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

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

1. Algorithm of AGAIN

Our training method is summarized in Algorithm 1.

Algorithm 1 AT with AGAIN

Input: Training Set, Initialized model $\mathcal{F}$, Maximum number of iterations N
Output: Robust model $\mathcal{F}$

for i in Range (0, N) **do**
  $prob = i/N$
  $flag = Random.uniform(0, 1)$
  Generate adversarial examples using PGD-10
  if $flag < prob$ **then**
    Training model using AGAIN
  else
    Training model using original AT
  end if
end for

2. Theoretical Motivation for Robust Generalization

We analyze the reasons why our proposed approach can improve robust generalization from two perspectives: model overfitting and the number of features learned by the model.

Ilyas et al. [4] argue that the features utilized by DNNs can be divided into robust features and non-robust features. Robust features are a small span of features that are interpretable to the human eye and AT usually makes the model focus on some specific visual spans, causing the model to ignore other features and making the attribution span smaller [1]. This is also consistent with our observations. In addition, adversarial examples generated under AT are not sufficient to cover all cases, which leads to insufficient training data. Therefore it can lead to model overfitting in this case [7].

Another intuitive explanation is that the key to a successful attack by an attacker is to corrupt the features that the model focuses on [9]. Our approach enables the model to use as many robust features as possible in its inference by enlarging the robust attribution span. Thus, if the model uses more robust features for inference, it is more expensive for an attacker to successfully attack it. Note that simply increasing the attribution span does not improve the robustness of the model. We need to expand the span of the robust features and therefore need to do it under AT. So, we expect to enlarge the attribution span of the model during AT so that it can learn richer features and improve generalization.

Next, we demonstrate through theoretical analysis that the proposed method can alleviate the phenomenon of model overfitting as well as can enable the model to learn more features.

We define $\mathbf{A}$ as the output of the second last layer of the model. The weight matrix of the last layer (fully connected layer) in the model as $\mathbf{W}$. The true label y in $\mathbf{W}$ corresponds to a weight of $\mathbf{W}^y$, and the fake label y' in $\mathbf{W}$ corresponds to a weight of $\mathbf{W}^{y'}$; $S(p_i) = \frac{\exp(pi)}{\sum_{j=0}^{K} \exp(p_j)}$ is

*Zhen Xiao and Kelu Yao are the corresponding authors.

softmax.

Theorem 1. *Define $F(\cdot)$ is a multi-classifier, when using Eq. (7) in main text to train $F(\cdot)$, the loss function can be defined as*

$$L = L_{CE}\left(S\left(\alpha \mathbf{A}^T\mathbf{W}^y + (1-\alpha)\,\mathbf{A}^T\mathbf{W}^{y'}\right), y\right). \quad (1)$$

Let $\mathbf{X} = \mathbf{A}^T\mathbf{W}^y - \mathbf{A}^T\mathbf{W}^{y'}$. Suppose $F(\cdot)$ is an ideal robust classifier, $L = 0$. The gradient of L w.r.t X is $\nabla_{\mathbf{X}} L = 0$. We can get the equation as

$$\mathbf{A}^T\mathbf{W}^y - \mathbf{A}^T\mathbf{W}^{y'} = \log\left\{\frac{\alpha\left[1+\sum_{j\neq y, j\neq y'}^{K-2} exp\left(\mathbf{A}^T\mathbf{W}^j - \mathbf{A}^T\mathbf{W}^{y'}\right)\right]}{1-\alpha}\right\}, \quad (2)$$

The detailed proof is presented in the Appendix. The following properties can be obtained from Theorem 1:

1). When α tends to 1, the model learns only the features associated with the correct class and $\mathbf{A}^T\mathbf{W}^y - \mathbf{A}^T\mathbf{W}^{y'} = \infty$. This indicates that during training, the model learns in the direction of infinitely increasing the logit difference between the correct and incorrect labels of the prediction, and too large a logit difference makes the model lack adaptability and overconfidence in the prediction [8]. And due to the existence of some regularization methods during model training, the output of logit is hardly infinity [3].

We analyze from the perspective of the number of features learned from the model. When considering the entire dataset, assuming that the dataset has K categories, then $\forall i, j \in K$ and $i \neq j$, it is necessary to satisfy $\mathbf{A}^T\mathbf{W}^y - \mathbf{A}^T\mathbf{W}^{y'} = \infty$, that is, $\mathbf{W}^y - \mathbf{W}^{y'} = \infty$. If $\mathbf{W}$ can satisfy the above condition, then the $\mathbf{W}$ will contain multiple invalid values that tend to infinity. In other words, the number of valid values used in the final calculation of the prediction probability becomes less when the neural network is inferred, which affects the generalization of the model.

