Journal of Network Intelligence ©2022 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **7**, Number **4**, November **2022**

SFM-Defence: Filtering Adversarial Perturbation by Sparse Feature Masker

Tingyue Yu

School of Cyberspace Science
Harbin Institute of Technology
92 Xidazhi Street, Harbin, 150000, China
xklsbl@126.com

Shen Wang*

School of Cyberspace Science
Harbin Institute of Technology
92 Xidazhi Street, Harbin, 150000, China
wangshenhit@163.com

Zhenbang Wang

State Grid Heilongjiang Power Co., Ltd
301 Hanshui Road, Harbin, 150000, China
wangzhenbang_power@163.com

Shigang Tian

State Grid Heilongjiang Power Co., Ltd
301 Hanshui Road, Harbin, 150000, China
tianshigang_power@163.com

Xiangzhan Yu

School of Cyberspace Science
Harbin Institute of Technology
92 Xidazhi Street, Harbin, 150000, China
yuxiangzhan_hit@163.com

*Corresponding author: Shen Wang

Received July 25, 2022, revised September 30, 2022, accepted October 27, 2022.

Abstract. *Among the researches on the security problem of adversarial examples, it is an important way to prevent adversarial examples by inhibiting the impact of adversarial perturbation. However, those existing defense methods based on image pre-processing or image reconstruction cannot achieve a satisfactory balance in terms of time complexity and defense effect. In order to solve this problem, we first qualitatively analyze the reasons why the defense methods based on succinct image pre-processing cannot achieve good performance. On this basis, an adversarial examples defense method based on non-robust feature inhibition combined with traditional image pre-processing methods is proposed in this paper, which is called SFM-Defense. It can eliminate redundant semantic information by training a sparse feature masker, so as to compress the features that can be used for attacks. The experimental results on CIFAR10, SVHN and Tiny-ImageNet show that the proposed method can achieve competitive defense performance with the existing SOTA method on the black box threat model, which can surpass the existing methods.*

Keywords: Deep learning, Adversarial examples, Adversarial perturbation filtering.

1. **Introduction.**

In recent years, with the continuous improvement of software and hardware performance, as well as the emergence of new theories and technologies, deep learning has played an important role in many fields of artificial intelligence. In many fields concerned by traditional machine learning technology, deep learning technologies have shown significant performance progress, such as image recognition [1], speech recognition [2], recommendation system [3] and machine translation [4]. In many related fields, deep learning has even achieved better performance than human beings. However, recent researches have shown that, when deep learning models are deployed in an adversarial environment, its security cannot be fully guaranteed. For several key points of the deep learning model, including training data [5], network parameters [6], test data [7] and model output [6], different types of security threats have been found.

Among the threats on deep learning models, the research on adversarial examples has become one of the central themes of the security research on deep learning models, the reason of which is that compared with other threats, adversarial attacks do not require to obtain privacy information which is difficult to obtain or affect in most cases, including training data and network parameters. In an ideal situation, the attacker can mislead the normal operation of the target deep learning model by adding adversarial perturbation that is difficult to detect and distinguish by human beings to the input samples, even with little knowledge of the target model. Recently, the adversarial examples threat has changed from a pure theoretical problem to an important security threat to the artificial intelligence system deployed in reality, as deep learning technology has been widely used in many security sensitive fields, such as face recognition [8], autopilot [9] and malicious detection [10]. This brings a very urgent motivation to enhance the security of deep learning system oriented to adversarial examples.

Briefly, the defense methods of adversarial examples are mainly divided into two categories: adversarial examples detection and adversarial robustness enhancement. The kind of method attempts to distinguish adversarial examples to clean samples by extracting the statistical difference between clean samples and adversarial samples, so as to directly refuse the detected adversarial examples. However, the existing detection methods cannot achieve the correct classification of adversarial examples, which limit the defense effect obviously, so it can only be used as an auxiliary defense measure. Therefore, most of the current works still focus on enhancing the adversarial robustness, which is to improve the recognition ability of adversarial examples by modifying the training process or inference process of the model.

In recent studies, adversarial training [11] and its latest variant [12] are considered the most effective way to improve adversarial robustness, in which the defender actively generates adversarial examples and adds them to the training data, so that the defended model can learn how to correctly identify adversarial examples. However, the effectiveness of adversarial training often depends on the intensity of active adversarial attacks, making the time cost of adversarial training that achieve high adversarial robustness much higher than that of the standard training process [13], which makes it difficult to deploy adversarial training on large datasets and models. Other works based on changing the training process often have similar problems [14, 15]. Therefore, other researchers choose to eliminate the negative impact caused by adversarial perturbation through denoising or other pre-processing operations on the adversarial examples, so as to ensure the correct classification of adversarial examples.

Up to present, it is difficult to balance the time cost and defense effect of the adversarial examples defense method based on input transformation. On the one hand, although the defense method based on simple image pre-processing hardly increases the time cost of the interface stage, its defense effect on color image datasets is very limited. On the other hand, the defense method based on complex image reconstruction algorithms can achieve a good defense effect, but it often cannot guarantee the real-time performance of the defended models, which makes it difficult to be applied in practical applications. However, we believe that the potential of image pre-processing in adversarial examples defense has not been fully exploited. Thus we first analyze the reasons why the existing defense methods based on simple image pre-processing are difficult to work, and then assume that by inhibiting non-robust features, simple image pre-processing methods can achieve considerable defense effects.

Based on our assumption, an adversarial examples defense method based on sparse feature masker is proposed in this paper, which is called SFM-Defense. This method design and implement a sparse feature masker to inhibit the redundant features in the input samples. Combined with a simple image pre-processing method, it can realize the defense against the adversarial perturbation. The main contributions of this paper are as follows:

1) Based on the analysis of the limitations of existing defense methods, a new defense framework for adversarial examples is proposed in this paper. The framework combines non-robust feature inhibition and simple image pre-processing methods to filter the impact of adversarial perturbation.

2) As the core of the proposed framework, a training method of sparse feature masker is proposed. By balancing the three loss functions including sparsity loss, semantic loss and reconstruction loss in the training process, the proposed sparse feature masker can succeed to inhibit the redundant information in the input image while ensuring the normal operation of the defended models.

3) The experimental results on CIFAR10, SVHN and Tiny-ImageNet show that the performance of the proposed method matches the existing SOTA method under the black box attacks, and exceeds the performance of the existing methods under the gray box attacks. What's more, the proposed method provides the ability on resisting adaptive attacks to a certain extent.

2. **Related Works.** In the task of image classification concerned in this paper, the attacker can make the image classifier based on deep learning models misclassify the input image with random or pre-selected categories by adding perturbation to the clean images sampled from the training distribution. This kind of attack is called adversarial attack, the added malicious perturbation is called adversarial perturbation or adversarial noise, and the malicious samples generated for attack are called adversarial examples. Theoretically, adversarial attacks are formalized to solve the optimization problems in Eq.1:

$$\begin{aligned} &\arg\min_{\delta x} \|\delta x\|_p \\ &\text{s.t.}\ \ F(x+\delta x) \neq y \end{aligned} \tag{1}$$

where $F|(\cdot|)$ is the image classifier based on deep learning, x is the clean sample for attack, and y is the ground true category label of clean sample x.

In order to resist the threat of adversarial attacks, Goodfellow et al. [11] proposed adversarial training, in which adversarial perturbation is actively generated and added to the training data, so that the model can learn how to correctly handle adversarial examples. In [13], the adversarial training is formalized as a max-min dual optimization problem. It is declared that the defense effect of adversarial training depends on the effectiveness of the adversarial examples generated in the internal loss maximization problem. On this basis, [16, 17, 18] expand the work of [11] from different angles. In [12], the latest variant of adversarial training called Trades is proposed, which can achieve a balance between standard generalization and robust generalization.

