ATTENTION-BASED INTERPRETATION AND RESPONSE TO THE TRADE-OFF OF ADVERSARIAL TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

To boost the robustness of a model against adversarial examples, adversarial training has been regarded as a benchmark method. However, it is commonly considered to be easily suffering from the trade-off dilemma between robustness and generalization in practice. This paper tries to make an intuitive explanation for this phenomenon in the perspective of model attention and provides an attention expansion viewpoint to learn a reliable model. To be specific, we argue that adversarial training does enable one model to concentrate on exact semantic information of input, which is beneficial to avoid adversarial accumulation. But it also easily make the model to cover fewer spatial region so that the model usually ignores some inherent features of the input. This may be one main reason to result in weak generalization on unseen inputs. To address this issue, we propose an *Attention-Extended Learning Framework (AELF)* built on the cascade structure of deep models. AELF advocates that clean high-level features (from natural inputs) are used to guide the robustness learning rather than hand-crafted labels, so as to ensure broad spatial attention of model to input space. In addition, we provide a very simple solution to implement AELF under the efficient softmax-based training manner, which avoids checking the difference between high-dimensional embedding vectors via additional regularization loss. Experimental observations verify the rationality of our interpretation, and remarkable improvements on multiple datasets also demonstrate the superiority of AELF.

1 INTRODUCTION

Deep learning has achieved great success in many applications. However, its vulnerability to adversarial examples has recently attracted wide attention (Szegedy et al., 2014; Ilyas et al., 2019; Bai et al., 2021) in many security-sensitive scenarios. The studies in adversarial learning can be simply categorized into adversarial attack and model defence. To mislead the network's decision, the attacks focus on how to produce imperceptible or deliberately crafted perturbations for natural inputs. In contrast, model defence aims to eliminate the misleading caused by adversarial perturbations. To make an effective model defence, this paper focuses on learning a reliable model.

For model defence, a variety of attempts have been made from different perspectives: 1) Adversarial Training (AT). Empirical AT methods (Goodfellow et al., 2014; Kurakin et al., 2017) directly use augmented adversarial examples to enhance the robustness of a model. To provide provable robustness guarantees, certified AT methods (Wong & Kolter, 2018; Mirman et al., 2018) attempt to optimize a model based on the upper bound of norm-bounded perturbations. 2) Model modification. To eliminate the gradient source of optimization-based attacks, gradient masking methods such as defensive distillation (Papernot et al., 2016) and thermometer encoding (Buckman et al., 2018) modify a model to produce useless gradient information so that it is difficult to directly construct adversarial examples. 3) Input transformation methods (Guo et al., 2018; Naseer et al., 2020) are devoted to removing the perturbations from adversarial inputs, to indirectly mitigate attacks.

Overall, as one of the most effective defence techniques, empirical AT directly focuses on learning a robust model and has become the standard practice owing to its convenience. However, augmented adversarial examples introduce a minimax problem that leads to difficulties in convergence. Besides the time-consuming cost (Shafahi et al., 2019; Wong et al., 2020), the other typical drawback is the

Table 1: Architectures of one basic CNN model (CNNB).

Type\Layer	$1th$	$2th$	$3th$	$4th$	$5th$	$6th$
Conv(3×3)	64	64	128	256	512	—
FC	—	—	—	—	—	512×64
Batch Normlization	Yes	Yes	Yes	Yes	Yes	—
Activation (LeakyReLU)	Yes	Yes	Yes	Yes	Yes	—
Pooling (Average)	—	2×2	2×2	2×2	4×4	—

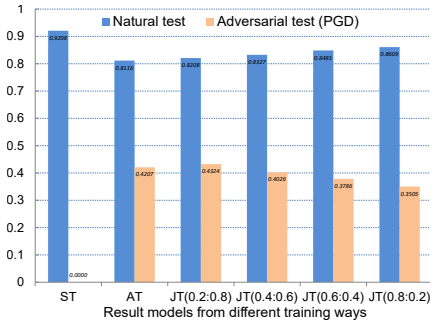


Figure 1: The trade-off phenomenon of CNNB on CIFAR-10. Table.1 details one basic CNNB. For the CNNB, we conduct ST, typical AT and JT (Joint AT with equivalent natural and adversarial inputs). Different trained models are listed on the horizontal axis, where the mark $*:*$ denotes the loss ratio between two kinds of data during training. The vertical axis shows their test accuracy. The adversarial examples used for the training and test sets are uniformly produced by PGD$(0.031, 5, 0.01)$[2]. Under the same training setting detailed in section 5.1, the histogram of test results reflects a clear trade-off tendency between generalization and robustness [1].

trade-off phenomenon[1] (Tsipras et al., 2019; Mehrabi et al., 2021), where the augmented examples usually cause performance degradation for natural inputs. Intuitively, one practical verification has been given in Fig. 1. This trade-off issue has received some attention (Zhang et al., 2019; Aditi et al., 2020; Yang et al., 2020) recently. This paper attempts to provide a more intuitive study from the perspective of model attention to the input space and gives one simple solution to alleviate it.

Previous interpretations usually treat a deep model as one entire mapping function from input to label. The separability of input data (Tsipras et al., 2019; Yang et al., 2020) plays a key role to search for a smooth decision boundary such as TRADES (Zhang et al., 2019). In this paper, we argue that the high-dimensional inputs are usually inseparable in the original data space. In fact, the architecture of deep models provides one major promotion to achieve impressive success. As Grad-CAM (Selvaraju et al., 2020) implied, CNN performs as a filter function to the input image so that the interesting areas can be highlighted. In this case, the identification of the label is related to some selective spatial features rather than the entire input. Herein, a better way to explain the trade-off issue is to observe the attention difference between Standard Training (ST) and AT models.

To investigate the trade-off issue, we use gradient-based observations to check the model's attention. As shown in Fig. 2, AT enables a model to capture more discriminative pixel-level features, which are semantically consistent with humans (Ilyas et al., 2019). This situation is the same as (Tsipras et al., 2019). However, we argue that AT may also promote the model to ignore broader spatial regions thus disregard some inherent features of the input more easily. In fact, the learned model from ST covers broader regions, which has given some intuition to testify our argument. Besides, one verification on the shifted MNIST test sets also provides strong supports for this judgment.

To address the trade-off issue, it is practical to conduct Joint network Training (JT) with natural and adversarial examples, such as BIM (Kurakin et al., 2017). However, as shown in Fig. 1, the simple regularization with natural data improves generalization but hurts robustness. In this paper, we argue that the label is only a hand-crafted scalar without any prior knowledge. The global check from adversarial inputs to labels may not gain enough smooth mapping from the input space to label space. Inspired from our insight on trade-off, to enhance the robustness, the key should be ensuring broad spatial attention of the model to the input space. Here, we advocate to append mapping learning from adversarial input to clean high-level features of natural inputs rather than labels. This is mainly derived from the following discussion. Since modern standard models have achieved SOTA performance due to their broad spatial attention to inputs, the high-level representations of natural inputs should be richer semantic integrations of broad input spaces. In this case, instead of hand-crafted labels, this rich feature representation should be able to provide an effective constrain to ensure the model's attention to a broad spatial region of an input.

[1] In this paper, the term 'generalization' means the discrimination capability of a model to unseen natural examples. It focuses on the model's adaption to various appearances of inherent features. The term 'robustness' implies the ability in dealing with adversarial examples with extrinsic noise or perturbation.

[2] PGD$(\epsilon, i, \epsilon_i)$: ϵ is the limited radius of perturbation based on l_∞-norm, the final perturbation is obtained from i iterations with once perturbation scale ϵ_i according to (Madry et al., 2018).

