



Article

Improving Adversarial Robustness of CNNs via Maximum Margin

Jiaping Wu , Zhaoqiang Xia  and Xiaoyi Feng *

School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China

* Correspondence: fengxiao@nwpu.edu.cn

Abstract: In recent years, adversarial examples have aroused widespread research interest and raised concerns about the safety of CNNs. We study adversarial machine learning inspired by a support vector machine (SVM), where the decision boundary with maximum margin is only determined by examples close to it. From the perspective of margin, the adversarial examples are the clean examples perturbed in the margin direction and adversarial training (AT) is equivalent to a data augmentation method that moves the input toward the decision boundary, the purpose also being to increase the margin. So we propose adversarial training with supported vector machine (AT-SVM) to improve the standard AT by inserting an SVM auxiliary classifier to learn a larger margin. In addition, we select examples close to the decision boundary through the SVM auxiliary classifier and train only on these more important examples. We prove that the SVM auxiliary classifier can constrain the high-layer feature map of the original network to make its margin larger, thereby improving the inter-class separability and intra-class compactness of the network. Experiments indicate that our proposed method can effectively improve the robustness against adversarial examples.

Keywords: adversarial examples; margin; auxiliary layer



Citation: Wu, J.; Xia, Z.; Feng, X. Improving the Adversarial Robustness of CNNs via Maximum Margin. *Appl. Sci.* **2022**, *12*, 7927. <https://doi.org/10.3390/app12157927>

Academic Editors: Howon Kim and Thi-Thu-Huong Le

Received: 16 June 2022
Accepted: 3 August 2022
Published: 8 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Convolutional neural networks (CNNs) have been widely used in various domains, especially in computer vision. CNNs have satisfying learning capabilities as they can extract image features of different scales by multiple convolutional layers. However, various CNN architectures are vulnerable. The models will predict wrong results with small perturbations adding to the inputs that humans cannot recognize. In other words, adversarial examples [1–3], which are similar to clean examples, will mislead the CNN to output incorrect results. Moreover, adversarial examples can transfer across different CNN models [4,5]. These phenomena indicate that the CNN learns some simple patterns to accomplish tasks exactly on specified data but does not recognize more robust advanced features like the human brain. So the CNN is not safe for medical care [6], autonomous driving [7], and other fields.

The potential serious harm of adversarial examples has drawn the attention of the scientific research community, forming a new field called adversarial machine learning. There are two main types of defense methods against adversarial examples, i.e., certified methods and empirical methods. Certified methods [8,9] can obtain provable robustness, but the provable robustness is low and most of them only aim at the attack with disturbance of l_2 norm. Empirical methods account for the majority of defense methods, including stochastic activation pruning (SAP) [10], distillation defense [11], asymmetrical adversarial training [12], ME-net [13], etc. However, the robustness of empirical methods is not reliable, and it may be broken by stronger attacks [14,15].

Adversarial training (AT) [16] is one of the most successful empirical defense methods at present which is a data augmentation technique for training models on both natural and adversarial examples. Although AT is an empirical method, it has been proved to be highly robust even under adaptive attack [14]. AT effectively robustifies CNN but

has two challenges: lower clean accuracy [17] and the overfitting problem [18,19]. To address these challenges, many variants of AT have been proposed, such as TRADES [20], customized adversarial training (CAT) [21], dynamic adversarial training (DAT) [22], and geometry-aware instance-reweighted adversarial training (GAIRAT) [23]. However, these two problems are still unsolved very well. So this paper is devoted to them by considering the maximum margin.

The key insight is that the purpose of adversarial machine learning, i.e., the maximizing margin is very similar to that of SVM. Theoretically, the margin of the classifier in the input space is positively correlated with its robustness. However, for multi-layer neural networks, the margin on the input space is difficult to be calculated and constrained. AT can indirectly increase the margin of the classifier in the input space because the adversarial examples are obtained by moving clean examples toward the decision boundary. We find that ordinary training methods such as using cross-entropy loss cannot effectively increase the margin. So we propose adversarial training with supported vector machine (AT-SVM) that increases the margin by SVM. Specifically, we insert an SVM auxiliary classifier before the last layer in the network, which can maximize the margin on this feature space. The SVM is a better classifier for high-level feature space, so we use it to identify more confusing points which will be trained on the original network. With continuous learning, the boundary of the original network will gradually approach the boundary of the SVM. At the same time, in order to ensure that the margin on the feature space is consistent with the margin on the input space, we constrain the distance between the clean example and the corresponding adversarial example in the margin direction of this feature space. Experiments indicate that our proposed method can effectively improve the robustness against adversarial examples.

2. Related Work

In this section, we briefly review the adversarial training and defense methods based on it.

2.1. Adversarial Attacks

With the proposal of adversarial examples, the earliest method used to generate adversarial examples is the fast gradient sign method (FGSM) [1]. FGSM generates adversarial examples with a single normalized gradient step. Since then, a number of gradient-based and multi-step iterative attacks have been proposed such as projected gradient descent (PGD) [16], C&W attack [24], and DeepFool [25]. We review the PGD attack which is the most popular and widely used attack.

PGD aims to find an adversarial example \hat{x} for an input x that satisfies a given boundary $\|\hat{x} - x\|_p < \epsilon$. Let \mathbf{B} denotes the l_p -ball of radius ϵ centered at x . The attack initializes the adversarial example with a random point $x_0 \in \mathbf{B}$, and iteratively updates it with the gradient, as shown in Equation (1).

$$\begin{aligned} x_{i+1} &= \text{Proj}_{\mathbf{B}}(x_i + \alpha \cdot g) \\ \text{s.t. } g &= \arg \max_{\|v\|_p \leq 1} v^\top \nabla_{x_i} L(f(x_i), y) \end{aligned} \quad (1)$$

where $L(\cdot, \cdot)$ is a suitable loss-function (e.g., cross-entropy), f is the objective model, α is a step size, $\text{Proj}_{\mathbf{B}}$ projects an input onto the l_p -ball \mathbf{B} , and g is the direction of gradient that loss ascents fastest for a given l_p -norm. For the l_∞ -norm, $\text{Proj}_{\mathbf{B}}$ is a clipping operator and $g = \text{sign}(\nabla_{x_i} L(x_i, y))$.