However, on the one hand, adversarial training and its variants still cannot achieve satisfactory performance in the classification of adversarial examples, and there is a big gap between the robust risk of the training set and the test set. On the other hand, adversarial training methods that achieves high adversarial robustness need to generate strong adversarial examples through iterative optimization processes, which makes the time cost of adversarial training often increase by nearly 10 times compared with standard training methods. Therefore, more and more works have focused on eliminating the possible adversarial perturbation in the input samples to enhance the ability to resist the threat of adversarial examples in the interface process.

In the early works on eliminating adversarial perturbation, adversarial perturbation is regarded as a kind of high-frequency noise in the legal samples. They try to eliminate the influence of malicious noise in the input image by image denoising or pre-processing, so as to restore the adversarial examples to the legal category. This kind of works mainly rely on the use of conventional or artificially designed feature processing algorithms to transform the input image, including color bit depth compression, local smoothing, non-local smoothing filtering [19], layered feature denoising [20], JPEG compression (feature distillation) combined with adaptive quantization [21]. Although the defense methods based on simple image preprocessing hardly increase the time cost of interface, their defense effect is very limited. To solve this problem, other researchers attempt to learn the prior distribution of legal samples, so as to map the adversarial examples to the manifold of legal samples. This kind of method can recover adversarial examples from the semantic level rather than the signal level, so it can obtain a better recovery effect in theory.

The PixelDefense [22] proposed by Song trains a variant of the PixelCNN model to learn the distribution of legitimate training data, and search for the nearest point on the training distribution for a possible adversarial example through greedy algorithm, so as to filter the impact of adversarial perturbation in the representation space. The DefenseGan [23] proposed by Samangouei adopts a similar filtering framework to [22], but they learn the distribution of legitimate training samples by training a GAN. The ComDefend [24] proposed by Jia inputs the adversarial examples into the compressed CNN to eliminate the adversarial perturbation while maintaining the structure information of the original image.

In recent research, Dai proposed the adversarial examples reconstruction method based on the Deep Image Prior (DIP) [25] to minimize the reconstruction error through the reconstruction network of DIP, so as to reconstruct the adversarial examples into legal samples. Although the performance of this method is outstanding, its time cost is very large, which makes it difficult to be applied in the actual scene. Zhao [26] divided the image pixels into high sensitivity points and low-sensitivity points, and respectively uses the norm based total variation image smoothing algorithm and low-rank image completion algorithm to eliminate the adversarial perturbation. This method has achieved a good balance between time cost and defense effect, but its performance still has some room for improvement.

Generally speaking, the defense methods based on image reconstruction can achieve a good defense effect, but most of them may take much more time to reconstruct an image than the normal interface process, resulting in the inability to guarantee the real-time performance of the defended model. Therefore, in view of the problems of the two kinds of existing methods, we try to propose an effective defense method against adversarial perturbation without significantly increasing the time complexity of the defended model.

3. **Framework of proposed method.**

3.1. **Theoretical motivation.** Reviewing the existing works of filtering adversarial perturbation to achieve the correct classification so as to defend against adversarial examples, they can be roughly divided into three categories, including methods based on image preprocessing, manifold mapping, and end-to-end filters. The first kind of early works regard the adversarial perturbation as the abnormal noise in the legal samples, and then they eliminate the influence of noise through denoise on the input images. Although the implementation of this kind of method is very simple and hardly increases the computational cost of the model, empirical research also shows that it can hardly achieve satisfactory defense effects on complex datasets, such as CIFAR10 and ImageNet. In this paper, we first attempt to analyze the reasons why this kind of method cannot achieve satisfactory results, and then we propose an improved filtering method on this basis.

For a potential adversarial input sample, the defense method based on image preprocessing first inputs it into a preprocessing function, which often removes the adversarial perturbation by image filtering, reconstruction and other measures, then the processed samples are input into the target model for normal classification. However, Geirhos et al. [27] pointed out that deep learning models tend to rely on texture features rather than shape for classification. This means that adversarial perturbation often changes the characteristics of legal samples in the unit of local invisible textures rather than independent individual pixels. It can be found that though image preprocessing methods can destroy uncorrelated adversarial pixels, it is quite difficult to destroy adversarial texture features from a global perspective. Weakening all pixels on the adversarial textures doesn't mean that the corresponding adversarial features can be inhibited. The third row of Figure 1 shows the visualization of the adversarial perturbation generated by the attacker when using the defense based on image preprocessing. It is easy to find that in this case, the generated adversarial perturbation has obvious semantic features, and simple image preprocessing operations are difficult to destroy such adversarial features. But when no defense is performed, as shown in the second row, the adversarial perturbation is similar to random noise, which is easily destroyed by the image processing measures.

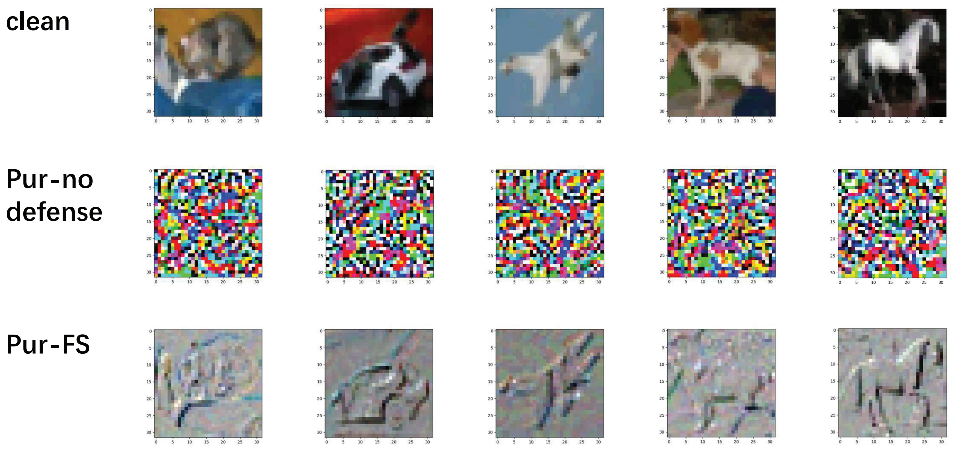

FIGURE 1. visualization of perturbation with no defense (second row) and with image pre-processing defense (third row) on CIFAR10

Ilyas et al. [28] also pointed out that adversarial examples are not bugs in the deep learning models, but inherent attributes of features in the data distribution. They divided the features in the data distribution into robust features that can resist adversarial perturbation and non-robust features that have a large amount of redundant information and cannot resist adversarial perturbation. We believe that the reason for the limited effect of defense methods based on image preprocessing is that such methods can only eliminate the adversarial noise on robust features, but cannot effectively eliminate the adversarial noise on non-robust. By modifying a large amount of redundant semantic information, the attacker can produce adversarial features causing misclassification which cannot be distinguished by the naked eye in a global sense.

Therefore, if most of the redundant information in the adversarial image is removed, the influence of non-robust features can be fully inhibited, so that the adversarial perturbation filtering method based on image preprocessing can also play a sufficient defense effect. Based on the above analysis, in order to achieve effective adversarial examples defense, a method that can effectively inhibit non-robust features is first proposed in this paper. Combined with simple image pre-processing methods, it can be used to eliminate the negative impact caused by adversarial examples.

3.2. **Methodological framework.** Although a method of extracting robust features through iterative search has been proposed in [28], this method is difficult to be used in the actual case due to the high time cost, similar to [25]. Meanwhile, other works of reconstructing adversarial examples by training end-to-end models adopt completely different design ideas. Although they indirectly limit the influence of non-robust features, they cannot directly inhibit non-robust features, so the defense effect is relatively limited. This paper attempts to propose a method to inhibit non-robust features by eliminating redundant information in input samples, which is called SFM-Defense. Combined with some image preprocessing methods of which the defense effect has been proven to be insufficient in experience, the proposed method can effectively defend against adversarial examples. The framework of the proposed method is shown in Figure 2.