For a more detailed introduction to related work, please refer to Appendix A. The main contributions of our work include:

- Attention-based explanation to the trade-off issue. We provide an intuitive insight for the trade-off issue via an attention observation of the model to input space. To avoid the misleading of adversarial perturbations. AT enables a model to capture some semantic pixel-level features. This sparse spatial region is beneficial to avoid adversarial accumulation for robustness. But this also yields that AT unrestrainedly forces the model to ignore some discriminative features of the input so as to further avoid stronger adversarial accumulation. This intuitive judgment inspires us to ensure the spatial attention of the model during AT.
- Attention-Extended Learning Framework (AELF). Given the cascaded structure of a neural network, AELF does not treat a model as an integral function to construct a mapping between input and label. To enhance the robustness, it recommends directly appends mapping learning from adversarial inputs to clean embeddings of natural inputs, so as to conduct an attention expansion. Note that AELF is only a unidirectional mapping learning that avoids misleading the learning from natural inputs to adversarial embeddings. In this case, we treat perturbations as normal noises and train a model to ignore them.
- Simplified implementation of AELF (AELFs). Based on the back-propagation algorithm, AELFs provides a clever solution to AELF under a traditional training manner. It achieves attention expansion with a single softmax-based loss. This avoids the difficult optimization to deal with embedding learning using additional losses. Given its simplicity, it is easy to implement for model defence with a more efficient way.

2 Exploration to the Trade-off Phenomenon

In this section, we briefly review model training about generalization and robustness. Further, we give an explanation to the trade-off issue from the perspective of the model's attention.

2.1 Review of generalization and robustness

Given an underlying joint distribution $\mathbb{D}$ with inputs $\mathbf{x} \in \mathcal{X}$ and their labels $y \in \mathcal{Y}$, the classification task aims to learn a predictor $\mathcal{M}_\theta$ to achieve the mapping $\mathcal{X} \overset{\theta}{\mapsto} \mathcal{Y}$. In practice, we usually learn θ by accessing a labeled training set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$. ST minimizes its empirical risk using

$$\arg\min_\theta \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{S}}[\mathcal{L}(\theta; \mathbf{x}, y)]\,, \tag{1}$$

where $\mathcal{L}$ denotes a suitable loss function. Obviously, this conventional optimization only focuses on the standard generalization ability of the predictor $\mathcal{M}_\theta$ for unknown inputs $\mathbf{x} \notin \mathcal{S}$.

However, the model, $\mathcal{M}_\theta$, produced by ST is usually vulnerable to adversarial examples. In this case, given a natural input $\mathbf{x}$, we can modify it as adversarial $\mathbf{x}'$ to mislead $\mathcal{M}_\theta$.

$$\mathbf{x}' = \mathbf{x} + \delta, \text{ subject to } ||\delta||_p \leq \epsilon, \tag{2}$$

where, the perturbation δ provides adversarial information under the constraint of the l_p-norm boundary. To characterize δ, FGSM has been one of the most direct and effective way. Further, BIM and PGD use its iterative version to generate more powerful $\mathbf{x}'$.

To improve the robustness of $\mathcal{M}_\theta$ against $\mathbf{x}'$, AT directly use augmented $\mathbf{x}'$ to empirically boost the mapping ability of $\mathcal{M}_\theta$. The training manner is formalized as the following minimax problem.

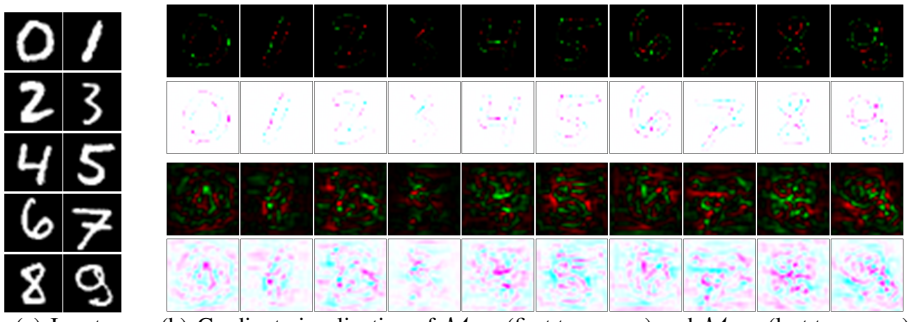
$$\arg\min_\theta \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{S}}[\max_{\delta\in\Delta} \mathcal{L}(\theta; \mathbf{x}+\delta, y)]\,, \tag{3}$$

where Δ=$\{||\delta||_p \leq \epsilon\}$ is the perturbation set, which covers the l_p ball with radius ϵ. Incompletely, it only focuses on the robustness but ignore generalization so as to yield the trade-off issue. To promote both, it is natural to seek a moderation θ using $\mathbf{x}$ and $\mathbf{x}+\delta$. Therefore, the JT framework solves the combination of the above two problems.

$$\arg\min_\theta \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{S}}[\mathcal{L}(\theta; \mathbf{x}, y) + \lambda \max_{\delta\in\Delta} \mathcal{L}(\theta; \mathbf{x}+\delta, y)]\,. \tag{4}$$

Actually, as the observation to data separation (Yang et al., 2020) demonstrated, flexible label-level outputs in TRADES can provide more smooth regularization to robustness.

$$\arg\min_\theta \mathbb{E}\{\mathcal{L}[\mathcal{M}_\theta(\mathbf{x}), y] + \lambda\mathcal{L}^*[\mathcal{M}_\theta(\mathbf{x}), \mathcal{M}_\theta(\mathbf{x}')]\}\,, \tag{5}$$



(a) Inputs (b) Gradient visualization of $\mathcal{M}_{\mathcal{ST}}$ (first two rows) and $\mathcal{M}_{\mathcal{AT}}$ (last two rows)

Figure 2: Exhibition of normalized gradient[3] on inputs with respect to loss. (a) The natural input images; (b) Gradients on inputs from $\mathcal{M}_{\mathcal{ST}}$ and $\mathcal{M}_{\mathcal{AT}}$. Overall, the color depth reflects the magnitude of the gradient. Small gradient values present appearances close to the background. To make more clear observations, we separately use a white and black background to show them. In detail, red denotes positive gradient and green denotes negative gradient on black background. The reverse marks are shown on white background. Overall, the most intuitive finding is that $\mathcal{M}_{\mathcal{AT}}$ captures high-frequency shape information of character, which is semantically consistent with humans but covers a fewer area of original input than $\mathcal{M}_{\mathcal{ST}}$.

where $\mathcal{L}^*$ is the KL-divergence loss. This global regularization provides more guarantee for the upper bound of the gap between robust error and natural error. Besides, turn to data augmentation, RLFAT (Song et al., 2020) and RST (Aditi et al., 2020) have demonstrated that local features of $\mathbf{x}'$ and extra unlabeled data also can effectively improve this trade-off situation.

2.2 Attention-based exploration for the trade-off phenomenon

As we all know, for a natural input $\mathbf{x}$, to generate its adversarial version $\mathbf{x}'$, the loss-based gradient sign has been one baseline to characterize an effective perturbation. Further, Grad-CAM also has indicated that gradient information is a good source to understand the internal mechanism of deep models. This motivates us to provides an intuitive explanation to the trade-off issue via the observation of gradient on inputs. Overall, to study the internal mechanism of the trade-off issue, we start from the interesting area of a pre-trained model to the pixel-level features of natural inputs.

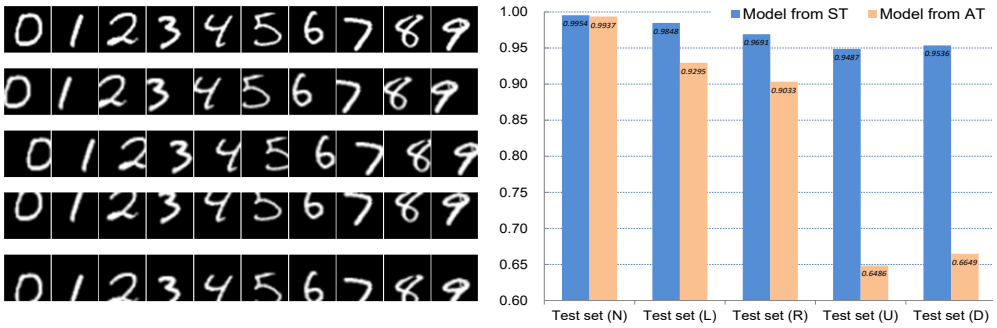
In detail, we separately train one CNN model with ST and AT, and check its attention to test inputs through observing the gradient values on inputs. To obtain reasonable observations, one fair preparation is needed. 1) For model, a basic CNN$_{\text{B}}$ has been defined in Table 1. 2) For the data set, two keys are that there is only a little trade-off between ST and AT and the gradient visualizations on inputs are intuitive for humans, so we choose the MNIST data set. 3) For training, we use the popular cross-entropy loss. Following (Madry et al., 2018), we separately conduct ST with Eq. (1) and AT with Eq. (3). Especially, for AT, the augmented examples are produced with PGD$(0.3, 16, 0.02)^2$. A more detailed description of the training settings is provided in Appendix C.

