42.68%	42.00%	40.56%
Layer4	87.88%	54.43%	53.13%	55.80%

Table 6. Results of ablation experiments

ASE	HFF	Clean	PGD-20	C&W
✗	✗	84.25%	42.36%	43.75%
✓	✗	86.33%	52.74%	53.97%
✗	✓	86.19%	44.71%	43.91%
✓	✓	87.88%	54.43%	55.80%

5.5. Location of Attribution Span Enlargement.

ASE module is flexible to use on different layers of the DNNs. Therefore, we explored the performance of the proposed method on different layers and the experimental results are shown in Table 5. It can be seen from the experimental results that the deeper the layers are, the better the results of our proposed method. The reason for this phenomenon is that ASE can be seen as a feature filtering process, and if used in the first few layers of DNNs, it will make the model lose some detailed features, which will affect the learning of more advanced features by the deeper network, and will eventually lead to a decrease in the accuracy of the model. The deeper layers are more relevant to the final prediction, so using it in the deeper layer will have a significant effect enhancement.

5.6. Analysis of the Generalization

In this section, we explore the generalization ability of the model under different methods. The experimental results are shown in Figure 3. From the experimental results, we can see that our methods, AGAIN and MART w/ AGAIN, can effectively reduce the standard generalization gap (standard gen. gap) and robust generalization gap (robust gen. gap) compared with PGD-AT and MART. Some other methods, such as AWP, LAS, etc., have smaller generalization gaps, but they are all implemented at the cost of reducing accuracy on the training dataset. Our method can maintain or even improve the accuracy on the training dataset while still achieving a smaller generalization gap. Meanwhile, when our method is combined with AWP, it can further narrow the generalization gap and improve the accuracy of the model on clean data and adversarial samples while ensuring the accuracy of the model on the training dataset.

5.7. Ablation Study

Our proposed method mainly consists of two parts: ASE and HFF. In this section, we verify the effectiveness of the proposed method. During the experiments, we remove the ASE and the HFF, respectively, and the experimental results are shown in Table 6. The accuracy of the clean data is significantly improved when only ASE is available, and the robustness under all attacks is improved. It can also be seen that the generalization of the robust model can be improved when HFF is used alone. The best results can be achieved when ASE and HFF are combined.

6. Conclusion

In this paper, we discover the difference in attribution span between standard and robust models, and explore a possible reason for the low generalization of the robust model from a new perspective. In order to improve the generalization of the robust model, we propose an AT approach based on attribution span enlargement and hybrid feature fusion. The method ensures that the model learns robust features while paying extra attention to features in other spans, and combines feature fusion to improve the accuracy of the model on clean data and adversarial examples. Comprehensive experiments show that our method is effective and general enough to improve the robustness of the model across different AT methods, network architectures and datasets.

7. Acknowledgment

The authors would like to thank the anonymous reviewers for their comments. Zhen Xiao and Kelu Yao are the corresponding authors.

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018. 2, 5

[2] Yang Bai, Yuyuan Zeng, Yong Jiang, Shu-Tao Xia, Xingjun Ma, and Yisen Wang. Improving adversarial robustness via channel-wise activation suppressing. *arXiv preprint arXiv:2103.08307*, 2021. 3, 5, 6, 7

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017. 2, 6, 7

[4] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[5] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018. 3

[6] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020. 2, 6, 7

[7] Jiequan Cui, Shu Liu, Liwei Wang, and Jiaya Jia. Learnable boundary guided adversarial training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15721–15730, 2021. 3, 6

[8] Christian Etmann, Sebastian Lunz, Peter Maass, and Carola-Bibiane Schönlieb. On the connection between adversarial robustness and saliency map interpretability. *arXiv preprint arXiv:1905.04172*, 2019. 2

[9] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 5

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2, 3

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3, 6

[12] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 5

[13] Xiaojun Jia, Yong Zhang, Baoyuan Wu, Ke Ma, Jue Wang, and Xiaochun Cao. Las-at: Adversarial training with learnable attack strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13398–13408, 2022. 2, 3, 6, 7

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3, 6

[15] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1, 2, 3, 4, 6, 7

[16] Pratyush Maini, Eric Wong, and Zico Kolter. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning*, pages 6640–6650. PMLR, 2020. 1

[17] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016. 1

[18] Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The odds are odd: A statistical test for detecting adversarial examples. In *International Conference on Machine Learning*, pages 5498–5507. PMLR, 2019. 1

[19] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *Advances in neural information processing systems*, 31, 2018. 2

[20] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 3

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[22] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1, 2

[23] Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. Augmax: Adversarial composition of random augmentations for robust training. *Advances in neural information processing systems*, 34:237–250, 2021. 2

[24] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. *arXiv preprint arXiv:2112.08304*, 2021. 2

[25] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019. 2, 3, 6, 7

[26] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016. 7

[27] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *neural information processing systems*, 2020. 2, 6, 7

[28] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1369–1378, 2017. 4

[29] Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In *International conference on machine learning*, pages 7085–7094. PMLR, 2019. 2

[30] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 6, 7

[31] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019. 2, 3, 6, 7

[32] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International conference on machine learning*, pages 11278–11287. PMLR, 2020. 2, 3

[33] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 3