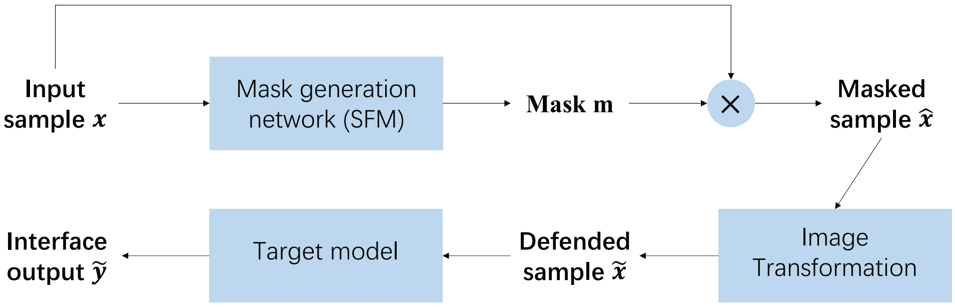
Input sample x
Mask generation network (SFM)
Mask m
×
Masked sample $\hat{x}$
Image Transformation
Defended sample $\tilde{x}$
Target model
Interface output $\tilde{y}$

FIGURE 2. Framework of proposed method

For a potential adversarial image $x \in \mathbb{R}^{C\times W\times H}$, it is first input into the mask generation network $G(\cdot)$ to produce a corresponding mask $m = G(x)$. Each element in $x$and m is restricted to [0,1]. Then the pixels on each color channel on the input sample x are multiplied pixel-wised with mask m to obtain the masked sample $\widetilde{x} \in \mathbb{R}^{C\times W\times H}$. It should be noted that for an input image with C color channels, a mask with only one single channel is produced so as to reduce the training complexity of the mask generation network and prevent it from excessively interfering with the normal operation of the model. The masked sample is input into the image pre-processor to obtain the defensed sample that finally eliminates the impact of adversarial perturbation. The defensed image is input into the target classification network for final classification.

It is easy to find that the core content of this paper is to train the mask generation network which is used to remove the redundant information in the image and inhibit the non-robust features in the data. We will first introduce the training objectives of the mask generation network, and then describe the training process and implement details of the defense model.

4. Loss function of sparse feature masker.

In order to eliminate the redundant information in the input samples and inhibit the non-robust features, in this paper a novel masker generation network needs to be designed. We propose a mask generation network, which is called sparse feature masker (SFM), to achieve the following two design goals: First, it should make the output samples as sparse as possible, that is, the output samples retain as little pixel information as possible. Then on the basis of output sparsity, it's designed for retaining the important semantic information in the original sample, so that it does not seriously affect the normal classification of the target model.

The phrase 'sparse feature' means that, unlike the traditional generation models which mainly focus on learning (sparse) representations of input samples, the training goal of SFM is to remove redundant information and make the input features sparse. At the same time, unlike the existing adversarial perturbation filtering methods based on adding restorative noise, SFM masks the redundant features in the input samples by multiplying the generated mask and the input samples. In addition, it should be noted that a variant of Auto-encoder is called Sparse Auto-Encoder (SAE), but its purpose is to generate sparse representations rather than sparse reconstruction samples, which is different from the design objective of SFM. To sum up, the training goal of SFM proposed in this paper is to minimize the following loss function in Eq.2:

$$L_{mask} = w_1 \cdot L_{sp} + w_2 \cdot L_{seg} + w_3 \cdot L_{recon} \tag{2}$$

The loss function of SFM is composed of three components, and their weights are adjusted by three different hyper-parameters, including $w1$, $w2$ and $w3$. L_{sp} is the sparsity loss. Minimizing L_{sp} will make the generated masked samples retain as few brightness as possible in the original images, so as to achieve the purpose of feature sparsity. L_{seg} is the semantic loss, which is used to guide the parts of the mask that should be retained and discarded, by the standard that SFM should remove redundant information and retain features that make an important contribution to normal classification. L_{recon} is the reconstruction loss, which is used to moderate the decrease of sparsity loss and prevent SFM from generating a mask of all 0. The following contents will discuss the design of the three main components of the loss function.

4.1. Design of sparsity loss.

In order to retain as few pixels or pixels as possible in the masked samples, an intuitive choice is to limit the L_0 norm of the masked samples, that is, to minimize the number of non-zero elements in the masked samples:

$$\|\widetilde{x}\|_0 = \#|(i \mid \widetilde{x}_i \neq 0|)$$

However, as the regularization of L_0 norm is a dis-continuous and non-convex process, it cannot be solved directly by optimization, and should traverse all feasible solutions on paper. Therefore, this problem is NP-hard, which means that L_0 norm cannot be directly used in designing sparsity loss.

However, according to the theory in [29], L_1 norm is the optimal convex approximation of L_0 norm. By minimizing the L_1 norm of the masked sample, the goal of minimizing its L_0 norm can be approximately achieved. Therefore, as an alternative, we select the L_1 norm of the masked samples as the sparsity loss, that is, minimize the sum of the absolute values of each pixel in the masked sample. Therefore, the sparsity loss L_{sp} is defined according to Eq.3

$$L_{\mathrm{sp}} = |||\widetilde{x}|||_1 = \sum_{i=1}^{C\times H\times W} ||\widetilde{x}_i|| \tag{3}$$

4.2. **Design of semantic loss.** In addition to making the masked samples retain as little brightness or pixels as possible, another pivotal task is to make the masked samples retain as much important semantic information as possible, so that it can be correctly recognized by the classification model. An intuitive idea is to take the classification error of the masked samples on the target model as the semantic loss, which can be measured by the Cross-Entropy loss (CE loss) commonly used in the training process of classifiers. However, the main problem with using CE loss guide SFM training is that the generator may overfit the features related to classification. Although such retained features can also make the masked samples correctly classified, they may not belong to the features in the original samples, but produce new non-robust features similar to adversarial perturbation.

The possible reason for this phenomenon is that although both robust and non-robust features in the data have obvious contributions to model classification, non-robust features can provide richer information related to categories, which is essentially the over-fitting of the target model to non-robust features [28]. For an SFM based on classification error trained from scratch, it will tend to learn features that can reduce the classification error as soon as possible, that is, non-robust features related to the correct category. Therefore, the masked samples generated by SFM become a special kind of adversarial examples, which is "adversarial examples aiming at the correct category". It will lead to the fact that SFM cannot really be used to extract important semantic information from the original input samples.

In order to solve this problem, rather than the classification error measured by CE loss, the (negative value of) cosine similarity between the outputs of the original sample and the masked sample on the target model is selected in this paper to be the semantic loss of SFM. So L_{seg} is defined according to Eq.4:

$$L_{seg} = 1 - \cos(F(x), F(\widetilde{x})) = 1 - \frac{F(x)\cdot F(\widetilde{x})}{|||F(x)|||_2|||F(\widetilde{x})|||_2} \tag{4}$$

The reasons for using cosine similarity as semantic loss are as follows. Firstly, making the output vector of the original sample and the masked sample more similar to the target model, rather than only using the information related to the category label, can retain richer semantic information in the original sample. Secondly, by preventing the category labels from directly participating in the training of SFM, we can effectively avoid the over-fitting of the masked samples to the non-robust features related to the correct category. Finally, cosine similarity is more appropriate than Euclidean distance with a similar effect, because the existence of L_{sp} makes original samples and masked samples have different distributions, and their eigenvectors may have different norms. Compared with Euclidean distance, cosine similarity makes features belonging to different distributions align with each other through normalization, so that the training process of the model will become more stable.