After sufficient training with 50 epochs, we obtain standard $\mathcal{M}_{\mathcal{ST}}$ and adversarial $\mathcal{M}_{\mathcal{AT}}$. Their baseline test results are reported in Table 2. Here, we check their attention to the spatial region of a natural input via gradient visualization[3], as shown in Fig.2. Intuitively, the gradient images show a similar situation with the visualization from (Tsipras et al., 2019). For example, the gradients from $\mathcal{M}_{\mathcal{AT}}$ describe the specific outline of a character so that they are significantly more interpretable than the gradients from $\mathcal{M}_{\mathcal{ST}}$. Here, we briefly discuss the trade-off problem as follows.

As we known, the attention mechanism (Jaderberg et al., 2016; Hu et al., 2019) of modern models plays a key activation role to capture effective input-level features from high-dimensional data. As Grad-CAM revealed, given a 2-D image $\mathbf{x}$ to the pre-trained $\mathcal{M}$, the magnitude of gradient $G^{\mathbf{x}}_{i,j}$ highlights the importance of spatial location $\mathbf{x}_{i,j}$ to the label y. In other words, to identify an input $\mathbf{x}$, its label only selectively focuses on part inherent features of $\mathbf{x}$. In this case, the identification process of our both models can be reinterpreted as

$$\begin{cases} y = \mathcal{M}_{\mathcal{ST}}(\mathbf{x}) \Rightarrow \mathbf{x} \rightarrow \mathbf{x}_{\mathcal{S}} \rightarrow y \\ y = \mathcal{M}_{\mathcal{AT}}(\mathbf{x}) \Rightarrow \mathbf{x} \rightarrow \mathbf{x}_{\mathcal{A}} \rightarrow y \end{cases}, \tag{6}$$

[3] Exhibition of gradient information: Given a 2-D input image $\mathbf{x}$, its gradient value $G^{\mathbf{x}}$ is firstly normalized as $\widetilde{G}^{\mathbf{x}}$=$G^{\mathbf{x}}/\|G^{\mathbf{x}}\|_2$. To give a clearer display, we show $\bar{G}^{\mathbf{x}}_*$=$\widetilde{G}^{\mathbf{x}}/\max\{|\widetilde{G}^{\mathbf{x}}|\}$.



(a) Natural images and their transformations (b) Results of $\mathcal{M}_{\mathcal{ST}}$ and $\mathcal{M}_{\mathcal{AT}}$ on different test sets

Figure 3: (a) Some natural images and their transformation examples in the test set. The first row shows some natural images. The nether four rows separately show result examples via left (L), right (R), up (U) and down (D) shifts. (b) Comparison results between $\mathcal{M}_{\mathcal{ST}}$ and $\mathcal{M}_{\mathcal{AT}}$ on different test sets. We apply the 4-direction shifts transformation on natural test set, and generate 4 new test set (U, L, R and D). The right sub-figure shows test results of both models. The test set (N) list their baseline accuracy on natural test set. The last 4 bar charts check their generalization on 4 shift sets. As the verification showed, under almost equal ability on natural set (N), $\mathcal{M}_{\mathcal{AT}}$ performs worse adaptation to shifts of characters than $\mathcal{M}_{\mathcal{ST}}$, especially for the up and down shifts.

where, $\mathcal{S}$ and $\mathcal{A}$ separately denote the outstanding attentions of $\mathcal{M}_{\mathcal{ST}}$ and $\mathcal{M}_{\mathcal{AT}}$ to $\mathbf{x}$. As shown in Fig. 2, it is obvious that $\mathbf{x}_{\mathcal{A}}$ covers better semantic information. However, abundant studies indicate that $\mathcal{M}_{\mathcal{AT}}$ achieves worse generalization than $\mathcal{M}_{\mathcal{ST}}$. This counter-intuitive situation reminds whether $\mathcal{M}_{\mathcal{AT}}$ falls into an over-fitting state. Actually, turn to check the robustness, we find that $\mathcal{S} > \mathcal{A}$ ($\mathcal{S}$ covers broader foreground and background). This naturally makes sense that it is easier to increase loss with gradient attack for $\mathcal{M}_{\mathcal{ST}}$ due to broader accumulation of effective perturbations. Meanwhile, the smaller size of $\mathbf{x}_{\mathcal{A}}$ is beneficial to avoid the perturbation accumulation. Therefore, one natural inference to the trade-off issue can be drawn as follows.

Inference: To avoid broad perturbation accumulation, AT trends to produce a model which uses few inherent features to achieve mapping learning from training inputs to labels. Moderately this enables the model to capture some exact semantic information. But the continuous internal maximization of Eq. (3) can force the model to overly ignore some discriminative features. This yields that AT model usually concentrates on more sparse spatial regions and ignores relatively global checks to input space. This naturally reduces the model's adaptability to unseen changes of inputs.

To verify the above inference, we apply different shift transformations on the MNIST test set to check the well-known shift-invariant of CNN. For the original set, we generate 4 direction-based shift sets to test them. Fig. 3 shows some examples with $8/32$ pixel shift on 4 directions and lists test results on different shift sets. Obviously, $\mathcal{M}_{\mathcal{ST}}$ shows stronger adaptation to various appearances of the test images. It is noteworthy that we don't use any data argumentation (such as random cropping) during both trainings and there are without remarkable trade-off situations on the original test set.

An acknowledgement can be reached from the above investigation. AT usually promote one model to achieve mapping learning via capturing fewer pixel-level features, so that the model ignores some discriminative information of inputs. This is certainly adverse to deal with potential variations of inputs. Actually, this over-fitting judgment is also indirectly supported with the effectiveness of local features in (Song et al., 2020) and dropout improvement in (Yang et al., 2020). Empirically, both operations can prevent the model from overly focus on the local region of input during training.

3 Attention-Extended Learning View for AT

As discussed in Section 2, AT easily compels the model to ignore part spatial region of an input so that it hurts the standard generalization. In general, the claimed joint learning $\mathbf{x}, \mathbf{x}' \mapsto y$ of JT can improve this issue. However, as Fig. 1 shown, the simple regularization with normal $\mathbf{x}$ also redraws a new trade-off to hurt the adversarial robustness.

3.1 Attention-extended Learning Framework(AELF)

In this section, we attempt to achieve a reinforced AT. Its definition is clear to enhance the model's robustness to adversarial data under the guarantee of its discrimination for natural data. Inspired

Algorithm 1 Attention-Extended Learning Framework

Initialize: $\mathcal{M}(\ , c, w)$, $\mathcal{L}_1, \mathcal{L}_2$ and hyperparameter λ
Input: Natural training set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}$
Output: c and w

1: **for** each mini-batch set **do**
2: Randomly select example set: $\{\mathbf{x}, y\} \leftarrow \mathcal{S}$
3: Generate adversarial examples: $\mathbf{x}' = \mathbf{x} + \delta$
4: Forward Compute: $\mathbf{x} \xrightarrow{c} \mathbf{f} \xrightarrow{w} \hat{y}$;
5: Compute loss $\mathcal{L}_1$: $\mathcal{L}_1(\hat{y},\ y)$
6: Update parameter w: $w = w - \frac{\partial \mathcal{L}_1}{\partial w}$
7: Forward Compute: $\mathbf{x}' \xrightarrow{c} \mathbf{f}'$
8: Compute loss $\mathcal{L}_2$: $\mathcal{L}_2(\hat{\mathbf{f}}',\ \bar{\mathbf{f}})$, where $\bar{\mathbf{f}} \leftarrow \mathbf{f}$
9: Update parameter c: $c = c - \frac{\partial(\mathcal{L}_1 + \lambda\mathcal{L}_2)}{\partial c}$
10: **end for**

We summarize an alternating parameter update strategy. One key is that feature $\bar{\mathbf{f}}$ is treated as constant vector taken from $\mathbf{f}$ in each iteration.