2.2. AT-Based Defense Methods

For a CNN model with parameters θ , the goal of adversarial defense is to solve the following min-max optimization problem, as shown in Equation (2)

$$\min_{\theta} \mathbb{E}_{(x,y) \sim D} \max_{\hat{x} \in \mathbb{S}_0(x)} \mathcal{L}(h_{\theta}(\hat{x}), y) \quad (2)$$

where the inner maximization is intractable, so adversarial training replaces it with a lower bound which is obtained by an adversarial attack. The standard adversarial training (SAT) [16] maximized the inner loss using PGD attack and found it performs well against other adversarial attacks.

Several improvements for standard adversarial training have been proposed. Curriculum adversarial training [26] trains a model from weak attack to strong attack. Dynamic adversarial training (DAT) [22] proposes a dynamic training strategy to increase the convergence quality of the generated adversarial examples gradually. TRADES [20] traded adversarial robustness for accuracy. Misclassification aware adversarial training (MART) [27] explicitly differentiates the misclassified and correctly classified examples during the training. Friendly adversarial training (FAT) [28] searches for the least adversarial data (i.e., friendly adversarial data) by minimizing the loss that makes results confidently misclassified rather than employing the most adversarial data to maximize the loss. Customized adversarial training [21] proposes auto-tuning perturbation and adaptive label smoothing for adversarial training. Goyal et al. [29] improved the state-of-the-art robustness by largening the model's Swish/SiLU activations and model weight averaging. Geometry-aware instance-reweighted adversarial training (GAIRAT) [23] regards iterations needed for an attack to misclassify an example as its importance and then uses it to reweight all training examples.

2.3. AT-Used Defense Methods

In addition to the improvements of AT, there are many defense methods requiring the assistance of AT to implement specific functions or achieve higher robustness.

Mustafa et al. [30] learned centers in hidden feature space to separate higher-order representations of different classes. Taghanaki et al. [31] proposed a defense strategy based on kernel function, which increases robustness through learnable Mahalanobis distance. Channel-wise activation suppressing (CAS) [32] suppressed redundant activation from being activated by adversarial perturbations. Ref. [33] used spatial attention to realize adversarial robustness. Channel-wise importance-based feature selection (CIFS) [34] built a probe network to get the importance mask for each channel. Feature denoising [35] uses non-local mean filtering, mean filtering, median filtering, and bilateral filtering to perform feature denoising for improving the robustness of countermeasures.

These methods have changed the network structure and used it as the main contribution, and need to be combined with adversarial training. Although they provide a theoretical basis for changing the network structure, it is difficult to get effective improvement if the adversarial training is not added at the same time in the experiment (even if there is, it is probably due to gradient masking).

3. Proposed Method

In this section, we firstly introduce the motivation and basic theories of our proposed method. Then we consult its learning objective as well as its algorithmic implementation. Lastly, we compare our method with other existing defense methods.

3.1. Motivation

On the one hand, according to a common empirical result from researchers of adversarial defense [16,23], a larger model capacity is needed for adversarial examples. Therefore, in the case of insufficient model capacity, it is unwise to treat all examples equally. Training more important examples is helpful to make better use of the model capacity. On the other

hand, intuitively, the examples close to the classification boundary are more vulnerable to adversarial attacks and are more important for the classification boundary. This widely accepted assumption can be proved empirically. Figure 1 shows the distribution of the feature map before the last layer of the two randomly selected classes in the CIFAR-10 dataset. It can be seen that the features of the misclassified examples are concentrated near the boundary no matter whether it is adversarial examples using targeted or untargeted attack.

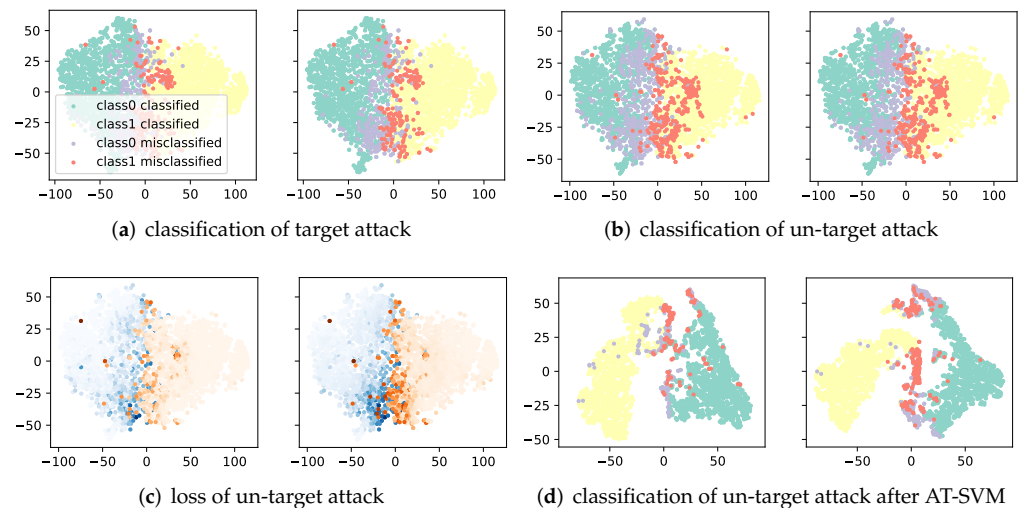


Figure 1. Two-dimensional visualizations of the model's output distribution of natural examples and adversarial examples from two separated classes from the CIFAR-10 dataset. For each subfigure, the left channel indicates the result from the original model and the right channel indicates the result from the SVM auxiliary classifier. First, we train the original model using SAT. Then we insert our SVM auxiliary classifier and train it but with the original model fixed. The classifying results are shown in (a,b). The loss of them is shown in (c). Last, we use our AT-SVM to train a model, the result is shown in (d).

According to the idea of SVM [36], to obtain a linear binary classifier that maximizes the margin between two classes, only the examples closest to the decision boundary, i.e., the support vectors, are needed. For a CNN network, although it is not a linear classifier, the last layer is always a fully connected (FC) layer that is linear. So the examples can be transformed into linearly separable high-order feature space after passing the front layers. According to these high-order features, we can find out which examples are support vectors that are close to the boundary. Training models on these examples can improve the efficiency of adversarial training.