2). When $\alpha \in [0.5, 1)$, $\mathbf{A}^T\mathbf{W}^y - \mathbf{A}^T\mathbf{W}^{y'} = C$, where C is a constant with respect to α. It can be found that when using our proposed method for AT, the difference between logit between correct labels and incorrect labels does not converge to infinity. With a guaranteed error of a certain size between the logit values of correct and incorrect labels, the loss will be small. And there will be more valid values in $\mathbf{W}$, so the model will use more features in the inference phase. This can improve the generalization ability of the model.

3. Proof of Theorem 1

First, let's recall the notations defined in the previous section and define some new ones.

$\mathbf{A}$ is the output of the second last layer of the model, where N is the number of features; the weight matrix of the last layer (fully connected layer) in the model is $\mathbf{W}$, where K is the number of sample categories; the true label y in $\mathbf{W}$ corresponds to a weight of $\mathbf{W}^y$, and the fake label y' in $\mathbf{W}$ corresponds to a weight of $\mathbf{W}^{y'}$; $S(p_i) = \frac{\exp(p_i)}{\sum_{j=0}^{K}\exp(p_j)}$ is softmax, where $p_i = A^T W^i$. $L_{CE}(S(\mathbf{A}^T\mathbf{W}), \mathbf{y}) = -\sum_{j=0}^{K} y_j \log(S(\mathbf{A}^T\mathbf{W}^j))$ is CrossEntropy Loss, where y_j is the true value of class j. Notice that $\mathbf{y}$ is encoded in one-hot, so only one value is 1 and the rest are 0 in $\mathbf{y}$. Therefor, when using Eq. 3 in the submission (Line 423) to train the model, we get

$$\begin{aligned}
L &= -\log\left\{\frac{\exp\left[\alpha\mathbf{A}^T\mathbf{W}^y + (1-\alpha)\,\mathbf{A}^T\mathbf{W}^{y'}\right]}{\sum_{j=0}^{K}\exp\left(\mathbf{A}^T\mathbf{W}^j\right)}\right\} \\
&= -\left[\alpha\mathbf{A}^T\mathbf{W}^y + (1-\alpha)\,\mathbf{A}^T\mathbf{W}^{y'}\right] + \log\left[\sum_{j=0}^{K}\exp\left(\mathbf{A}^T\mathbf{W}^j\right)\right] \\
&= -\alpha\left(\mathbf{A}^T\mathbf{W}^y - \mathbf{A}^T\mathbf{W}^{y'}\right) + \log\left[\sum_{j=0}^{K}\exp\left(\mathbf{A}^T\mathbf{W}^j\right)\right] - A^T W^{y'} \\
&= -\alpha\left(\mathbf{A}^T\mathbf{W}^y - \mathbf{A}^T\mathbf{W}^{y'}\right) + \log\left[\frac{\sum_{j=0}^{K}\exp\left(\mathbf{A}^T\mathbf{W}^j\right)}{\exp\left(\mathbf{A}^T\mathbf{W}^{y'}\right)}\right].
\end{aligned}$$

Let $X = \mathbf{A}^T\mathbf{W}^y - \mathbf{A}^T\mathbf{W}^{y'}$, we can get

$$L = -\alpha X + \log\left[1 + \sum_{j\neq y, j\neq y'}^{K-2}\exp\left(\mathbf{A}^T\mathbf{W}^j - \mathbf{A}^T\mathbf{W}^{y'}\right) + \exp(X)\right].$$

Let $1 + \sum_{j\neq y, j\neq y'}^{K-2}\exp\left(\mathbf{A}^T\mathbf{W}^j - \mathbf{A}^T\mathbf{W}^{y'}\right) = c$, the gradient of L w.r.t X is

$$\nabla_X L = -\alpha + \frac{\exp(X)}{c + \exp(X)}.$$

Suppose $F(\cdot)$ is an ideal robust classifier, $L = 0$. The gradient of L w.r.t X is $\nabla_X L = 0$. We can get the equation as

$$\begin{aligned}
&-\alpha + \frac{\exp(X)}{c + \exp(X)} = 0 \\
&\Longrightarrow \exp(X) = \alpha c + \alpha\exp(X) \\
&\Longrightarrow X = \log\frac{\alpha c}{1-\alpha} \\
&\Longrightarrow \mathbf{A}^T\mathbf{W}^y - \mathbf{A}^T\mathbf{W}^{y'} = \\
&\log\left\{\frac{\alpha\left[1+\sum_{j\neq y, j\neq y'}^{K-2} exp\left(\mathbf{A}^T\mathbf{W}^j - \mathbf{A}^T\mathbf{W}^{y'}\right)\right]}{1-\alpha}\right\}.
\end{aligned}$$