4.3. **Design of reconstruction loss.** Intuitively, sparsity loss and semantic loss are enough to achieve the design goal of SFM. However, in practice, only using the above two loss functions would tend to output masked samples with all pixels being 0s. The fundamental reason is that sparsity loss and semantic loss are contradictory in nature. With the decrease of sparsity loss, the generated masked samples will become sparser, and less semantic information will be retained. At the beginning of SFM training, the generated mask is almost random, so it is difficult to quickly find the training direction to reduce semantic loss. At this time, sparsity loss occupies the dominant position of training loss. As a result, the sparsity loss will quickly decrease to zero and make SFM generate masked samples with all 0s. At this time, although the sparsity loss would no longer decrease, the vanishing gradient phenomenon caused by the zeroes output by target models makes the model unable to decrease the semantic loss through gradient descent. Finally, the trained SFM will only generate masked samples with all 0s or with no semantics.

In order to solve this problem, we need to introduce an additional loss to mitigate the decrease of sparsity loss, which is defined as reconstruction loss L_{recon}. L_{recon} is defined as the Euclidean distance between the original input sample and the masked sample, as shown in Eq.5.

$$L_{recon} = |||x - \widetilde{x}|||_2 = \sqrt{\sum_{i=1}^{C\times W\times H} (x_i - \widetilde{x}_i)^2} \tag{5}$$

Minimizing the reconstruction loss means making the masked sample as close to the original sample as possible, that is, increasing the L_2 norm of the mask generated by SFM as much as possible. In the training process, although the sparsity loss still plays a leading role, the existence of reconstruction loss makes the model not rush to the completely sparse state quickly, so that the semantic loss can get a space to play a guiding role. Balancing the above three losses can make the masked samples sparse and retain the necessary semantic features.

5. Detailed Implement of SFM-Defense.

After introducing the training loss of the sparse feature masker (SFM), we can comprehensively describe the complete working process of the proposed defense method, SFM-Defense. The working process of the proposed method is mainly divided into three stages, including the target model pre-training stage, the defense model training stage and the defense model operation stage. The working process and relevant implementation details of the three stages are introduced in detail below.

5.1. Pre-training of the target model.

Since the training process of SFM includes semantic loss which is to minimize the output changes of the masked samples on the target model, if the target model can extract the semantic information of the input samples more accurately before the training process of SFM, SFM can learn how to retain the semantic information more quickly, which can speed up the training convergence of SFM. Therefore, before the defense model starts training, we choose to carry out several epochs of pre-training on the target model at first. The pre-training process is completely consistent with that of the ordinary deep neural network, so it will not be introduced in detail.

5.2. Training process of the Defense Model.

The training process of SFM is the pivotal part of the proposed method, and its process is shown in Algorithm 1. The objective of the training phase of the defense model is to obtain an SFM that can effectively sparse the input samples while retaining the semantic information of the input samples. A simple training strategy is to train the target model completely at first, then freeze the parameters of the trained target model, and then train SFM independently. But in practice, through this simple strategy, we cannot get the SFM that achieves the objective. Specifically, using a target model with fixed parameters to guide the training of SFM will make it quickly fall into local minima under the joint action of semantic loss and reconstruction loss, which means that it tends to generate masks with all positions of 1.

Algorithm 1: Training process of sparse feature masker

```
Input: Data: Training Set; $w_1, w_2, w_3$: Hyper-parameters for proposed method; $E$: total
       epoches of training; $T(\cdot)$: Image preprocessing method.
Output: $F_\theta$: Trained classifier; $M_\eta$: Trained SFM.
for $e < E$ do
    // Training for classifier
    for sample $(x, y)$ as a mini-batch of Data do
        $x = T|(x|)$
        $L_c = L_{ce}|(F_\theta|(x|), y|)$
        Updating $F_\theta$ using SGD by descending $L_c$.
    end
    // Training for SFM
    for sample $(x, y)$ as a mini-batch of Data do
        $x = T|(x|)$
        $\widetilde{x} = M_\eta(x)$
        $L_{sp} = |||\widetilde{x}|||_1 = \sum_{i=1}^{C\times H\times W} ||\widetilde{x}_i||$
        $L_{seg} = 1 - \frac{F(x)\cdot F(\widetilde{x})}{|||F(x)|||_2 |||F(\widetilde{x})|||_2}$
        $L_{recon} = \sqrt{\sum_{i=1}^{C\times W\times H} (x_i - \widetilde{x}_i)^2}$
        $L_{mask} = w_1 \cdot L_{sp} + w_2 \cdot L_{seg} + w_3 \cdot L_{recon}$
        Updating $M_\eta$ using SGD by descending $L_{mask}$.
    end
end
```

In this case, the sparse loss will lose its guiding role in SFM training, and the model will lose the ability to sparse the input samples.

Therefore, we adopt an improved training strategy to avoid the above problems, where the target model and the SFM are trained alternately. In this way, in the early stage of the training process of SFM, the target model will not produce fixed output temporarily, and the sparse loss will dominate the training loss, which will make SFM tend to learn the mask for sparse output at first. After that, in the later stage of training, when the output of the target model is relatively fixed, the sparse loss and reconstruction loss are in a relatively balanced stage, and the semantic loss will dominate in turn, making SFM learn how to retain the semantic information of the input samples.

Another problem that needs to be noticed is that whether in the training process of SFM or target model, it is necessary to carry out pre-processing on the training data, and the pre-processing method used is recommended to be consistent with that used in the test phase. As the image pre-processing method will be used to filter the input image in the test stage, if the same pre-processing process would not used in the training stage, it will cause inconsistency between the training distribution and the test distribution, which would create a distribution shift problem and affect the accuracy of the defended model to clean samples.

5.3. **Interface process of defended Model.** After the training of the target model and SFM, the defense model can be used for defending adversarial examples. First, the input samples are sent to the SFM to generate the sparse mask, and the sparse mask and the input samples are multiplied by the color channel to obtain the masked samples. Then a specified image pre-processing method is applied to the masked samples to obtain the final robust samples that can be classified correctly. In order to facilitate comparison, the simple image pre-processing methods used for defense in [19] are selected in this paper, including color bits squeezing, local smoothing (local median filtering) and non-local Smoothing (global mean filtering). The image pre-processing methods involved are briefly introduced below.

1) Color bit depth squeezing

As a differentiable model, a neural network always assumes that the input space is continuous. However, digital computers only support discrete data representation as an approximation of continuous natural data. Two common styles of the color standard are used in the image classification datasets, which are 8-bit grayscale images and 24-bit color images. An 8-bit grayscale image provides $2^8 = 256$ possible values for each pixel, where 0 is black, 255 is white, and the middle number represents different gray shadows. Color images expand the 8-bit gray image to contain the color information of red, green and blue channels respectively. Therefore, each pixel of the color image provides more than $2^{24} \approx 1600$ different colors. Xu et al. [19] assume that reducing the bit depth of the image can reduce the success rate of adversarial perturbation without damaging the accuracy of the classifier.

Specifically, it is often difficult for the human eye to distinguish between the original image with 8-bit color per channel and the image with only 4-bit color depth. Therefore, the original 8-bit image (with each RGB channel) can be reduced to fewer bits without significantly reducing the human image recognition rate. In order to reduce a color channel from 8 bits to i-bit depth ($1 \leq i \leq 7$), we first multiply the input value by 2^{i-1} (and then subtract 1), and then round it to an integer. Next, we scale the resulting integer to $|[0, 1|]$ and divide it by 2^{i-1}. Through the integer rounding operation, the information capacity of the representation is reduced from 8 bits to i bits.