For linear classification tasks, SVM can learn the classifier with the largest margin. We extend the SVM and add an SVM auxiliary layer into the CNNs when training so that the features extracted by the CNNs have a larger margin, hence improving the adversarial robustness of the CNN.

3.2. Theories

3.2.1. Linear Binary-Classification Task

For a binary linearly separable classification problem, there exist many hyperplanes that might classify the data. SVM is designed to compute the classification hyperplane which represents the largest separation, or margin, between two classes. The max-margin hyperplane is also known as the optimal classification hyperplane.

The main idea of finding the optimal classification hyperplane is to maximize the distance between different classes. Because of the possible inseparable points in data, slack variable ζ and penalty coefficient C are generally added. So the essence is to solve the following optimization problems, as shown in Equation (3)

$$\begin{aligned} & \min \frac{1}{2} \|\omega\|_2^2 + C \sum_{i=1}^m \zeta_i \\ & \text{s.t. } y^{(i)} \left(\omega^T x^{(i)} + b \right) \geq 1 - \zeta_i \\ & \quad \zeta_i \geq 0, \quad i = 1, 2, \dots, m. \end{aligned} \tag{3}$$

where $y^{(i)} = \pm 1$ indicate that $x^{(i)}$ is a positive or negative example.

These constraints of the optimization problem in Equation (3) can be solved by mathematical methods such as the sequential minimal optimization (SMO) [37] algorithm. However, in order to replace the original linear FC layer in the training of CNN, this formula cannot be directly used as a new loss function, as there are inequality constraints. So the following content will deform this optimization problem.

The constraints are equivalent to Equation (4).

$$\zeta_i \geq \max \left(0, 1 - y^{(i)} \left(\omega^T x^{(i)} + b \right) \right) \tag{4}$$

The inequality constraint in Equation (4) shows the lower bound of ζ_i . When solving the minimization problem outside, we approximately replace ζ_i with the lower bound of it. Let $C' = 1/2C$. Equation (3) can be written as Equation (5).

$$\min \sum_{i=1}^m \max \left(0, 1 - y^{(i)} \left(\omega^T x^{(i)} + b \right) \right) + C' \|\omega\|_2^2 \tag{5}$$

Since there is no inequality constraint, the new formula can be directly used as the loss function. Notice that $\sum_{i=1}^m \max \left(0, 1 - y^{(i)} \left(\omega^T x^{(i)} + b \right) \right)$ is the hinge loss function. The other item $C' \|\omega\|_2^2$ can be regarded as a regularization term added to improve the generalization performance of the model.

3.2.2. Linear Multi-Classification Task

For a multi-class classifier, the predicted label is chosen by the maximal logit attained over all classes:

$$\hat{y} = \arg \max_k f_k(x)$$

where $f_k(x)$ is the output of $f(x)$ corresponds to the k th class. In linear case, $f_k(x) = \omega_k^T x + b_k$.

For any two classes (p, q) , the decision boundary between them is called $l_{p,q}$.

$$\begin{aligned} l_{p,q} &= \left\{ x \mid \omega_p^T x + b_p = \omega_q^T x + b_q \right\} \\ &= \left\{ x \mid \left(\omega_p^T - \omega_q^T \right) x + b_p - b_q = 0 \right\} \end{aligned} \tag{6}$$

The geometric distance of a point x from $l_{p,q}$ is:

$$d_{p,q}(x) = \frac{\left(\omega_p^T - \omega_q^T \right) x + b_p - b_q}{\left\| \left(\omega_p^T - \omega_q^T \right) \right\|_2} \tag{7}$$

Some margin-based methods only consider the margin between the ground truth class and the most easily misclassified class, i.e., setting $p, q = \arg \max_{i \neq y} f_i(x)$. Instead, our method considers all margins between each class and the other classes. In other words, we regard a multi-classification task as a multiply binary classification task. Equation (5) can be generalized to the following optimization problem in Equation (8).

$$\min \sum_{i=1}^m \sum_{j=1}^n \max \left(0, 1 - y_j^{(i)} \left(\omega_j^T x^{(i)} + b_j^T \right) \right) + C' \|\omega_j\|_2^2 \tag{8}$$

where n is the class number, $y_j^{(i)} = \pm 1$ indicate $x^{(i)}$ is/not belong to class j .

3.2.3. General Classification Task

For a nonlinear classifier, the approximation scheme from Elsayed et al. [38] can be adopted to capture the distance of a point to the decision boundary which is a first-order Taylor approximation to the true distance. Equation (8) can be extended to:

$$d_{p,q}(x) = \frac{f_p(x) - f_q(x)}{\|\nabla_x f_p(x) - \nabla_x f_q(x)\|_2} \tag{9}$$

Using this approximation, we can calculate margins on all layers, just replace the input x in the above formula with intermediate features x^l on different layers. The training data x induces a distribution of distances at each layer l which, following earlier naming convention [39,40], we refer to as margin distribution (at layer l).

Intuitively, constraining the margin distribution of the total network to be large enough can lead to a robust model. However, this approximation will lose accuracy fast as inputs move away from the decision boundary. Moreover, it is difficult to learn a model with large margin distribution. So that our proposed method only constrains the margin distribution on the last FC layer, just solve the optimization problem in Equation (8). At the same time, use l_{margin} introduced later to limit the deviation between the margin distribution on the last FC layer and the margin distribution of the total network.

3.3. AT-SVM

In this section, we introduce our supported vector machine adversarial training (AT-SVM) method, which dynamically learns an SVM auxiliary classifier and restricts the decision boundary of the original network to achieve a large margin.

3.3.1. Overview

As Figure 2 shows, our method is mainly inserting an auxiliary SVM classifier in the penultimate layer of the network, which is parallel to the last layer. The SVM layer contains as many neurons as the number of classes and is purposed to obtain the maximum margin decision boundary in this feature space. The SVM auxiliary classifier treats the front layer of the original network as a feature extractor and uses a separate optimizer for training. In each epoch of training, the SVM auxiliary classifier and the original network are trained successively, and a mask on which examples will be trained is produced by the SVM auxiliary classifier. The SVM auxiliary classifier is discarded during inference. The specific implementation will be introduced in the algorithm.