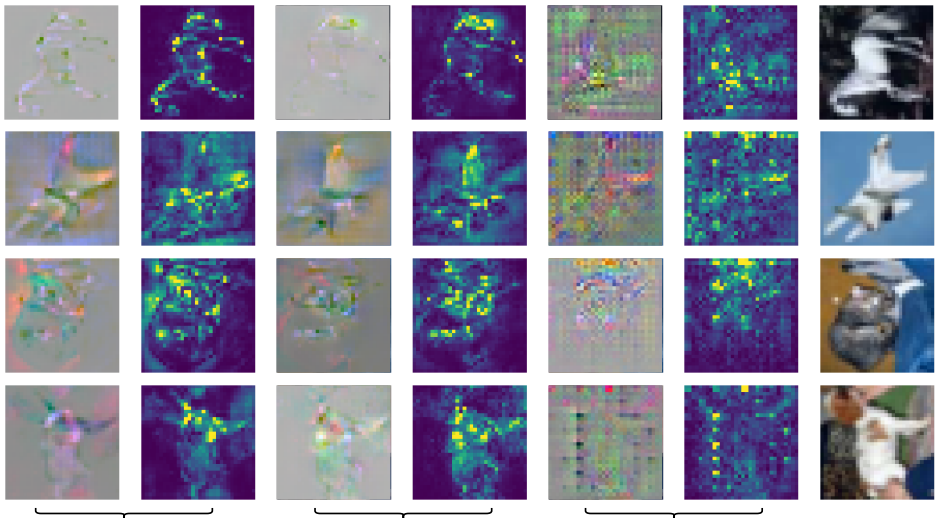


Figure 1. Use Smoothgrad to visualize the features learned by the model.

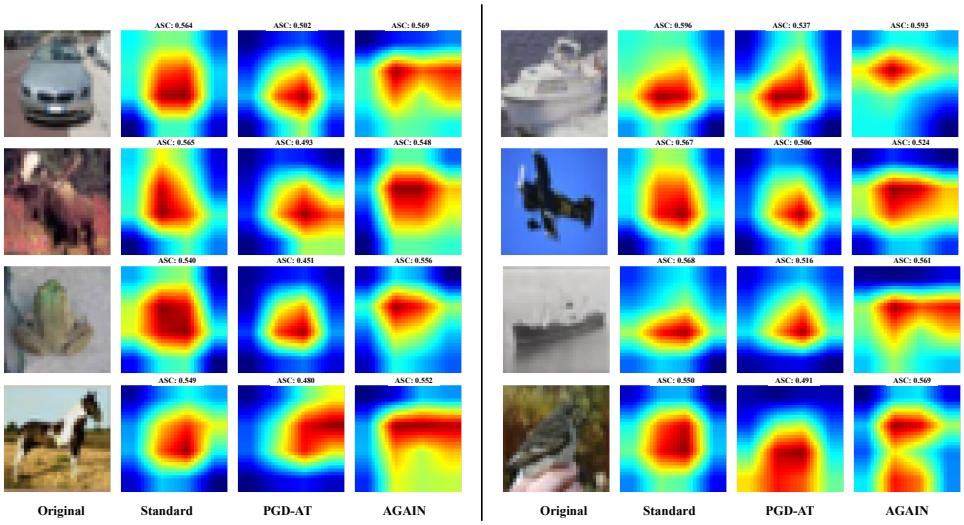


Figure 2. A visual illustration of attribution span.

Table 1. The average ASC values of ResNet-18 under CIFAR-10 dataset with different training methods.

Dataset	Training method	Average ASC
CIFAR-10	Standard	54.86%
	PGD-AT	51.80%
	AGAIN	53.54%