Theoretically, this strategy can obviously narrow the color space that attackers can use, so as to decrease the feasible region of adversarial examples and reduce the success rate of adversarial attacks. However, in practice, this strategy plays a certain role in filtering adversarial perturbation on grayscale images, represented by MNIST, but it has little defense effect on color image datasets, such as CIFAR10 and Tiny-ImageNet. This may be due to the fact that gray-scale images have a small color space originally. After bit depth squeezing where the available color space of an image is further compressed, it is more difficult for adversarial attacks to find a feasible solution. However, for color images with relatively large color space, even after bit depth squeezing, the color space that attackers can search in is still large enough to ensure the success rate of adversarial attacks.

2) Local median filtering

The local median filter runs a sliding window on each pixel of the image, in which the central pixel is replaced by the median of adjacent pixels in the window. Unlike bit depth squeezing, median filtering does not reduce the number of pixels in the image, but spreads the pixel values to nearby pixels. Median filtering is particularly effective in removing sparse black-and-white pixels in the image (like salt and pepper noise), while preserving the edge of the object well. As the pixel values are averaged in a certain part, local median filtering essentially limits the number of features that attackers can use by making adjacent pixels more centralized.

In practice, median filtering has an obvious filtering effect on the adversarial perturbation generated by the attack constrained by L_2 norm. The characteristic of this kind of perturbation is that the perturbation amplitude is relatively small locally, but the number of disturbed pixels is large, which is widely distributed in the whole image. However, its defense effect on L_∞ constrained attack is limited. At the same time, due to the existence of adaptive attacks, median filtering has little defense effect against adversarial perturbation in the white box threat model.

3) Global mean filtering

Global mean filtering is a kind of non-local smoothing method. Unlike local smoothing, non-local smoothing methods smooth similar pixels in a larger area, not just adjacent pixels. For a given image block, non-local smoothing methods find several similar blocks in a large range of the image, and then replace the middle block with the average value of these similar blocks. Assuming that the average value of the noise is zero, averaging similar small blocks can cancel the noise while preserving the edge of the object in a wide region of the image. In practice, the fast non-local denoising method implemented in OpenCV is used. It first converts the color image into CIELab color space, then denoises its L and AB components respectively, and then converts it back to RGB space.

From experience, the defense effects of the global mean filter and local median filter are very similar, and both of them have obvious defense effects against the wide-range and small-scale perturbation caused by L_2 norm constrained attacks, while it is difficult to defend against the large-scale perturbation caused by L_∞ norm constrained attacks. These two methods also have similar weaknesses in the white box threat model, that is, they are completely unable to resist adaptive attacks.

6. Experiment.

6.1. Experimental settings.

(1) Datasets and models

In this paper, experiments are carried out on CIFAR10[30], SVHN[31] and Tiny-ImageNet [32] datasets. The following describes the experimental settings in different datasets.

CIFAR10 is a universal item dataset with 50000 training samples and 10000 test samples, in which each sample contains $32 \times 32 \times 3$ pixels and 10 categories of real labels. The experiment in the white box environment uses ResNet18 as the defended classifier [33], and the experiment in the black box environment uses another independently trained ResNet18 as the source model.

SVHN is a house number dataset, including 73257 training samples and 26032 test samples, each of which contains $32 \times 32 \times 3$ pixels and a real label from 10 categories. The experiment in the white box environment uses ResNet18 as the classifier [33], and the experiment in the black box environment uses another independently trained ResNet18 as the source model.

Tiny-ImageNet is an item identification dataset with 200 categories, each of which has 500 training images, 50 verification images and 50 test images. It is a subset of ILSVRC, and each sample contains $64 \times 64 \times 3$ pixels. Resnet50[33] is used as the target model of the white box experiment, and another model with the same architecture is used as the source model of the experiment under the black box setting.

In addition, on all of the above three datasets, we use Attention Unet [34] as the backbone architecture of the mask generator. Attention Unet is one of the most important improved versions of Unet[35], which is usually used in the field of image segmentation.

(2) Threat models

The proposed method is tested by attacks on the black box, gray box and white box threat models. In the black box environment, adversarial examples of the test data on an alternative model are generated, and they are input into the target defended model. The ability to resist black box attacks is measured by the average accuracy of the adversarial examples on the target model. It should be noted that the black box setting in this experiment is stricter than the general black box setting, where attackers can obtain the parameters of the target model to obtain stronger attack ability. In many cases, this setting would be attributed to the gray box threat model. However, in order to distinguish from the gray box threat model in the following, we call this setting black box threat in a compromise way.

In the grey box threat model, in addition to the parameters of the target model, the attacker can also obtain the parameters of the sparse feature mask (SFM) involved in the defense. Based on SFM parameters, attackers hope to bypass SFM's filtering of redundant features while changing the classification results of the target model. Further speaking, in the white box threat model, in addition to the above knowledge, the attacker can also use adaptive attack algorithm to deal with the further filtering through the image pre-processing algorithm against adversarial perturbation.

(3) Attack settings

On both of CIFAR10, SVHN and Tiny-ImageNet, the same attack methods are used to test the defense proposed method. Under the black box threat model, the used attacks include PGD attacks with L_∞ norm of 8/255 and 16/255, MIM attacks with L_∞ norm of 8/255 and 16/255, and the Diverse attack [36] with L_∞ norm of 8/255 and 16/255 proposed. In grey box and white box environment, the used attacks include FGSM attack with L_∞ norm of 8/255 and 16/255, PGD attack L_∞ norm of 8/255 and 16/255, and CW attack. The number of iterations of PGD attack and MIM attack is set to 30, and the iteration step is set to 1/255. The binary search times of CW is set 5, the number of iterations each round is set to 20, and the initialization weight parameter is set to 0.1.

6.2. **Hyper-parameter selection experiment.** In order to make the proposed method achieve the best defense effect, it's needed to experiment and evaluate on different combination of hyper-parameters. However, due to the large number of hyper-parameters involved in this method, the grid search strategy is selected to select the hyper-parameters. Specifically, the hyper-parameters within the feasible range are sampled and traversed at fixed intervals, and the optimal combination of hyper-parameters are selected through the experimental results. It can be found that, when the appropriate combination of $w_2$and w_3 are selected, and the value of $w_1$is located in the feasible region, the classification result of the model on clean samples is negatively correlated with the value of w_1, while the classification accuracy on adversarial examples is positively correlated with the value of w_1.

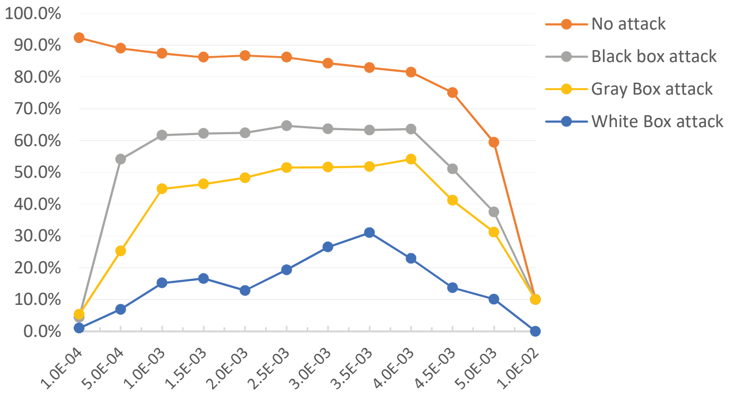

FIGURE 3. Clean and adversarial accuracy with different w_1

In addition, through empirical observation, it can be found that on CIFAR10 and SVHN datasets, fixing w_2 to 1.0 and w_3 to 1.0 can achieve the best performance; While on Tiny-ImageNet, fixing w_2 to 0.01 and w_3 to 100 can achieve the best performance.