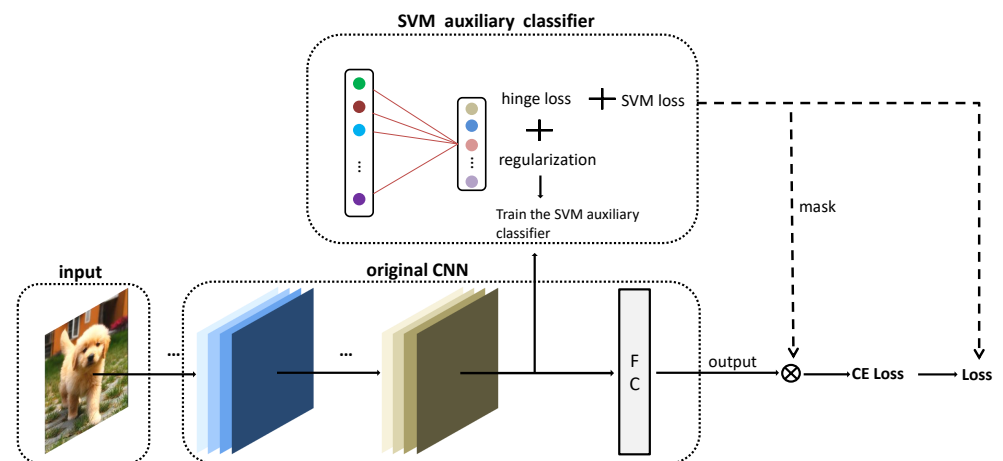


Figure 2. Framework of our proposed AT-SVM.

In CNNs, the last FC layer is linear, so we regard the front layers as a feature extractor. Using ϕ to denote the extracted feature, we can train an SVM auxiliary classifier using Equation (10).

$$\min \sum_{i=1}^m \sum_{j=1}^n \max \left(0, 1 - y_j^{(i)} \left(\omega_j^T \phi^{(i)} + b_j^T \right) \right) + C' \|\omega_j\|_2^2 \tag{10}$$

where the first item is the hinge loss represented by L_{hinge} below. However, there is a problem in training the network with Equation (10) that the feature space ϕ is constantly changing. Accordingly, maximizing the margins in Equation (10) can be trivially attained by scaling up the space ϕ . This problem can be solved by using the SVM auxiliary layers, because $C' \|\omega_j\|_2^2$ only helps training the SVM auxiliary and indirectly affects L_{hinge} . The training of the main network is only using L_{hinge} , which constrains the feature extractor to separate input examples of different classes in the feature space. The auxiliary classifier is affected by both L_{hinge} and $C' \|\omega_j\|_2^2$, and continuously obtains a better classification boundary. The change of the boundary will also affect the value of L_{hinge} . The boundary and L_{hinge} will be converged under the continuous alternating training of the original network and the auxiliary classifier.

The distance from $\phi^{(i)}$ to the binary decision boundary of class j is:

$$d_j(\phi^{(i)}) = \frac{\omega_j^T \phi^{(i)} + b_j}{\|\omega_j^T\|} \tag{11}$$

For a clean example $x^{(i)}$, we denote its adversarial example as $\hat{x}^{(i)}$. Only a large margin in SVM auxiliary cannot guarantee robustness because the extracted feature $\hat{\phi}^{(i)}$ of $\hat{x}^{(i)}$ may have large distance with $\phi^{(i)}$. So we will constrain the movement of the adversarial examples relative to the clean examples in the margin direction, as shown in Equation (12).

$$d_j(\phi^{(i)}) - d_j(\hat{\phi}^{(i)}) = \frac{\omega_j^T (\phi^{(i)} - \hat{\phi}^{(i)})}{\|\omega_j^T\|_2} \tag{12}$$

After normalization, the following objective function can be obtained, as shown in Equation (13).

$$L_{margin} = \sum_i \sum_j \max \left(\frac{d_j(\phi^{(i)}) - d_j(\hat{\phi}^{(i)})}{d_j(\phi^{(i)})}, 0 \right) \tag{13}$$

In each epoch, we will first train the SVM auxiliary classifier using Equation (10). Then compute L_{margin} , L_{hinge} and add them to the loss of the original network to train.

3.3.2. Algorithm Implementation

In order to pick out the important examples close to the decision boundary, AT-SVM adds an SVM auxiliary classifier before the last layer. For each mini-batch, adversarial examples will be generated using the gradient of the original network, and a mask of all examples is obtained according to the results of the SVM auxiliary classifier to pick out the more important examples. Finally, the original network trains only on these selected examples. In the final inference, the SVM auxiliary is removed.

In each mini-batch, first, we train the SVM auxiliary classifier with all parameters of the original model fixed. When training the SVM layer, the loss is as Equation (10) including both hinge loss and the weights' l_2 norm as a regularization term. Then we set the SVM layer fixed and recalculate the hinge loss L_{hinge} and compute loss L_{margin} from SVM auxiliary classifier using Equation (13), send them to backpropagation. At the same time, use whether this loss is 0 to generate a binary mask, and multiply the output from the

original model with the mask before calculating the cross-entropy loss. The procedure is summarized in Algorithm 1.