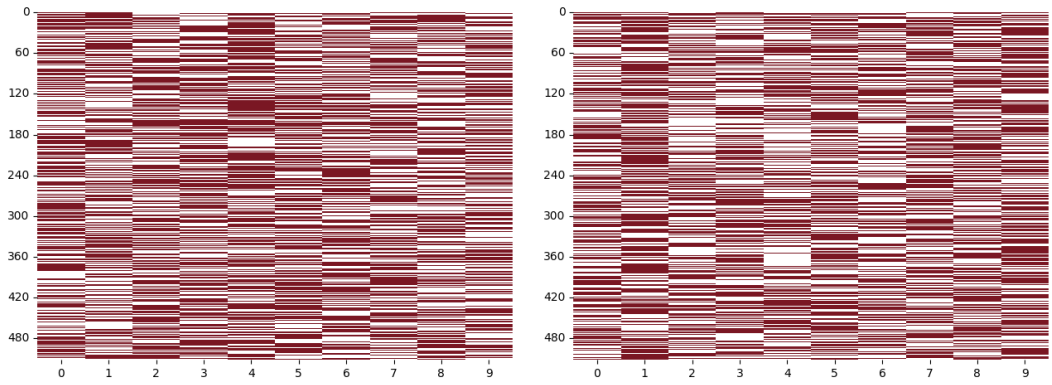
4. More Experimental Result

4.1. Hyper-parameter Selection.

The hyperparameter α controls the ratio between true labels and fake labels. The larger α is, the more attention the model pays to the features corresponding to true label. When $\alpha = 1$, only the features related to the true label are enhanced; The smaller α is, the more attention the model pays to the corresponding features of fake label, and when $\alpha = 0$, only the corresponding features of fake label are enhanced. We chose different parameters α and conducted experiments using ResNet-18 [2] on the CIFAR-10 dataset [5] to determine the best hyperparameter α. The experimental results are shown in Figure 4. When $\alpha = 0$ or $\alpha = 1$, it means that the model focuses only on the feature span under fake labels and those under true labels, respectively. From the experimental results, we can see that when $\alpha = 0.6$ is the best result. Therefore, we set α to 0.6 during experi-

Table 2. The number of valid values in the parameter weights $\mathbf{W}$ of the fully connected layer.

method \ class	0	1	2	3	4	5	6	7	8	9	Average
w/ AGAIN	302	268	332	319	330	326	318	302	296	287	308
w/o AGAIN	269	302	236	265	259	274	258	274	288	298	272.3



(a) AT w/ AGAIN (b) AT w/o AGAIN

Figure 3. Visualization of the weights of the fully connected layers (where red indicates that the value at this position is a valid value, and white indicates an invalid value).

ments to ensure that the feature span corresponding to the true labels dominate, while also allowing the model to learn other feature span.

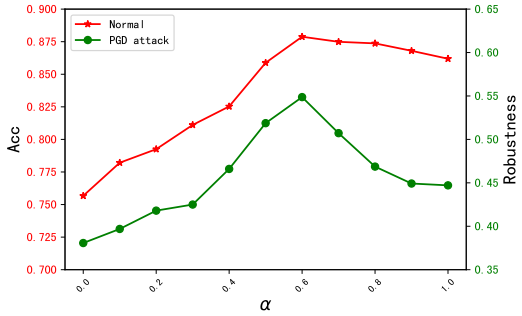


Figure 4. Effect of hyperparameter α on experimental results.

4.2. Impact of HFF on the Model

To verify the effect of HFF on the model, we visualize the inference process using Smooth Grad [6], and the experimental results are shown in Figure 1. From the results, it can be seen that the model trained with the proposed method is able to pay more attention to the structural information of the data than the model under standard training and the robust model without HFF (W/O HFF), which improves the ability to resist the adversarial examples. The experimental results verify the effectiveness of our proposed method.

4.3. Analysis of the Model's Attribution Span

We train the ResNet-18 model on the CIFAR-10 dataset using the standard training, PGD-AT and the proposed method, respectively. We visualize the attribution span for each of the three models and calculated the ASC values. The experimental results are shown in Figure 2 and Table 1. From the experimental results, it can be seen that our method effectively enlarges the attribution region of the robust model. This also verifies our proposed hypothesis.

4.4. Number of Valid Values in Weights

We train the ResNet-18 model using the proposed method and the traditional AT, respectively, and extract the weight matrix $\mathbf{W} \in \mathbb{R}^{512\times10}$ in the linear classification layer. When the value of an element in $\mathbf{W}$ is greater than the mean, we consider the value to be a valid value. We visualize the final result, and the experimental result is shown in Figure 3 and Table 2. From the experimental results, it can be seen that most of the classes have more valid values for the weights when trained with the proposed method compared to the traditional AT. The model trained with the proposed method has an average of 308 valid values in the weights of each class. The model trained using traditional AT has an average of 272.3 valid values in the weights of each class. This indicates that the model trained using the proposed method uses more feature values in the inference process, thus expanding the attribution span of the model. This verifies the correctness of our proposed method.

References

[1] 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. 1

[2] 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

[3] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 558–567, 2019. 2

[4] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019. 1

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

[6] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 4

[7] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 1

[8] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 2

[9] Dawei Zhou, Nannan Wang, Chunlei Peng, Xinbo Gao, Xiaoyu Wang, Jun Yu, and Tongliang Liu. Removing adversarial noise in class activation feature space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7878–7887, 2021. 1