Figure 3 shows the accuracy of the defense model under no attack, black box attack, gray box attack when using the appropriate w_2 and w_3 on CIFAR10, where the adopted attack method is PGD with L_∞ norm of 8/255.

Since when the appropriate hyper-parameters are selected, the classification accuracy on the clean samples are negatively correlated with the value of w_1, while the accuracy on the adversarial examples are positively correlated with the value of w_1, in the subsequent experiments on each dataset, we will report the experimental results with the highest sum of the clean accuracy and the adversarial accuracy in black box environment. Specifically, the w_1 values used on CIFAR10 and SVHN are 2.5e-3, and the w_1 value used on Tiny-ImageNet is 0.1.

6.3. **Defense effect comparison experiment.** In order to measure the effectiveness of the proposed method, we will first test the defense effect of the proposed method under the black box threat model, of which the average accuracy of the defended clean samples and the adversarial examples on the target model are used as evaluating indicators. At the same time, in order to test the ability of the proposed method to resist the threat of more powerful adversarial examples, we will test the proposed method under gray box and white box models respectively in the next two sub-sections.

The proposed defense method is compared with the following most advanced defense methods for adversarial examples based on image denoising or pre-processing: feature squeezing [19] (which is also the baseline method in this paper, referred as FS); image quilting [37] (referred as Quilt); total variation minimization [37] (referred as TVM); Pixeldefend [22] and high-low sensitivity adaptive defense method [26] (referred as Sensitivity). The method without defense is represented by Vanilla. Similar to the method, these methods are based on image denoising or pre-processing to remove the impact of adversarial perturbation and try to ensure the correct classification of images.

Experiment result on CIFAR10. The performance under black box attack on CIFAR10 is shown in Table 1. It can be found that the defense effect of the proposed method is better than all existing methods under small perturbation (8/255). While In the case of large-scale perturbation (16/255), the defense effect of the proposed method is not better than Sensitivity [26], but obviously better than other existing methods. In addition, the accuracy of the proposed method on clean samples is significantly higher than Sensitivity [26], which is close to the accuracy in the case of no defense.

Table 1. Defense performance comparison in black box environment on CIFAR10

Method	Clean	PGD-8	PGD-16	MIM-8	MIM-16	Diverse-8	Diverse-16
Vanilla	87.1%	34.8%	21.3%	24.4%	12.2%	20.8%	10.4%
FS	85.0%	44.5%	28.4%	32.5%	15.4%	28.3%	13.5%
TVM	82.7%	50.5%	23.6%	36.6%	16.2%	32.4%	14.1%
Quilt	77.5%	64.8%	52.9%	53.6%	34.1%	51.3%	30.5%
PixelDefend	78.9%	46.9%	29.1%	34.9%	16.0%	31.1%	13.8%
Sensitivity	77.6%	62.2%	**56.3%**	53.8%	**45.4%**	51.0%	**44.5%**
Ours	**86.2%**	**64.6%**	40.3%	**59.7%**	33.3%	**55.1%**	31.3%

The performance under gray box attack on CIFAR10 is shown in Table 2. It can be found that the defense effect of the proposed method under all attacks exceeds that of the existing methods. Specifically, under small perturbation (8/255), the performance of the proposed method significantly exceeds existing methods. However, the performance advantage of the proposed method is relatively small under large perturbation (16/255) compared with Sensitivity [26]. Otherwise, the proposed method can almost completely prevent the CW attack which generates the minimum amplitude of adversarial perturbation.

Table 2. Defense performance comparison in gray box environment on CIFAR10

Method	Clean	FGSM-8	FGSM-16	PGD-8	PGD-16	CW
Vanilla	**87.1%**	14.2%	11.8%	7.2%	6.8%	8.2%
FS	85.0%	18.4%	13.4%	8.2%	6.9%	8.6%
TVM	82.7%	25.5%	13.3%	7.9%	6.8%	8.6%
Quilt	77.5%	32.4%	17.8%	25.8%	10.8%	35.5%
PixelDefend	78.9%	18.2%	13.0%	8.2%	6.8%	10.2%
Sensitivity	77.6%	34.3%	27.0%	43.8%	33.4%	51.1%
Ours	86.2%	**56.8%**	**34.2%**	**51.5%**	**34.7%**	**84.9%**

Experiment result on SVHN. The performance under black box and gray box attacks on SVHN is shown in Table 3 and Table 4 respectively. It can be found that the defense effect of the proposed method both in black box environment and gray box environment is better than the existing methods. Specifically, in the black box environment, the performance of the proposed method significantly exceeds that of the existing methods. While in the gray box environment, the performance advantage of the proposed method under FGSM attack is relatively large, but the performance advantage under PGD attack is relatively small. Similar to CIFAR10, the proposed method can almost completely prevent CW attack that generate the minimum amplitude of adversarial perturbation.

Table 3. Defense performance comparison in black box environment on SVHN

Method	Clean	PGD-8	PGD-16	MIM-8	MIM-16	Diverse-8	Diverse-16
Vanilla	**93.8%**	30.2%	13.6%	30.0%	10.0%	28.8%	9.3%
FS	92.7%	33.2%	15.5%	32.6%	11.4%	31.6%	10.7%
TVM	92.6%	34.3%	15.3%	33.6%	11.1%	32.7%	10.4%
Quilt	93.3%	36.0%	16.7%	35.1%	11.6%	34.2%	11.1%
PixelDefend	92.5%	30.9%	13.6%	30.6%	9.9%	29.4%	9.3%
Sensitivity	92.5%	57.9%	29.8%	60.6%	25.2%	60.4%	24.2%
Ours	91.3%	**70.0%**	**52.6%**	**67.1%**	**50.3%**	**65.8%**	**48.1%**

Experiment result on Tiny-ImageNet. The performance under black box attack on Tiny-ImageNet is shown in Table 5. It can be found that the defense effect of the proposed method is better than all existing methods under small amplitude of perturbation (8/255). And in the case of large-scale perturbation (16/255), the defense effect of the proposed method is worse than Sensitivity [26] with small disadvantage. However, the accuracy of the proposed method on clean samples is slightly lower than Sensitivity [26].

The performance under gray box attack on Tiny-ImageNet is shown in Table 6. It can be found that the defense effect of the proposed method under all attacks exceeds that of the existing methods. Specifically, the defense effect of the proposed method is obviously better than the existing methods against FGSM attack. And under PGD attack, the performance advantage of the proposed method is relatively small. In addition, the proposed method can prevent CW attack with high accuracy, which is similar to the performance on other two datasets.

In general, the performance of the proposed method in black box environment is competitive with that of the current SOTA method. For the adversarial examples with small

Table 4. Defense performance comparison in gray box environment on SVHN

Method	Clean	FGSM-8	FGSM-16	PGD-8	PGD-16	CW
Vanilla	**93.8%**	18.0%	5.9%	8.8%	3.7%	3.8%
FS	92.7%	20.2%	6.6%	11.1%	4.6%	5.1%
TVM	92.6%	92.6%	27.1%	7.3%	10.5%	3.8%
Quilt	93.3%	21.7%	6.8%	12.1%	4.4%	4.7%
PixelDefend	92.5%	18.2%	5.5%	9.4%	3.6%	3.9%
Sensitivity	92.5%	35.8%	16.5%	40.4%	17.5%	35.5%
Ours	91.3%	**64.1%**	**50.7%**	**42.1%**	**17.9%**	**87.9%**

Table 5. Defense performance comparison in black box environment on Tiny-ImageNet

Method	Clean	PGD-8	PGD-16	MIM-8	MIM-16	Diverse-8	Diverse-16
Vanilla	**69.9%**	19.3%	10.3%	17.4%	8.3%	14.8%	7.5%
FS	66.6%	20.2%	10.3%	17.1%	8.4%	14.8%	7.4%
TVM	67.1%	20.2%	10.9%	17.5%	8.7%	14.8%	7.7%
Quilt	64.6%	35.1%	24.6%	31.6%	14.2%	28.1%	12.4%
Sensitivity	60.1%	37.8%	**32.8%**	36.1%	**23.0%**	33.8%	**19.9%**
Ours	58.4%	**40.3%**	30.1%	**38.7%**	22.5%	**37.2%**	18.2%

perturbation amplitude, the proposed method has better defense effect. While for the adversarial examples with large amplitude of perturbation, the effect of the proposed method cannot exceed Sensitivity[26] in most cases. In consideration of the fact that adversarial examples need to be invisible to the naked eye in most cases, small perturbation amplitude is usually adopted. Therefore, the defense effect on adversarial perturbation with relatively small amplitude may bring more practical advantages. In addition, in the gray box environment, the proposed method achieves the best defense effect under all conditions. This shows that the proposed method can provide better defensive performance when resisting attackers with stronger ability.