Algorithm 1 AT-SVM

Require: network f_θ , SVM auxiliary classifier, training dataset $S = \left\{ \left(\mathbf{x}^{(i)}, \mathbf{y}^{(i)} \right) \right\}_{i=1}^n$, loss function $L(\cdot, \cdot)$, learning rate η , number of epochs T , batch size m , number of batches M , weight of loss from SVM auxiliary classifier λ

Ensure: robust network f_θ

- 1: Initialize f_θ, h_θ
- 2: **for** $epoch = 1, 2, \dots, T$ **do**
- 3: **for** $minibatch = 1, 2, \dots, M$ **do**
- 4: Sample a mini-batch $\left\{ \left(\mathbf{x}^{(i)}, \mathbf{y}^{(i)} \right) \right\}_{i=1}^m$ from training dataset S
- 5: **for** $i = 1, 2, \dots, m$ **do**
- 6: Use Equation (1) to generate PGD attack examples $\hat{\mathbf{x}}^{(i)}$
- 7: Get the feature $\phi^{(i)}$ before the last FC layer
- 8: **end for**
- 9: Use Algorithm 2 to train an SVM auxiliary classifier
- 10: Use the SVM auxiliary classifier to obtain a mask for all \mathbf{x} as well as compute loss L_{margin} using Equation (13)
- 11: $L' = \sum_{i=1}^m L\left(f\left(\hat{\mathbf{x}}^{(i)}\right), \mathbf{y}^{(i)}\right) \cdot mask$
- 12: $\theta \leftarrow \theta - \eta \nabla_{\theta} \frac{1}{m} (L' + \lambda L_{margin})$
- 13: **end for**
- 14: **end for**

Algorithm 2 is an SVM auxiliary classifier, which is trained using both the adversarial data and the natural data and returns a mask for them at the same time. AT-SVM leverages Algorithm 2 for obtaining the mask for all examples. For each mini-batch, AT-SVM selects examples that will be misclassified by the SVM auxiliary classifier and then updates the model parameters by minimizing the sum of the selected examples' loss.

Algorithm 2 SVM auxiliary classifier

Require: input feature and labels $\left\{ \left(\phi^{(i)}, \mathbf{y}^{(i)} \right) \right\}_{i=1}^n$, number of classes k , weights ω and bias \mathbf{b} for SVM auxiliary classifier, weight of regularization term C in Equation (10)

Ensure: a *mask* for all input data, new weights ω and bias \mathbf{b} for SVM auxiliary classifier

- 1: Initialize $L(i, j) = 0$
- 2: **for** $i = 1, 2, \dots, n$ **do**
- 3: **for** $j = 1, 2, \dots, k$ and $j \neq \mathbf{y}^{(i)}$ **do**
- 4: $L(i, j) = \max\left(0, 1 - \left(\omega_{\mathbf{y}^{(i)}, j}^T \phi^{(i)} + b_{\mathbf{y}^{(i)}, j}^T\right)\right)$
- 5: **if** $L(i, j) > 0$ **then** $mask[i] = 1$
- 6: **end if**
- 7: **end for**
- 8: **end for**
- 9: $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} \sum_i \sum_j \left(L(i, j) + C \cdot \|\omega_{\mathbf{y}^{(i)}, j}\|_2^2 \right)$
- 10: $\mathbf{b} \leftarrow \mathbf{b} - \eta \nabla_{\mathbf{b}} \sum_i \sum_j \left(L(i, j) + C \cdot \|\omega_{\mathbf{y}^{(i)}, j}\|_2^2 \right)$

3.3.3. Feasibility Verification

We train a standard model using AT first. This model is fixed as a feature extractor and adds the SVM auxiliary classifier before the last FC layer. Then we train the SVM layer. As Figure 1a,b show, the SVM auxiliary classifier can learn the correct decision boundary, which ensures that it will pass the correct mask and loss to the original network. It can be seen that, in general, the points which are misclassified are distributed near the boundary. Adversarial training can be regarded as a data augmentation making trained examples close to decision boundary and then getting larger loss.

In Figure 1c, we compare the original cross-entropy loss with the loss from the SVM auxiliary classifier. Points with darker colors have a greater loss. It can be found that the latter is more significant which is the reason why using the SVM auxiliary classifier: loss in the original CNN network will decrease to nearly zero after training so that training becomes difficult, but the loss of the SVM auxiliary layer is still significant, so training on this loss will make model efficiently learn a large margin on this layer.

With the help of SVM, the high-order features of the example can be separated more effectively. As shown in Figure 1a, using the SAT-trained classifier, the high-order features of the examples and their adversarial examples are not well separated, and a large number of them are distributed near the boundary to cause confusion. Using the classifier trained by the proposed method, the high-order features of the examples are more concentrated and the distance between classes is larger shown in Figure 1d. The misclassified examples are also separated from the correctly classified examples, and several small clusters are formed nearby. This shows that the SVM auxiliary classifier achieves the effect of increasing the inter-class distance and reducing the intra-class distance on high-order features. Empirically, this helps adversarial robustness. For the above reasons, the SVM auxiliary classifier can provide a more significant loss compared with the standard setting and therefore improve the adversarial robustness.

3.4. Comparisons to Other Adversarial-Based Defense Methods

First, the main idea of many variants of adversarial training is to weaken the adversarial examples when training [21,28] or use attacks from the weak to the strong drawing on the ideas of curriculum training [22,26]. Unlike these methods, AT-SVM does not change the attack. Moreover, stronger adversarial examples are more likely to be misclassified by the SVM auxiliary classifier and have the opportunity to be trained by the original network rather than being discarded.

Second, some methods treat adversarial data differently by explicitly assigning different weights to their losses, such as MART [27] and GAIRAT [23]. This kind of method is similar to ours because our method is equivalent to assigning a weight of 0 or 1 to all examples. However, our method is more concerned with finding more important examples near the decision boundary than assigning reasonable weights to all examples. We believe that as long as the training is carried out under the appropriate data, the original surrogate loss will be sufficient. Furthermore, MART regards natural examples which are misclassified as outliers and suppresses the influence of adversarial examples corresponding to these examples. On the contrary, AT-SVM will train more on these misclassified natural examples to avoid the decline in the accuracy of natural examples caused by adversarial training.

Last, max-margin adversarial (MMA) training [41] (Ding et al., 2019) is also declared to improve adversarial robustness by the maximum “margin” like SVM. We emphasize that our AT-SVM is different from MMA in the following aspects: (1) the “margin” in MMA means the “shortest successful perturbation” which is obtained from a PGD attack while the “margin” in AT-SVM is just the loss in SVM auxiliary classifier. So AT-SVM can increase the margin more simply, without using other objective function indirectly like MMA; (2) MMA will first test whether the classification of clean examples are correct or not. For correctly classified examples, MMA adopts cross-entropy loss on adversarial examples; for misclassified examples, MMA directly applies cross-entropy loss on natural examples. Our

AT-SVM treats all examples the same because we believe that the importance of examples can be adequately reflected by the SVM auxiliary classifier.