Algorithm 2 Simplified implementation to AELF

Initialize: $\mathcal{M}(\ , c, w)$, $\mathcal{L}$ and hyperparameter λ
Input: Natural training set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}$
Output: c and w

1: Initialize c and w with standard training on $\mathcal{S}$
2: **for** each mini-batch set **do**
3: Randomly select example set: $\{\mathbf{x}, y\} \leftarrow \mathcal{S}$
4: Generate adversarial examples: $\mathbf{x}' = \mathbf{x} + \delta$
5: Forward Compute: $\mathbf{x}, \mathbf{x}' \xrightarrow{c} \mathbf{f}, \mathbf{f}' \xrightarrow{w} \hat{y}, \hat{y}'$
6: Compute $\mathcal{L}$: $\mathcal{L}_{\mathbf{x}}(\hat{y},\ y)$, $\mathcal{L}_{\mathbf{x}'}(\hat{y}',\ y)$
7: Update parameter:
$c = c - \alpha \frac{\partial(\mathcal{L}_{\mathbf{x}} + \lambda\mathcal{L}_{\mathbf{x}'})}{\partial c}$
8: **end for**

The w is not updated during AELFs. α is the learning rate. In general, AELFs does not suggest using one network with ReLU activation, which sometimes causes some parameters to be locked after the initialization of standard training.

from our insight to trade-off, the main idea is to conduct one JT without reducing the model's attention to input space. As we all know, the label y is only a man-made scalar, which cannot provide any additional information to ensure broad spatial attention of the model. In this paper, $\mathcal{M}$ is no longer simply treated as an entire mapping function like Eq. (4) and Eq. (5). Turn to consider the cascaded structure of $\mathcal{M}$, we further divided both mapping of Eq. (6) into

$$\begin{cases} \mathbf{x} \mapsto y \Rightarrow \mathbf{x} \rightarrow \mathbf{x}_{\mathcal{S}} \rightarrow \mathbf{f}_{\mathbf{x}} \rightarrow y \\ \mathbf{x}' \mapsto y \Rightarrow \mathbf{x}' \rightarrow \mathbf{x}'_{\mathcal{A}} \rightarrow \mathbf{f}'_{\mathbf{x}} \rightarrow y \end{cases}, \tag{7}$$

where $\mathbf{f}_{\mathbf{x}}, \mathbf{f}_{\mathbf{x}}{}'$ separately denote high-level features from $\mathcal{M}$. As reminded in Section 2.2, since ST produces $\mathcal{M}_{\mathcal{ST}}$ that captures broader region of the input $\mathbf{x}$. $\mathbf{f}_{\mathbf{x}}$ should be a richer semantic integration to pixel-level features. In this case, $\mathbf{f}_{\mathbf{x}}$ provides a good constrain to carry out $\mathbf{x}, \mathbf{x}' \rightarrow \mathbf{f}_{\mathbf{x}} \rightarrow y$ rather than direct $\mathbf{x}, \mathbf{x}' \mapsto y$. Here, we use feature embedding to conduct the constrain.

Taking CNN model as an example, $\mathcal{M}_\theta$ can be divided into the feature extraction module $\mathcal{M}_c$ and label mapping module $\mathcal{M}_w$. In this view, the mapping learning $\mathbf{x} \overset{\theta}{\mapsto} y$ can be recognized as $\mathbf{x} \xrightarrow{c} \mathbf{f}_{\mathbf{x}} \xrightarrow{w} y$, where $\mathbf{f}_{\mathbf{x}}$ is feature embedding derived from natural $\mathbf{x}$. To boost the robustness, we suggest to append fitting learning $\mathbf{x}' \overset{c}{\mapsto} \mathbf{f}_{\mathbf{x}}$. This optimization problem can be simply formalized as

$$\arg\min_{c,w} \mathbb{E}\big\{\mathcal{L}_1\{\mathcal{M}_w[\mathcal{M}_c(\mathbf{x})], y\} + \lambda\mathcal{L}_2\{\mathcal{M}_c(\mathbf{x}'), \mathcal{M}_c(\mathbf{x})\}\big\} \tag{8}$$

where $\mathcal{M}_c(\mathbf{x})$ and $\mathcal{M}_c(\mathbf{x}')$ separately produce inherent $\mathbf{f}_{\mathbf{x}}$ and adversarial $\mathbf{f}_{\mathbf{x}}{}'$. $\mathcal{L}_1$ is one label loss, $\mathcal{L}_2$ is a loss to characterize the difference between $\mathbf{f}_{\mathbf{x}}$ and $\mathbf{f}_{\mathbf{x}}{}'$, such as commonly used $\|\mathbf{f}_{\mathbf{x}} - \mathbf{f}_{\mathbf{x}}{}'\|_{1,2}$. Noticeably, here $\mathcal{L}_2$ only checks the fitting $\mathcal{M}_c(\mathbf{x}') \rightarrow \mathcal{M}_c(\mathbf{x})$ rather than $\mathcal{M}_c(\mathbf{x}') \rightleftharpoons \mathcal{M}_c(\mathbf{x})$[4].

To make a clearer description, we outline the workflow of AELF in Algorithm 1. Clearly, 1) The parameter w with respect to label y is only updated with the backbone loss $\mathcal{L}_1$. This implies that only $\mathbf{f}_{\mathbf{x}}$ is learned to identify y. 2) To achieve feature extraction of both $\mathbf{x}$ and $\mathbf{x}'$, $\mathbf{f}_{\mathbf{x}}$ is used as a constant flag to guide the update of feature extraction module c.

Overall, to ensure generalization, attention view claims to enable the model to capture broader inherent features. Technically, we suggest learning a clean embedding distribution, in which features are only derived from natural inputs. This latent constrain to an embedding space implies: i) Adversarial perturbations are only treated as normal noises; ii) Model is trained to filter or ignore them.

3.2 Simplified implementation to AELF (AELFs)

AELF provides a clear process to conduct an attention expansion concept. But the claimed fitting $\mathbf{f}' \rightarrow \mathbf{f}$ is required to characterize the difference between high-dimensional embeddings $\mathbf{f}'$ and $\mathbf{f}$. The usual approach is to add a new loss $\mathcal{L}_2$. However the advised $\|\mathbf{f} - \mathbf{f}'\|_2$ provides different

[4] Eq. (8) cannot be directly used to optimize $\theta = \{c, w\}$. Because, the mapping $\mathbf{x}' \overset{c}{\mapsto} \mathbf{f}$ claims to unidirectional fitting $\mathbf{f}' \rightarrow \mathbf{f}$. But $\mathcal{L}_2[\mathcal{M}_c(\mathbf{x}'); \mathcal{M}_c(\mathbf{x})]$ actually checks bidirectional learning $\mathbf{f} \rightleftharpoons \mathbf{f}'$ like contrastive loss (Hadsell et al., 2006). This interactive way cannot avoid the misleading from perturbations due to $\mathbf{f} \rightarrow \mathbf{f}'$.

Table 2: Comparison results on MNIST with CNN$_B$, which is trained with total 50 epochs.

Method	Natural test					Adversaria test A$^*(0.3, 16, 0.02)$				
Test set	N	N(L)	N(R)	N(U)	N(D)	N	N(L)	N(R)	N(U)	N(D)
ST	99.54	98.48	96.91	94.87	95.36	0.0	0.0	0.0	0.0	0.0
AT	99.37	92.95	90.33	64.86	66.49	93.36	68.01	67.15	27.08	29.62
JT	99.38	92.50	91.32	69.10	65.19	92.72	65.32	68.77	27.03	29.97
AELFs	99.29	95.77	93.85	75.33	77.85	94.06	78.48	76.50	45.82	44.65

Table 3: Comparisons with CNN$_B$ on CIFAR-10.