Table 6. Defense performance comparison in gray box environment on Tiny-ImageNet

Method	Clean	FGSM-8	FGSM-16	PGD-8	PGD-16	CW
Vanilla	**69.9%**	7.4%	4.5%	7.1%	5.9%	11.4%
FS	66.6%	8.0%	6.1%	8.7%	6.4%	0.0%
TVM	67.1%	12.3%	6.4%	8.0%	5.5%	12.3%
Quilt	64.6%	15.4%	8.2%	12.8%	8.8%	20.4%
Sensitivity	60.1%	17.1%	9.9%	23.0%	17.2%	35.0%
Ours	58.4%	**24.9%**	**15.6%**	**25.3%**	**19.9%**	**53.0%**

6.4. **Adaptive attack experiment.** In order to evaluate the defense effect of the proposed method in the worst case and analyze whether the effect depends on the obfuscated gradients effect [38], the proposed method is tested with adaptive attack in the white box environment. The settings in the white box environment have been described in Section 6.1. Its result is shown in Table 7.

It can be found that the proposed method can effectively resist adaptive attacks on SVHN. But on the other two datasets, the ability to resist adaptive attacks is worse.

Table 7. Defense performance comparison in white box environment against adaptive attacks on all datasets

	Clean	FGSM-8	FGSM-16	PGD-8	PGD-16	CW
CIFAR10	86.2%	51.5%	37.4%	11.0%	4.7%	1.6%
SVHN	91.3%	66.2%	43.4%	29.6%	17.1%	31.7%
Tiny-ImageNet	58.4%	10.4%	7.6%	0.4%	0.1%	0.2%

Especially on Tiny-ImageNet, the proposed method can hardly resist the adversarial examples generated by iterative attacks, including PGD and CW. The effect difference on different datasets may be related to the identification objects and the number of categories concerned by these datasets. SVHN focuses on house numbers and its data manifold is relatively simple, while CIFAR10 and Tiny-ImageNet focus on complex objects identification task. In addition, the number of categories of Tiny-ImageNet is much larger than the other two datasets, where the attacker may be able to move the clean samples to other similar categories, such as different kinds of fish or dogs. The factors affecting the defense effect on adaptive attacks need to be further studied in the future work.

6.5. **Time cost experiment.** The time cost of defense methods is divided into the time cost of the training phase and the time cost of the testing phase. As the proposed SFM-Defense only needs to train the SFM network in the training phase, so the increased training time cost will be roughly the same as that of a standard model, depending on the architectural complexity of the SFM network. At the same time, the testing time cost only includes the running cost of the SFM network and the time cost of simple image denoising algorithms which can be almost ignored. Therefore, the time cost of the testing phase is about twice that of a standard model. To intuitively illustrate this point, we carry out experiment on the time cost of proposed method and several typical compared methods on the CIFAR10 dataset, and the results are shown in Table 8. It should be noted that in order to facilitate visual comparison, the training time cost includes the time cost of target model training and defense model training, and the testing time cost includes the running time cost of defense model and target model.

Table 8. Comparison of time cost for different filtering methods on CIFAR10

Time Cost	None	FS(Baseline)	PixelDefend	Sensitivity	Ours
Training Cost (50 epochs)	78.9min	78.9min	208.3min	78.9min	231.7min
Defending Cost (10000samples)	0.3min	0.3min	0.8min	9.6min	0.8min

It can be found that, compared with simple image denoising methods(FS), the propoed method and PixelDefend [22] based on manifold mapping need to train an additional network, so the training time cost is increased, but still within an acceptable range. Compared with the methods based on complex image reconstruction, such as Sensitivity [26], the time cost of the proposed method is very small, and will almost does not affect the real-time performance of the defended model. Considering that only the heuristic image reconstruction method has the similar defense performance with the proposed method, the time efficiency advantage of proposed method in the testing phase is obvious.

6.6. **Visualization.** In order to intuitively show the impact of the proposed method and explain the reason how the defense method works, part of input samples and defended samples on different datasets are shown in Figure 4. It can be found that on CIFAR10 and Tiny-ImageNet datasets, the proposed method tends to remove the pixels on the background, thus only retaining the features that have important contributions to classification. While On the SVHN, the proposed method tends to enhance semantic features, thus indirectly weakening the influence of non-robust features. As mentioned earlier, this difference may be related to the different types of classified object.

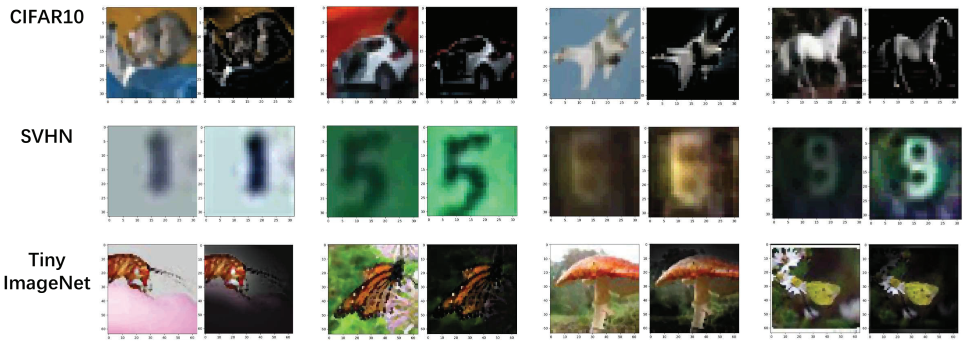

Figure 4. Visualization of defended samples on different datasets

In addition, in order to reflect the impact of different components of the loss function on the proposed method, we show the defended samples generated by the proposed method under the conditions of no semantic loss (No_Seg), no reconstruction loss (No_Recon) and complete loss function (Complete) respectively in Figure 5. It can be found that when semantic loss or reconstruction loss is not used, part of the defense samples generated by SFM-Defense retain very little semantic information, while the other part generates defense samples with almost all 0s, which is consistent with the qualitative analysis in Section 4. In addition, when semantic loss or reconstruction loss is not used, the clean accuracy and confrontation accuracy of the defended model are basically the same as those of random guess, and no additional demonstration will be made here.

7. **Conclusion.** Image transformation is an important way to inhibit the impact of adversarial perturbation to defend adversarial examples. In view of the imbalance of defense effect and time cost existing in the previous works, a defense method based on non-robust feature inhibition and image pre-processing is proposed in this paper, which is called SFM-Defense. In this method, a sparse feature masker is designed and trained to attenuate redundant information in the input image, and combined with simple image processing methods. SFM-Defense has the following two advantages:

1) SFM-Defense has little change to the interface process of the target model, and the added time cost in interface process is only the running cost of the mask generation network. Compared with DefenseGan [23], Dip [25] and other optimization based methods, the proposed method hardly affects the real-time performance of the model, so it is more suitable for application in the actual system.