4. Experiments

In this section, a set of experiments is firstly introduced. Then we use experiments to verify the proposed method and evaluate the impact of parameter settings on the experimental results. Finally, we verify that the proposed method is also effective under different models or data sets.

4.1. Experimental Setup

We use ResNet-18 [42] as the basic model on CIFAR-10 dataset [43]. The basic models are trained using SGD with momentum 0.9, weight decay 5×10^{-4} , and an initial learning rate of 0.1, which is divided by 10 at the 75th and 90th epochs. All clean images are normalized into $[0, 1]$, and simple data augmentations are included in training such as 4-pixel padding with 32×32 random crop and random horizontal flip. As for adversarial training, the training attack is PGD-10 (10 step PGD) with random start and step size $2/255$ and l_∞ maximum perturbation $\epsilon = 8/255$.

4.2. Robustness of AT-SVM Model

In Table 1, we compare the performance of the standard AT, GAIRAT, and AT-SVM. We compare these methods to the best checkpoint model (suggested by Rice et al. [19]). Besides the accuracy of clean data, we evaluate models' adversarial robustness by FGSM attack, momentum iterative method (MIM) attack [44], and PGD attack. All these attacks are generated with 20 iterations and $\epsilon = 8/255$. In addition, we follow the suggestion of evaluating transfer attacks by Athalye et al. [14] to inspect whether the models trained by AT-SVM will cause the issue of obfuscated gradient and give a false sense of model robustness. We test the black-box PGD attack (PGD-b), in which we generate adversarial examples of CIFAR-10 from SAT models and evaluate their attack performance on the target mode. Moreover, we also use the ray searching attack (RayS) [45] to test the robustness of the models under hard-label or gradient-independent attack. For RayS, we set the maximum number of queries to 10,000. All experimental data are obtained in our implementation.

Compared with SAT, our AT-SVM improve adversarial robustness without degradation of clean accuracy, which alleviates the trade-off problem between accuracy and robustness. GAIRAT also achieves this effect, but we found that its robustness under black-box attacks is lower than under white-box attacks, which means there is an obfuscated gradient. Instead, our approach avoids this. In Table 1, the "Robustness" is the lowest accuracy under all attacks in the experiment to approximate the adversarial robustness.

Table 1. Test accuracy of ResNet-18 on CIFAR-10 dataset.

Methods	Clean	FGSM	MIM	PGD	PGD-b	RayS	Robustness
Natural	90.09	0	0	0	0	13.01	0
SAT	82.64	57.15	55.15	52.23	52.23	62.05	52.23
GAIRAT	78.84	62.62	62.30	60.47	54.19	54.17	54.17
AT-SVM	83.51	59.35	58.54	56.38	59.76	62.40	56.38

In Figure 3, we show some sample images from the CIFAR-10 dataset, with their corresponding adversarial examples. We find that the robust models trained by AT-SVM have strong interpretability. It can be found that a normal model can be attacked by adversarial perturbation not perceptible to humans, while the adversarial image of the AT-SVM model has obvious features of the misclassified class. This shows that images around the decision boundary of the robust model have features of both classes.

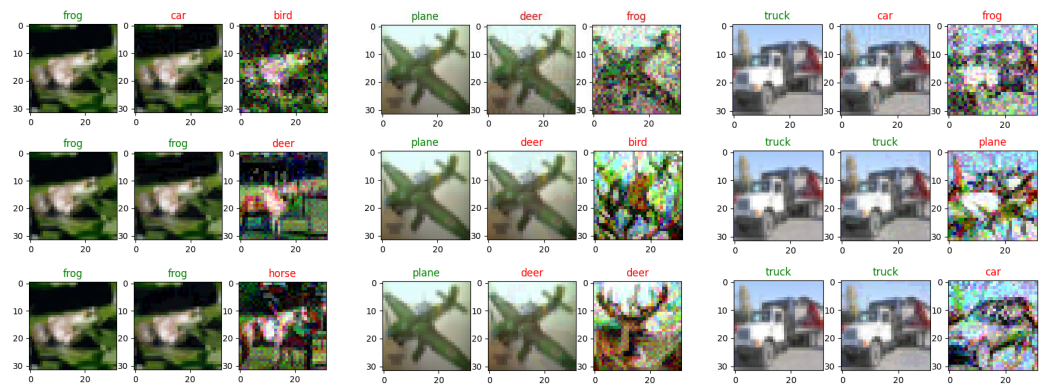


Figure 3. Sample images from the CIFAR-10 dataset, with corresponding adversarial examples. The first row represents the adversarial examples and predicted values obtained by the normal model, the second row represents the SAT model, and the third row represents the AT-SVM model. We show the original image and adversarial examples for epsilons of 0.03 and 0.3.

4.3. Effectiveness of SVM Batch Size

In theory, the SVM classifier should train all examples at the same time in order to find support vectors more accurately. However, such problems are difficult to optimize, so we use small batches to converge the training. Table 2 shows the influence of SVM batch size on the robustness of the model. k represents the ratio of the batch size of the SVM auxiliary classifier to that of the original network during training.

It can be seen that if k is too small, the effect of improving robustness is not obvious. When k is too large, the SVM auxiliary classifier will be difficult to train and the performance will decrease. The experiment found that set $k = 30$ is appropriate.

Table 2. Test accuracy of models training with different batch size ratios.

k	Clean	FGSM	MIM	PGD	PGD-b	Robutsness
1	81.37	56.92	54.99	51.13	59.42	51.13
5	84.64	58.42	56.42	54.17	61.06	54.17
10	84.02	58.83	57.22	54.59	60.26	54.59
20	83.27	60.30	58.91	56.11	59.62	56.11
30	83.51	59.35	58.54	56.38	59.76	56.38
40	82.25	58.14	56.94	53.28	58.20	53.28
50	81.99	58.55	57.26	53.99	58.38	53.99

4.4. AT-SVM with WideResNet

We use WideResNet-34-10 (depth 34 and width 10) [46] to evaluate the performance of our AT-SVM method in different networks. Other settings are the same as those in the experiment of ResNet-18. The results are shown in Table 3. We can get the same conclusion as ResNet-18, so AT-SVM is suitable for different networks.