Method	N	A*(5)	A*(10)	Note	Method (Ours)		N	A*(5)	A*(10)	Note
AT	81.16	42.07	41.39	AT(Madry)	ST		92.08	0.0	0.0	Ours
JT	83.27	40.26	39.60	Normal JT (Kurakin)	JT* β=0.5	*0.4:0.6*	83.63	43.60	42.99	JT with our two actions
	84.38	38.77	38.01			*0.5:0.5*	84.63	42.08	41.35	
	84.85	37.86	37.14			*0.6:0.4*	86.10	40.37	39.63	
TR$^-_{\lambda=1}$	83.04	40.28	39.66	TRADES (Zhang)	AELFs $\beta=0.5$	*0.4:0.6*	83.77	45.90	45.45	Implement of AELF
TR$^-_{\lambda=2}$	81.18	43.98	43.55			*0.5:0.5*	85.77	43.85	43.25	
TR$^-_{\lambda=3}$	79.14	44.93	44.63			*0.6:0.4*	86.75	40.90	40.07	

characteristics with softmax-based loss, resulting in the difficult tuning of λ and slow convergence due to its smoothness in the interval of small values.

To give a convenient implementation, we attempt to provide a simpler solution. Complying with conventional training manner, the main idea is to unify $\mathcal{L}_1$ and $\mathcal{L}_2$ as a single softmax-based $\mathcal{L}$

$$\arg\min_{c,w} \mathbb{E}\big\{\mathcal{L}_{\mathbf{x}}\{\mathcal{M}_w[\mathcal{M}_c(\mathbf{x})]; y\}+\lambda\mathcal{L}_{\mathbf{x}'}\{\mathcal{M}_w[\mathcal{M}_c(\mathbf{x}')]; y\}\big\} \tag{9}$$

where $\mathcal{L}_{\mathbf{x}}$ and $\mathcal{L}_{\mathbf{x}'}$ are from the single label-level loss $\mathcal{L}$. Overall, Eq. (9) is consistent with Eq. (4) of JT. But the separated description $\{c, w\}$ provides the chance for attention expansion. Following AELF in Algorithm 1, we provide a simplified AELFs, which is shown in Algorithm 2. Firstly, $\mathcal{L}_{\mathbf{x}}$ is used to conduct ST with the natural set $\mathcal{S}$. In this case, w is only related to inherent $\mathbf{f}$. To boost the robustness of $\mathcal{M}$, we directly conduct JT to update c. This simple operation is derived from our analysis for sofmax-based network, which has been discussed in Appendix B.1.

In practice, AELFs is very convenient to implement. It only needs to train the feature exaction module c in JT after a complete ST. In this paper, we minimize

$$\arg\min_{c,\bar{w}} \mathbb{E}[\lambda_1\mathcal{L}(c; \mathbf{x}, y)+\lambda_2\mathcal{L}(c; \mathbf{x}', y)+\beta(\|\mathbf{f}\|_2+\|\mathbf{f}'\|_2)] \tag{10}$$

where $\bar{w}$ denotes the pre-trained constant w that does not need to be updated here. To give a more intuitive description, the balance parameter λ is decomposed to λ_1 and λ_2 ($0 \leq \lambda_1, \lambda_2 \leq 1$ and $\lambda_1+\lambda_2=1$). Particularly, $\mathbf{f}$ and $\mathbf{f}'$are feature embeddings from $\mathcal{M}_c(\mathbf{x})$ and $\mathcal{M}_c(\mathbf{x}')$, their l_2-norm are simply appended to prevent over-fitting.

Note that tuning parameter β can be separately set to $\mathbf{f}$ and $\mathbf{f}'$ for better test performance in practice. In this paper, we only use a single β to verify the effectiveness of attention expansion. To give one rough range of β, we set λ_1, λ_2=0.5 and conduct a sensitivity analysis about β in Appendix B.2.

4 Experiments and Analysis

As revealed in a recent study (Pang et al., 2021), some training tricks can override the potential promotion of the proposed method. In this paper, we select a regular setting for all the experiments to verify the attention expansion viewpoint, which has been detailed in Appendix C.

4.1 Verification and insight on MNIST

In this subsection, to examine the mitigation of AELFs to trade-off, we compare AELFs with typical AT and JT on MNIST. As section 2.2 explored, although there are no obvious trade-off on the original test set, AT destroys strongly the shift-invariant of CNN. Here, we test them with CNN$_B$ on the four shift test sets described in Fig. 3. The training adapts the setting in Appendix C.

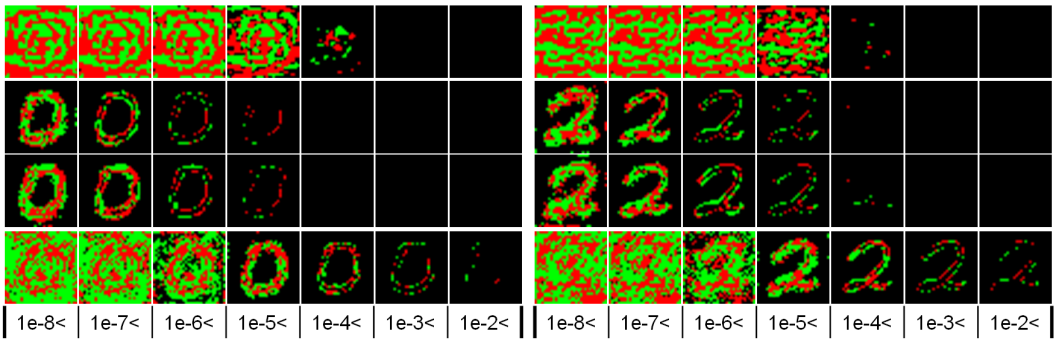


Figure 4: Exhibition of gradient signs on natural inputs. To observe attention difference of different models. We display their gradient signs[5] on test character '0' and '2' according to original magnitude of gradient. Corresponding to Table 2, the 4 rows separately show gradient signs from ST, AT, JT and AELFs. The bottom '$* <$' denotes that gradient signs are displayed only when its absolute value is bigger than $*$. The color distinguishes '+' and '-' of gradient. Overall, the big gradient values of AT and JT are very sparse. This implies that: To avoid the misleading of perturbation, AT and JT may be easier to ignore the some potential changes of inputs. In contrast, AELFs captures shape of character with bigger gradient values. Further, it covers broader spatial region of input space in the similar gradient interval with ST.

For adversarial data, all the training and test data are from PGD$(0.3, 16, 0.02)$. For AELFs, we simply fix λ_1, λ_2=0.5 and β=0.5. After 50 training epochs, we list all comparison results in Table.2.

Further, for different result models, we observe the difference of their gradient information by visualizing their gradient signs[5], which have been shown in Fig. 4.

To avoid that the setting of 50 epochs is a special case, we provide more verifications in Appendix D. Overall, two conclusions are reached. 1) For natural test, AELFs can better adapt to the 4 shift sets than JT and AT. For adversarial test, it also achieves remarkable improvements over other methods. 2) Insight to gradient, AT and JT ignore broad spatial region of inputs in terms of gradient value. This insight provides possible evidence for the improvements of AELFs on shift test sets.

4.2 Ablation study with CNN$_B$ on CIFAR-10

In this part, besides some regular comparisons with CNN$_B$ on CIFAR-10, we mainly check whether the attention expansion conception is valid. Actually, our implementation of AELFs contains two additional operations: parameter initialization with ST, and l_2-norm penalty to embedding features. So we improve the normal JT with the above additional actions to examine the effect of fixed $\bar{w}$.

Under an unified training setting in Appendix C, all test results of different models are listed in Table.3, where JT* is our improved version of JT. For adversarial examples, all augmented data is produced by PGD$(0.031, 5, 0.01)$ during training. Test data are seperately from PGD$(0.031, 5, 0.01)$ and PGD$(0.031, 10, 0.005)$, which are separately denoted as A$^*(5)$ and A$^*(10)$.

Overall, 1) Methods, JT performs the same situation with Fig.1, it is effective to improve natural test but negative to adversarial test. 2) The proposed two actions plays a positive influence to JT on all the tests. 3) AELFs achieves more outstanding accuracy with different examinations of λ. This indicates that fixed $\bar{w}$ is really useful to conduct the attention constrain. Besides, Note that AELFs hasn't achieved some surprising improvements like Mnist, this may be mainly derived from the small size of model and the using of data augmentation in training setting.