2) SFM-Defense hardly needs to make any changes to the training process of the target model. The increased training time cost is only the training cost of the SFM network, which is much lower than the time cost of the standard adversarial training [13], which makes this method more practical in the large-scale datasets and networks compared with adversarial training.

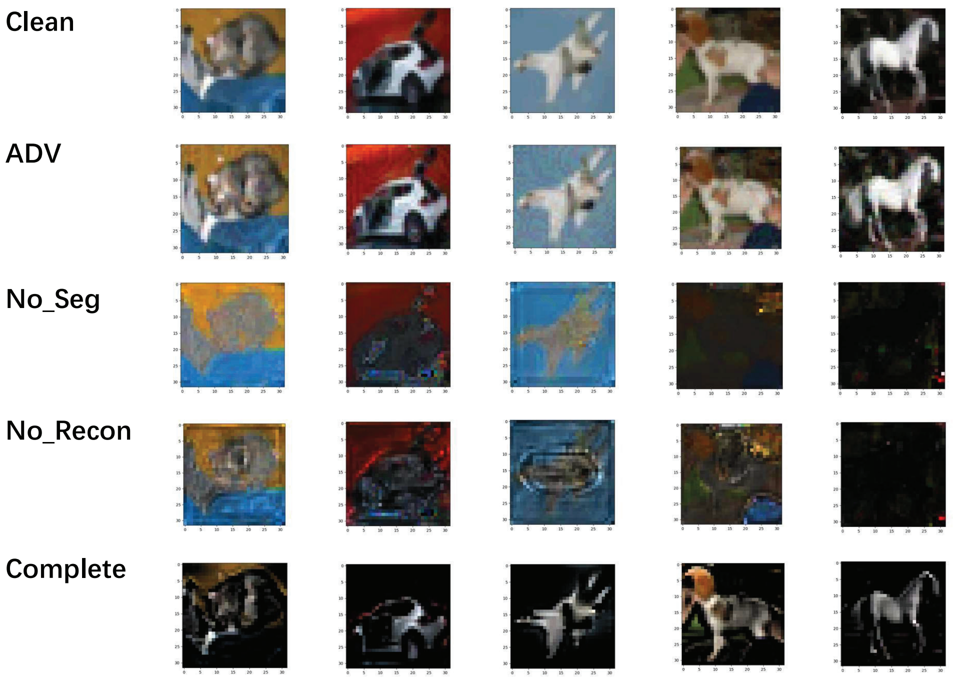

FIGURE 5. Visualization of defended samples with different loss function components

Experiments were carried out on CIFAR10, SVHN and Tiny-ImageNet. The experimental results in the black box environment show that the proposed method has a better defense effect against small amplitude of adversarial perturbation, while the performance against large perturbation is relatively lower. At the same time, in the gray box environment, the proposed method has an obvious advantage compared with the existing methods. However, the proposed method don't have an impressive effect on defending the adaptive attack in the white box environment, which is an important issue that needs to be improved in the future work.

Acknowledgment. This work is supported by the National Defense Basic Scientific Research Program of China (No.JCKY2018603B006). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] Y. Yuan, L. Chen, H. Wu, and L. Li, "Advanced agricultural disease image recognition technologies: A review," *Information Processing in Agriculture*, vol. 9, no. 1, pp. 48–59, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214317321000032

[2] N. H. Tandel, H. B. Prajapati, and V. K. Dabhi, "Voice recognition and voice comparison using machine learning techniques: A survey," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 2020, pp. 459–465.

[3] Y. Zhang and X. Chen, "Explainable recommendation: A survey and new perspectives," *Foundations and Trends® in Information Retrieval*, vol. 14, no. 1, pp. 1–101, 2020. [Online]. Available: http://dx.doi.org/10.1561/1500000066

[4] F. Stahlberg, "Neural machine translation: A review," *Journal of Artificial Intelligence Research*, vol. 69, pp. 343–418, 2020.

[5] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Madry, B. Li, and T. Goldstein, "Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[6] E. De Cristofaro, "A critical overview of privacy in machine learning," *IEEE Security and Privacy*, vol. 19, no. 4, pp. 19–27, 2021.

[7] G. R. Machado, E. Silva, and R. R. Goldschmidt, "Adversarial machine learning in image classification: A survey toward the defender's perspective," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–38, 2021.

[8] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, "Face recognition systems: A survey," *Sensors*, vol. 20, no. 2, p. 342, 2020.

[9] C. F. Camerer and X. Li, "Neural autopilot and context-sensitivity of habits," *Current Opinion in Behavioral Sciences*, vol. 41, pp. 185–190, 2021.

[10] T. Sharma and D. Rattan, "Malicious application detection in android—a systematic literature review," *Computer Science Review*, vol. 40, p. 100373, 2021.

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *STAT*, vol. 1050, p. 20, 2015.

[12] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International Conference on Machine Learning.* PMLR, 2019, pp. 7472–7482.

[13] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *stat*, vol. 1050, p. 9, 2017.

[14] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *International Conference on Machine Learning.* PMLR, 2019, pp. 1310–1320.

[15] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, "Parseval networks: Improving robustness to adversarial examples," in *International Conference on Machine Learning.* PMLR, 2017, pp. 854–863.

[16] H. Kannan, A. Kurakin, and I. Goodfellow, "Adversarial logit pairing," *arXiv preprint arXiv:1803.06373*, 2018.

[17] M. Takeru, M. Shin-Ichi, I. Shin, and K. Masanori, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1979 – 1993, 2018.

[18] T. Na, J. H. Ko, and S. Mukhopadhyay, "Cascade adversarial machine learning regularized with a unified embedding," in *International Conference on Learning Representations*, 2018.

[19] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *Proceedings 2018 Network and Distributed System Security Symposium*, 2018.

[20] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 501–509.

[21] Z. Liu, Q. Liu, T. Liu, N. Xu, X. Lin, Y. Wang, and W. Wen, "Feature distillation: Dnn-oriented jpeg compression against adversarial examples," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2019, pp. 860–868.

[22] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," in *International Conference on Learning Representations*, 2018.

[23] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-gan: Protecting classifiers against adversarial attacks using generative models," in *International Conference on Learning Representations*, 2018.

[24] X. Jia, X. Wei, X. Cao, and H. Foroosh, "Comdefend: An efficient image compression model to defend adversarial examples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6084–6092.

[25] T. Dai, Y. Feng, B. Chen, J. Lu, and S.-T. Xia, "Deep image prior based defense against adversarial examples," *Pattern Recognition*, vol. 122, p. 108249, 2022.

[26] Z. Zhao, H. Wang, H. Sun, J. Yuan, Z. Huang, and Z. He, "Removing adversarial noise via low-rank completion of high-sensitivity points," *IEEE Transactions on Image Processing*, vol. 30, pp. 6485–6497, 2021.

[27] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," in *International Conference on Learning Representations*, 2018.

[28] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *Advances in neural information processing systems*, vol. 32, 2019.

[29] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction.* Springer, 2009, vol. 2.

[30] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Handbook of Systemic Autoimmune Diseases*, vol. 1, no. 4, 2009.

[31] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[32] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[34] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[35] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 2015, pp. 234–241.

[36] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2730–2739.

[37] C. Guo, M. Rana, M. Cissé, and L. van der Maaten, "Countering adversarial images using input transformations," in *6nd International Conference on Learning Representations, ICLR 2018*, 2018.

[38] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International Conference on Machine Learning*. PMLR, 2018, pp. 274–283.