Table 3. Test accuracy of WideResNet-34-10 on the CIFAR-10 dataset.

Methods	Clean	FGSM	MIM	PGD	PGD-b	Robustness
Natural	95.63	0	0	0	0	0
SAT	86.91	61.39	59.10	55.66	55.66	55.66
GAIRAT	82.70	62.24	62.19	60.65	56.71	56.71
AT-SVM	86.87	62.82	60.70	58.70	63.24	58.70

4.5. AT-SVM in SVHN

We also used ResNet-18 to conduct experiments on the SVHN dataset [47] to verify that our AT-SVM method is effective on different datasets. The initial learning rate is set to

0.01, and the other experimental settings are the same as the basic model. The results are shown in Table 4. We can achieve the same conclusion as CIFAR-10, so AT-SVM is suitable for different datasets.

Table 4. Test accuracy of ResNet-18 on the SVHN dataset.

Methods	Clean	FGSM	MIM	PGD	PGD-b	Robustness
Natural	96.32	0	0	0	0	0
SAT	91.08	68.06	62.42	57.49	57.49	57.49
AT-SVM	92.38	75.77	65.94	60.32	67.49	60.32

4.6. AT-SVM with RST

The experiments use the unlabeled data from [48] to enhance model training. Ref. [48] augment CIFAR-10 with 500K unlabeled images sourced from 80 Million Tiny Images and use robust self-training (RST) to train a more robust model. We combine the proposed method with the RST method and the other experimental settings are the same as the basic model. The results are shown in Table 5. The conclusion is that our proposed method can be combined with the RST method to obtain a more robust model.

Table 5. Test accuracy of models training with/without RST.

Methods	Clean	FGSM	MIM	PGD	PGD-b	Robustness
SAT	82.64	57.15	55.15	52.23	52.23	52.23
SAT+RST	85.21	60.31	58.17	54.53	63.84	54.53
GAIKAT	78.84	62.62	62.30	60.47	54.19	54.19
GAIKAT+RST	82.03	67.90	67.67	67.16	56.93	56.93
AT-SVM	83.51	59.35	58.54	56.38	59.76	56.38
AT-SVM+RST	81.68	59.72	59.53	58.34	60.06	58.34

4.7. AT-SVM with TRADES

We use ResNet-18 to evaluate the performance of our AT-SVM method under AutoAttack (AA) [49]. We use CW attack to generate adversarial examples at training time, and the other experimental settings are the same as the basic model. The results are shown in Table 6. It can be seen that although AT-SVM is more robust than SAT, TRADES [20] gets higher robustness. We state that our approach is not designed to defeat other existing defense methods, but rather a plug-and-play defense strategy. As shown in Table 6, adversarial training using both AT-SVM and TRADES gives better results. This proves that our algorithm can be integrated into other effective defense methods and make improvements.

Table 6. AutoAttack's test accuracy of ResNet-18 on the CIFAR-10 dataset.

Methods	Clean	AA
Natural	95.63	0
SAT	86.91	47.73
GAIKAT	82.70	32.19
AT-SVM	86.87	48.09
TRADES	78.89	48.73
AT-SVM&TRADES	81.04	49.49

5. Conclusions

This paper proposed a novel adversarial training method AT-SVM, which inserts an SVM auxiliary classifier in CNN. The input of the SVM auxiliary classifier was linearly separable features after passing through part of the original network. It can learn the decision boundary with the largest margin in this feature space. When training the original network, a new loss function obtained by the SVM auxiliary layer was used. In addition,

we used the SVM layer to select examples close to the boundary for training. Experiments show that conventional cross-entropy loss cannot constrain the margin while our method can effectively increase the margin of the network to improve the robustness against adversarial examples.

Author Contributions: Conceptualization, J.W., Z.X. and X.F.; methodology, J.W.; software, J.W.; validation, J.W., Z.X. and X.F.; formal analysis, J.W., Z.X. and X.F.; investigation, J.W.; writing—original draft preparation, J.W.; writing—review and editing, Z.X.; visualization, J.W.; supervision, J.W., Z.X. and X.F.; project administration, X.F.; funding acquisition, X.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research is partly supported by the Key Research and Development Program of Shaanxi (Program Nos. 2021ZDLGY15-01, 2021ZDLGY09-04, 2021GY-004, 2022ZDLGY06-07), and Shenzhen International Science and Technology Cooperation Project (No. GJHZ20200731095204013).

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: <https://www.cs.toronto.edu/~kriz/cifar.html> (accessed on 10 April 2022) and <http://ufldl.stanford.edu/housenumbers/> (accessed on 10 April 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the ICLR 2014, Banff, AB, Canada, 14–16 April 2014.
2. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. In Proceedings of the ICLR 2015, San Diego, CA, USA, 7–9 May 2015; Volume 1050, p. 20.
3. Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Šrndić, N.; Laskov, P.; Giacinto, G.; Roli, F. Evasion attacks against machine learning at test time. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 387–402.
4. Liu, Y.; Chen, X.; Liu, C.; Song, D. Delving into transferable adversarial examples and black-box attacks. In Proceedings of the ICLR 2017, Toulon, France, 24–26 April 2017.
5. Wang, X.; Ren, J.; Lin, S.; Zhu, X.; Wang, Y.; Zhang, Q. A unified approach to interpreting and boosting adversarial transferability. In Proceedings of the ICLR 2021, Virtual Event, 3–7 May 2021.
6. Finlayson, S.G.; Bowers, J.D.; Ito, J.; Zittrain, J.L.; Beam, A.L.; Kohane, I.S. Adversarial attacks on medical machine learning. *Science* **2019**, *363*, 1287–1289. [[CrossRef](#)] [[PubMed](#)]
7. Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; Song, D. Robust physical-world attacks on deep learning visual classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1625–1634.
8. Cohen, J.; Rosenfeld, E.; Kolter, Z. Certified adversarial robustness via randomized smoothing. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 1310–1320.
9. Balunovic, M.; Vechev, M. Adversarial training and provable defenses: Bridging the gap. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
10. Dhillion, G.S.; Azizzadenesheli, K.; Lipton, Z.C.; Bernstein, J.D.; Kossai, J.; Khanna, A.; Anandkumar, A. Stochastic Activation Pruning for Robust Adversarial Defense. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
11. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 582–597.
12. Yin, X.; Kolouri, S.; Rohde, G.K. Adversarial example detection and classification with asymmetrical adversarial training. In Proceedings of the ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020.
13. Yang, Y.; Zhang, G.; Katabi, D.; Xu, Z. ME-Net: Towards Effective Adversarial Robustness with Matrix Estimation. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 7025–7034.
14. Athalye, A.; Carlini, N.; Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 274–283.
15. Tramer, F.; Carlini, N.; Brendel, W.; Madry, A. On adaptive attacks to adversarial example defenses. *arXiv* **2020**, arXiv:cs.m/2002.08347.
16. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
17. Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; Madry, A. Robustness May Be at Odds with Accuracy. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.