4.3 General comparisons with related works

Following the latest studies, we use wide residual networks WRN (Zagoruyko & Komodakis, 2016) to conduct two comparisons on CIFAR-10 and CIFAR-100. For different models, we separately use ReLU and LeakyReLU to examine our methods with the uniform training setting in Appendix C. For the comparisons with recent SOTA methods, we directly report their results to avoid that our training epochs are adverse for them.

[5] Exhibition of gradient sign: Initially, we directly visualize its gradient $\bar{G}^{\mathbf{x}}$ following Fig. 2, but find that $\bar{G}^{\mathbf{x}}_{\text{AELFs}}$ is similar with $\bar{G}^{\mathbf{x}}_{\text{AT}}$. Finally, we check their difference with gradient sign $\bar{G}^{\mathbf{x}}_{*}$ from original values $G^{\mathbf{x}}_{*}$.

Table 4: Comparisons with WRN-40 on CIFAR-10 (TR^- denotes TRADES method).

Method	Natural	$A^*(0.031)$	Note	Method	Natural	$A^*(0.031)$	Note
LLR	91.44	22.05	(Qin et al., 2019)	ST	95.33	0.11/0.10	ReLU (Our)
AT	83.51	43.51	(Madry et al., 2018)	ST*	95.71	0.16/0.15	LeakyReLU (Our)
$RST_{\lambda=0.5}$	85.11	39.58	JT with RST without	$TR^-_{\lambda=1}$	84.96	43.66	(Zhang et al., 2019)
$RST_{\lambda=1.0}$	84.61	40.89	extra unlabeled data	$TR^-_{\lambda=3}$	85.55	46.63	reported from
$RST_{\lambda=2.0}$	83.87	41.75	(Aditi et al., 2020)	$TR^-_{\lambda=6}$	84.46	48.58	(Yang et al., 2020)
$TR^-_{\lambda=3}$	86.43	49.01	Dropout improved	AELFs	90.42	55.39/55.46	$\lambda_1, \lambda_2 = 0.5$
$TR^-_{\lambda=6}$	84.69	52.32	by (Yang et al., 2020)	AELFs*	90.04	56.86/56.80	β=0.2

Table 5: Comparisons on CIFAR-100 (TR^- and RL^- separately denotes TRADES and RLFAT method).

Method	Natural	$A^*(0.03)$	Note(WRN-32)	Method	Natural	$A^*(0.03)$	Note(WRN-28)
AT	55.86	23.32	(Madry et al., 2018)	ST	80.27	0.14	ReLU (Our)
$TR^-_{\lambda=6}$	52.13	27.26	(Zhang et al., 2019)	ST*	80.20	0.11	LeakyReLU (Our)
RL^-+AT	56.70	31.99	(Song et al., 2020)	AELFs	65.08	26.69	$\lambda_1, \lambda_2 = 0.5$
RL^-+TR^-	58.96	31.63		AELFs*	65.17	27.11	β=0.2

1) Following (Yang et al., 2020), the first comparison is conducted with WRN-40 (dropout=0.2) on CIFAR-10. We train and test the model with PGD(0.031, 10, 0.0062). The results are reported in Table 4, where $*/*$ report two test rates separately from PGD without initial noise and PGD with random noise. According to the results, AELFs obtains remarkable improvements for typical AT on both tests. Further, it also gains much higher accuracy than other advanced methods.

2) Following RLFAT (Song et al., 2020), the second comparison is conducted on CIFAR-100. Note that the original study used an irregular network, WRN-32. We use a regular WRN-28 to test AELFs. We adapt the claimed PGD(0.03, 7, 0.0075) for evaluation. The results are reported in Table 5. Overall, although we use the smaller WRN-28, AELFs performs much better than the others for generalization. For robustness, AELFs performs better than AT but slightly worse than TRADES, particularly in combinations of RLFAT and TRADES.

5 Conclusion

Given the cascaded structure of a deep model, we recognize the CNN module as a feature extraction item that conducts feature selection and reconstruction for an input. In this view, to check the trade-off issue, the separability of inputs in the original space is not important. Because the global mapping should be from embedding to label. Instead, the spatial attention of the model to input space plays a key role in feature selection. Intuitively, more broad spatial attention can produce more pixel-level feature references to deal with potential variations of unseen inputs.

To study the trade-off issue, we provide an attention-based comparison between ST and AT. This insight reveals that AT usually learns the mapping from training inputs to labels with less spatial attention. In fact, the original minimax construction has decided that the initial motivation of AT is to achieve the mapping learning with as little spatial features as possible. This certainly avoids the perturbation accumulation. However, due to the ceaseless internal maximization during training, AT usually discards some discriminative information of inputs, thus ignores relatively global spatial attention. This seems to be one most straightforward inducement to hurt the generalization.

Overall, AT naturally yields an over-fitting situation according to the view of attention. The experiment verification on the shifted MNIST test sets confirms our argument. In practice, we present the AELF to ensure the model's attention to input space. To boost robustness, it advocates to append mapping from adversarial inputs to clean embedding (from natural inputs) under a standard training setting. Further, we provide a clever AELFs approach to give a convenient implementation. The evaluation results provide strong supports for the rationality of our method.

REFERENCES

Raghunathan Aditi, Sang Xie Michael, Yang Fanny, Duchi John, and Liang Percy. Understanding and mitigating the tradeoff between robustness and accuracy. In *ICML*, pp. 7909–7919, 2020.

Tao Bai, Jinqi Luo, Jun Zhao, and Bihan Wen. Recent advances in adversarial training for adversarial robustness. In *IJCAI*, pp. 4312–4321, 2021.

Jacob Buckman, Aurko Roy, Colin Raffel, and J. Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *ICLR*, 2018.

M. Jeremy Cohen, Elan Rosenfeld, and Zico J. Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, pp. 1310–1320, 2019.

J. Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2014.

Chuan Guo, Mayank Rana, Moustapha Cissé, and van der Laurens Maaten. Countering adversarial images using input transformations. In *ICLR*, 2018.

R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, volume 2, pp. 1735–1742, 2006.

Zheng Haizhong, Zhang Ziqi, Gu Juncheng, Lee Honglak, and Prakash Atul. Efficient adversarial training with transferable adversarial examples. In *CVPR*, pp. 1178–1187, 2019.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2019.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NIPS*, pp. 125–136, 2019.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2016.

Connie Kou, Kuan Hwee Lee, Khim Teck Ng, and Ee-Chien Chang. Enhancing transformation-based defenses against adversarial attacks with a distribution classifier. In *ICLR*, 2020.

Alexey Kurakin, J. Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR*, 2017.

Rice Leslie, Wong Eric, and J. Kolter Zico. Overfitting in adversarially robust deep learning. In *ICML*, pp. 8093–8104, 2020.

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

Mohammad Mehrabi, Adel Javanmard, A. Ryan Rossi, Anup Rao, and Tung Mai. Fundamental tradeoffs in distributionally adversarial training. In *ICML*, pp. 7544–7554, 2021.

Matthew Mirman, Timon Gehr, and T. Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *ICML*, pp. 3575–3583, 2018.

Muzammal Naseer, Salman Khan, Munawar Hayat, Shahbaz Fahad Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *CVPR*, pp. 259–268, 2020.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *ICLR*, 2021.

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)*, pp. 582–597, 2016.

Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. In *NIPS 2019*, pp. 13824–13833, 2019.

R. Ramprasaath Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *IJCV*, pp. 336–359, 2020.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, P. John Dickerson, Christoph Studer, S. Larry Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free. In *NIPS*, pp. 3353–3364, 2019.

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

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

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *ICLR*, 2019.

Lily Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. Towards fast computation of certified robustness for relu networks. In *ICML*, pp. 5276–5285, 2018.

Eric Wong and Zico J. Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*, pp. 5283–5292, 2018.

Eric Wong, Frank R Schmidt, Jan Hendrik Metzen, and J Zico Kolter. Scaling provable adversarial defenses. In *NIPS*, pp. 8410–8419, 2018.