18. Chen, T.; Zhang, Z.; Liu, S.; Chang, S.; Wang, Z. Robust overfitting may be mitigated by properly learned smoothening. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021; Volume 1.
19. Rice, L.; Wong, E.; Kolter, Z. Overfitting in adversarially robust deep learning. In Proceedings of the International Conference on Machine Learning, Virtual Event, 13–18 July 2020; pp. 8093–8104.
20. Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; Jordan, M.I. Theoretically Principled Trade-off between Robustness and Accuracy. In Proceedings of the ICML 2019, Long Beach, CA, USA, 9–15 June 2019.
21. Cheng, M.; Lei, Q.; Chen, P.Y.; Dhillon, I.; Hsieh, C.J. Cat: Customized adversarial training for improved robustness. *arXiv* **2020**, arXiv:cs.ml/2002.06789.
22. Wang, Y.; Ma, X.; Bailey, J.; Yi, J.; Zhou, B.; Gu, Q. On the Convergence and Robustness of Adversarial Training. In Proceedings of the ICML 2019, Long Beach, CA, USA, 9–15 June 2019; Volume 1, p. 2.
23. Zhang, J.; Zhu, J.; Niu, G.; Han, B.; Sugiyama, M.; Kankanhalli, M. Geometry-aware Instance-reweighted Adversarial Training. In Proceedings of the ICLR 2021, Virtual Event, 3–7 May 2021.
24. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 39–57.
25. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. Deepfool: A simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2574–2582.
26. Cai, Q.Z.; Liu, C.; Song, D. Curriculum adversarial training. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 3740–3747.
27. Wang, Y.; Zou, D.; Yi, J.; Bailey, J.; Ma, X.; Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
28. Zhang, J.; Xu, X.; Han, B.; Niu, G.; Cui, L.; Sugiyama, M.; Kankanhalli, M. Attacks which do not kill training make adversarial learning stronger. In Proceedings of the International Conference on Machine Learning, Virtual Event, 13–18 July 2020; pp. 11278–11287.
29. Goyal, S.; Qin, C.; Uesato, J.; Mann, T.; Kohli, P. Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples. *arXiv* **2020**, arXiv:cs.ml/2020.03593.
30. Mustafa, A.; Khan, S.; Hayat, M.; Goecke, R.; Shen, J.; Shao, L. Adversarial defense by restricting the hidden space of deep neural networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 3385–3394.
31. Taghanaki, S.A.; Abhishek, K.; Azizi, S.; Hamarneh, G. A kernelized manifold mapping to diminish the effect of adversarial perturbations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 11340–11349.
32. Bai, Y.; Zeng, Y.; Jiang, Y.; Xia, S.T.; Ma, X.; Wang, Y. Improving adversarial robustness via channel-wise activation suppressing. In Proceedings of the ICLR 2021, Virtual Event, 3–7 May 2021.
33. Zoran, D.; Chrzanowski, M.; Huang, P.S.; Goyal, S.; Mott, A.; Kohli, P. Towards robust image classification using sequential attention models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9483–9492.
34. Yan, H.; Zhang, J.; Niu, G.; Feng, J.; Tan, V.Y.; Sugiyama, M. CIFS: Improving Adversarial Robustness of CNNs via Channel-wise Importance-based Feature Selection. *arXiv* **2021**, arXiv:physics.soc-ph/0803.4058.
35. Xie, C.; Wu, Y.; Maaten, L.V.D.; Yuille, A.L.; He, K. Feature denoising for improving adversarial robustness. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 501–509.
36. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
37. Platt, J. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Report MSR-TR-98-14. 1998. Available online: <https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/> (accessed on 10 April 2022).
38. Elsayed, G.; Krishnan, D.; Mobahi, H.; Regan, K.; Bengio, S. Large Margin Deep Networks for Classification. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 842–852.
39. Garg, A.; Har-Peled, S.; Roth, D. On generalization bounds, projection profile, and margin distribution. In Proceedings of the International Conference on Machine Learning (ICML), Sydney, Australia, 8–12 July 2002; pp. 171–178.
40. Langford, J.; Shawe-Taylor, J. PAC-Bayes & margins. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2003; pp. 439–446.
41. Ding, G.W.; Sharma, Y.; Lui, K.Y.C.; Huang, R. MMA Training: Direct Input Space Margin Maximization through Adversarial Training. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
43. Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*; Tech. Rep. 1; Computer Science Department, University of Toronto: Toronto, ON, Canada, 2009.
44. Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; Li, J. Boosting adversarial attacks with momentum. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9185–9193.

45. Chen, J.; Gu, Q. RayS: A Ray Searching Method for Hard-label Adversarial Attack. In Proceedings of the 26rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Virtual Event, 6–10 July 2020.
46. Zagoruyko, S.; Komodakis, N. newblock Wide Residual Networks. *arXiv* **2016**, arXiv:cs.CV/1605.07146.
47. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A.Y. Reading digits in natural images with unsupervised feature learning. In Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011, Granada, Spain, 12–17 December 2011.
48. Carmon, Y.; Raghunathan, A.; Schmidt, L.; Liang, P.; Duchi, J.C. Unlabeled data improves adversarial robustness. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 1192–11203.
49. Croce, F.; Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In Proceedings of the International Conference on Machine Learning, Virtual Event, 13–18 July 2020; pp. 2206–2216.