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

Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. In *NIPS*, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, pp. 87.1–87.12, 2016.

Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. In *ICLR*, 2020.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, P. Eric Xing, El Laurent Ghaoui, and I. Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, pp. 7472–7482, 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 *NIPS*, pp. 4939–4948, 2018.

Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. In *ICLR*, 2020.

A RELATED WORK

A.1 ADVERSARIAL TRAINING

Empirical adversarial training, to alleviate the vulnerability of a model to adversarial examples, empirical AT uses natural examples $\mathbf{x}$ to generate their adversarial versions $\mathbf{x}'$ for parameter update. One main line is how to generate adversarial perturbations δ for natural input $\mathbf{x}$.

The pioneering FGSM (Goodfellow et al., 2014) directly used loss-based gradient sign to characterize the adversarial effects of δ. This gradient sign has been the foundation of various optimization-based attacks. 1) To overcome more powerful attacks, stronger BIM and PGD (Madry et al., 2018) iteratively applied FGSM under a bound constrain ϵ. However, multiple iterations increase the computational cost. 2) To improve efficiency, the transferability (Haizhong et al., 2019) of δ between neighbor training epochs is leveraged. For example, (Shafahi et al., 2019) re-modified the training process to combine the iteration of parameter update and perturbation generation. Even (Haizhong et al., 2019) achieved reasonable accumulation of δ across epochs. Actually, as shown in (Wong et al., 2020), it does not need to conduct iterative gradient accumulation. One random noise initialization is enough to enable FGSM to produce strong perturbations.

Certified adversarial training, to give provable guarantees on robustness, certified AT attempts to conduct AT under the upper bound of the perturbations instead of empirical perturbations. Typically, the convex outer adversarial polytope method (Wong & Kolter, 2018) used a dual-network to optimize a convex outer bound of perturbations thus provided a safe region . Further, (Mirman et al., 2018) bridged abstract interpretation and gradient-based optimization to conduct training.

Certified ATs are often limited to shallow networks with the ReLU activation and run slowly due to the slow bound propagation process. Thus recent works (Weng et al., 2018; Zhang et al., 2020) are proposed to improve training efficiency for a large model (Wong et al., 2018), and one study (Zhang et al., 2018) extended ReLU activation to general activation functions. Besides, randomized smoothing (Cohen et al., 2019; Zhai et al., 2020) have proved to induce l_2-norm based certifiable robustness.

Overall, empirical AT has provided one simple but effective way to boost the robustness. However, its empirical supply of perturbations be only viewed as lower bound of inner max, so it is often criticized for the lack of formally certified guarantees. Certified AT provides provable guarantees, but it is usually limited to shallow models and ReLU activation due to difficult parameter optimization.

A.2 STUDIES OF THE TRADE-OFF ISSUE

AT has been the most successful approach to build a robust model. However, both empirical AT and certified AT have shown the trade-off issue. Pioneering work (Tsipras et al., 2019) argued the trade-off might be inevitable due to different goals of robustness and generalization. Theoretically, TRADES (Zhang et al., 2019) decomposed the robust error as the sum of the natural and the boundary error to characterize the trade-off, and provide a differentiable upper bound using the theory of classification-calibrated loss. Further, (Yang et al., 2020) presented that they can be achieved together by rounding a locally Lipschitz function for an r-separated dataset. Besides, (Kou et al., 2020) focused on the trade-off issue on transformation-based techniques.

To address the trade-off issue, the most natural idea is to use natural data to append a regularization such as JT in BIM. Further, instead of the rigid label mapping from adversarial inputs, TRADES checked label-level vector output between natural and adversarial input to boost robustness. This pushes the smoothness of output. Besides, by investigating the relationship between generalization and local features, RLFAT (Song et al., 2020) presented a random block shuffle transformation technique to provide robust local features for AT. (Aditi et al., 2020) proved that robust self-training with extra unlabeled data also can improve this issue.

In this paper, we no longer consider a deep model as one entire mapping function from global input to label. We focus on the model's interesting area to input space. This approach provides a new view to understanding the trade-off issue.

B THE ANALYSIS OF THE PROPOSED AELFS

B.1 WHY CAN AELFS CONDUCT THE ATTENTION EXPANSION CONCEPTION OF AELF

As section 3.1 claimed, for a model $\mathcal{M}_\theta$, AELF recognizes global mapping learning $\mathbf{x} \overset{\theta}{\mapsto} y$ as split $\mathbf{x} \overset{c}{\to} \mathbf{f_x} \overset{w}{\to} y$. To boost robustness, AELF suggests to conduct $\mathbf{x}, \mathbf{x}' \overset{c}{\to} \mathbf{f_x} \overset{w}{\to} y$. The key is to append a mapping training $\mathbf{x}' \overset{c}{\to} \mathbf{f_x}$ to avoid the misleading from adversarial perturbations. To give a convenient solution, AELFs directly fixed the pre-trained parameter w to implement the idea of AELF under single softmax-based loss. This is mainly derived from following discussion.

The label mapping layer w of one deep model is usually a linear structure without activation function. Given the clean feature embedding $\mathbf{f_x}$ from natural $\mathbf{x}$. Its label usually identifies it via the matrix mapping $w = (\mathbf{w}, \mathbf{b})$ and the softmax operation. This can be formalized as

$$y = \frac{e^{\bar{\mathbf{w}}_y^{\mathrm{T}} \mathbf{f_x}}}{\sum_{k=1}^{K} e^{\bar{\mathbf{w}}_k^{\mathrm{T}} \mathbf{f_x}}} . \tag{11}$$

where, we use $\bar{\mathbf{w}}$ to replace $(\mathbf{w}, \mathbf{b})$, and $\bar{\mathbf{w}}_k^{\mathrm{T}}$ is the kth column of matrix $\bar{\mathbf{w}}$. For one-hot coding of y, $\bar{\mathbf{w}}$ maps feature $\mathbf{f_x}$ into the softmax space, in which the label vector $l_k \in \mathbb{R}^K$ is represented by $l^{k(k=y)} = 1$ and $l^{k(k\neq y)} = 0$. To ignore the constant operation of softmax, this reveals $\mathbf{f_x}$ is linearity relevance to $\bar{\mathbf{w}}_y^{\mathrm{T}}$ and irrelevance (even negatively relevance) to $\bar{\mathbf{w}}_{k(k\neq y)}^{\mathrm{T}}$ when the loss$\to 0$. In this case, a inference can be concluded as that w dominates the distribution of embedding $\mathbf{f_x}$.

Further, for the optimization of $\mathcal{L}\{\mathcal{M}_w[\mathcal{M}_c(\mathbf{x})];\ y\} = \mathcal{L}[\mathrm{softmax}(\bar{\mathbf{w}}^{\mathrm{T}} \cdot \mathbf{f});\ y]$, we can obtain the gradient as $\frac{\partial \mathcal{L}}{\partial c} \Rightarrow \frac{\partial \mathcal{L}}{\partial \mathbf{f}} \frac{\partial \mathbf{f}}{\partial c} \Rightarrow \mathcal{L}' \cdot \bar{\mathbf{w}}^{\mathrm{T}} \cdot \frac{\partial \mathbf{f}}{\partial c}$ under the chain rule of back propagation, where $\mathbf{f} = \mathcal{M}_c(\mathbf{x})$. As an intermediate variable, w ($\bar{\mathbf{w}}$) is multiplicatively related to the gradient $\frac{\partial \mathcal{L}}{\partial c}$. In other word, w plays as a guidance to update the fronted feature extraction module c.

Thus a conclusion is reached, under the situation that w dominates the distribution of embeddings, $\mathcal{L}_\mathbf{x}$ of Eq. (9) produces a clean $w_\mathbf{f}$ by inherent $\mathbf{f_x}$ of natural $\mathbf{x}$. This clean label mapping parameter $w_\mathbf{f}$ enables $\mathcal{L}_{\mathbf{x}'}$ indirectly achieve $\mathbf{x}' \mapsto \mathbf{f}$ during the update of feature extraction module c.

B.2 SENSITIVITY ANALYSIS ON HYPER-PARAMETER β

In this subsection, to give one rough range of β in our implementation using Eq. (10), we fix $\lambda_1, \lambda_2 = 0.5$ to conduct an experiment with CNNB on CIFAR-10. Fig. 5 shows its result tendency.

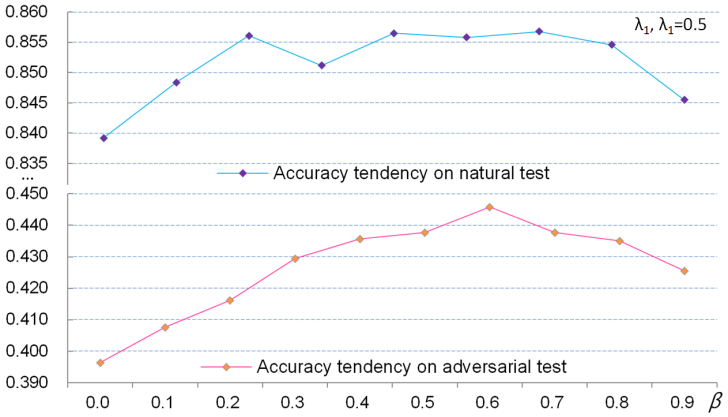


Figure 5: Sensitiveness test of β with CNNB on CIFAR-10.

Overall, as β being increased, both accuracies are in upward tendency before 0.6. When β >0.8, it yields a negative effect. In this study, β is suggested to $[0.1, 0.5]$ for CNNB. For other models with higher embedding dimensions, we suggest setting smaller β to avoid non-convergence. Actually, because the model parameter is from standard training, the big constrain $\beta(*)$ can limit them to flee the minimum region of initialization. In practice, we find that higher embedding dimension usually needs to smaller β value, to avoid that feature norm $\|*\|_2$ controls entire loss.

C TRAINING SETTING

We conduct all trainings with pytorch under random seed=8. All adversarial data are produced by PGD based on l_∞-norm without initial noise. To avoid the overlay of training tricks, we select a normal setting from (Pang et al., 2021). Note that our training settings don't adapt early stopping tricks (Leslie et al., 2020) like TRADES (Zhang et al., 2019).

Uniform settings for all experiments: 1) Inputs are normalized to $[0, 1]$, batch size is uniformly set to 128. For different JT ways, the second half inputs (64) are used to generate adversarial data and combined with rest of natural data. 2) For all models, we adapt eval mode during the generation of adversarial data. 3) All trainings adapt SGD optimizer with initial learning rate lr=0.1 and l_2 regularization $5e$-4. Exceptionally, because CNN$_B$ is very small to deal with CIFAR-10, we apply weight decay $2e$-4 to its training steming from (Madry et al., 2018). Besides, all adversarial data are produced by PGD based on l_∞-norm.

Setting on MNIST: 1) Inputs are resized as $32 * 32$ and not applied with any data argumentation before feeding to the network. 2) The initial lr=0.1 is changed to 0.01 and 0.001 after $20th$ and $40th$ epoch. We conduct 50 epochs for all methods. To avoid special case from single epoch setting, more results from total 30 epoch are listed in Appendix D

Setting on CIFAR-10 and CIFAR-100: 1) Inputs adapt random flips and cropping (padding=2). 2) ST is conducted with 100 epochs, lr is changed into 0.01 and 0.001 after $40th$ and $80th$ epoch. All other methods are trained with 160 epochs, lr is equivalently changed after $50th$ and $100th$ epochs. To avoid customized settings for different models, this setting is applied to all models in this paper.

D MORE INVESTIGATION AND COMPARISONS ON MNIST

In this section, we provide additional results to further support the viewpoint in the main text.

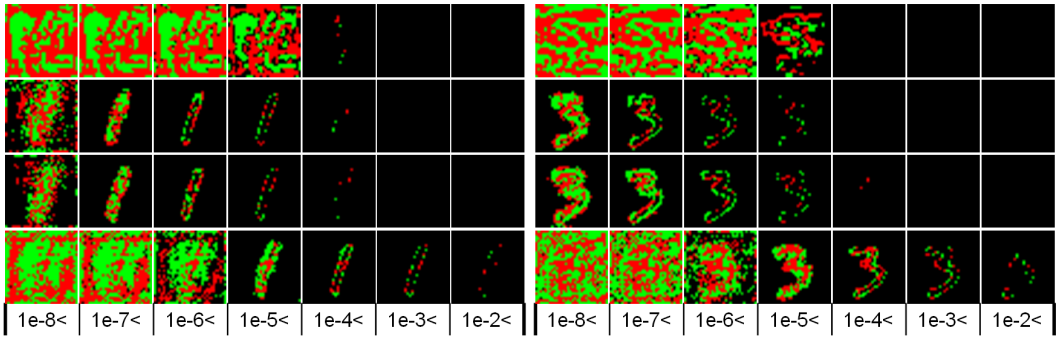
1) The first comparison is to reduce the total training epoch. The main motivation is avoid that we provide a special case in main text, such as the 50 epochs facilitate an over-training or make an over-fitting situation. In this subsection, we adapt uniform settings following section 4.2. Exceptionally, we only train the CNN$_B$ with 30 epochs, and the initial lr=0.1 is changed to 0.01 and 0.001 after $10th$ and $20th$ epoch. The test results on natural and shift test sets have been listed in Table 6.

Table 6: Comparison results on MNIST with CNN$_B$, which is trained with total 30 epochs.

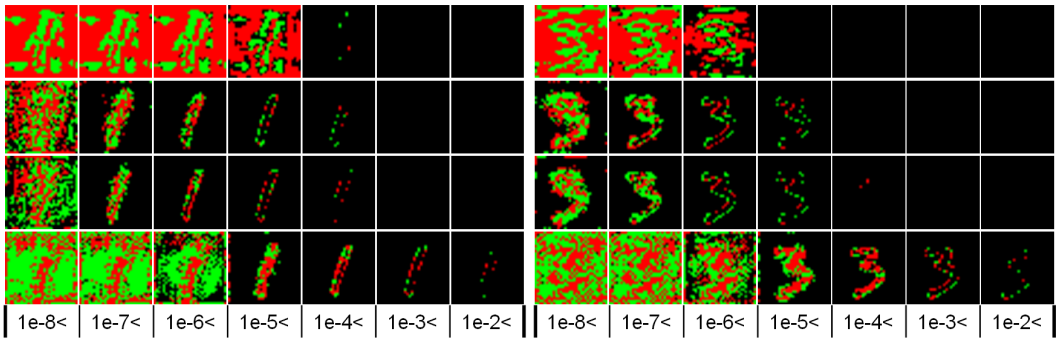
	Natural test					Adversaria test				
Method	N	N(L)	N(R)	N(U)	N(D)	N	N(L)	N(R)	N(U)	N(D)
ST	99.60	98.57	96.07	94.55	93.04	0.0	0.0	0.0	0.0	0.0
AT	99.31	92.33	90.87	60.32	64.22	93.35	67.66	67.75	24.43	30.40
JT	99.29	92.79	93.99	68.28	67.68	92.97	68.80	72.93	27.05	32.67
AELFs	99.27	95.67	93.52	77.18	80.89	93.39	77.19	74.40	48.68	45.12

2) The second comparison is to further observe the difference of gradient signs between different trained models. Here, following Fig. 4, we provide more exhibitions about another characters '1' and '3' based on our two result models CNN$_B$(50) and CNN$_B$(30). Fig. 6 shows their comparison results between CNN$_B$(50) and CNN$_B$(30).

Overall, according to above two comparisons, we can find that: In terms of test performance, there was no significant changes between CNN$_B$(50) and CNN$_B$(30). Further, Fig. 6 shows similar situation about gradient signs. This imply that AT maybe inherently yields a over-fitting tendency in perspective of spatial attention.



(a) Gradient signs exhibition of natural character '1' and '3' from $\text{CNN}_{\text{B}}(50)$.



(b) Gradient signs exhibition of natural character '1' and '3' from $\text{CNN}_{\text{B}}(30)$.

Figure 6: Comparison of gradient signs on natural character '1' and '3' between $\text{CNN}_{\text{B}}(50)$ and $\text{CNN}_{\text{B}}(30)$.