International Journal of Multimedia Information Retrieval (2021) 10:227–237
https://doi.org/10.1007/s13735-021-00219-0

REGULAR PAPER

Towards a high robust neural network via feature matching

Jian Li[1] · Yanming Guo[1] · Songyang Lao[1] · Yulun Wu[1] · Liang Bai[1] · Yingmei Wei[1]

Received: 24 July 2021 / Revised: 13 September 2021 / Accepted: 28 September 2021 / Published online: 13 October 2021

Abstract
Image classification systems have been found vulnerable to adversarial attack, which is imperceptible to human but can easily fool deep neural networks. Recent researches indicate that regularizing the network by introducing randomness could greatly improve the model's robustness against adversarial attack, but the randomness module would normally involve complex calculations and numerous additional parameters and seriously affect the model performance on clean data. In this paper, we propose a feature matching module to regularize the network. Specifically, our model learns a feature vector for each category and imposes additional restrictions on image features. Then, the similarity between image features and category features is used as the basis for classification. Our method does not introduce any additional network parameters than undefended model and can be easily integrated into any neural network. Experiments on the CIFAR10 and SVHN datasets highlight that our proposed module can effectively improve both clean data and perturbed data accuracy in comparison with the state-of-the-art defense methods and outperform the L2P method by 6.3%, 24% on clean and perturbed data, respectively, using ResNet-V2(18) architecture.

Keywords Feature matching · Deep neural network · Adversarial attack and defense · Robustness

1 Introduction

Deep neural networks (DNNs) have demonstrated superior performance in diverse research areas, such as image classification [1] and machine translation [2]. However, recent researches [3–5] indicate that deep models are vulnerable to adversarial examples, thereby seriously limiting their application in safely-critical scenarios. For the image classification task, an adversarial example is an image with carefully designed perturbation, which is not visually perceptible, but can drastically affect the model performance. Based on the prior knowledge of the model, the adversarial attack algorithms can be generally divided into white-box attack and black-box attack. For the white-box attack, the adversary can get access to the entire information of the model (including the structure and the parameters); therefore, the gradient can be precisely calculated according to the predefined loss function and be propagated to the original input to generate the adversarial examples. While for the black-box attack, the model information is only partially accessible to the adversary. It needs to query the model frequently, in order to mimic the real output and conduct an effective attack. Compared to the white-box attack, the black-box attack has less information about the attacked model, so it normally has a lower attack success rate.

To deal with the adversarial attack, various defense algorithms have been proposed, including data compression [6,7], gradient masking [8,9] and adversarial training [4,10], in which the adversarial training is considered as the simplest and effective way to improve the model robustness.

Recently, several works [11–15] have proven that regularizing the network by introducing randomness is another effective way to deal with adversarial examples. Although these methods add noise at different ways, their ultimate goals allow the output of the network layers to change within an acceptable range in the training phase, which makes the network adapt to the impact of adversarial examples. Regrettably, the introduction of a large amount of randomness has led to a phenomenon of over-regularization (i.e., under-fitting), and these methods generally involve complex training process [13,14] and need to manually set multiple noise hyper-parameters [11,12,15], which greatly affect the final performance and have to be tuned carefully.

To overcome the above shortcomings, in this paper, we propose a feature matching (FM) module to predict the image

✉ Yanming Guo
guoyanming@nudt.edu.cn

[1] National University of Defense Technology, Changsha, China

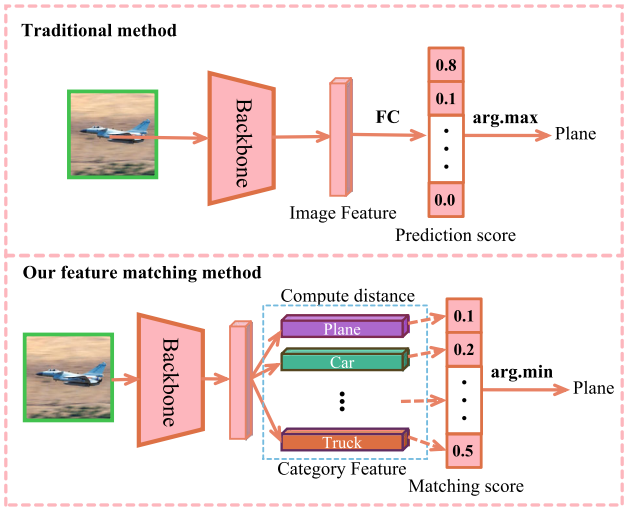

Fig. 1 Comparison of our method and against traditional methods

category. The main function of FM module is to regularize the model. Inspired by the randomness methods, we also allow the image features to change within an acceptable range during training. In this way, we hope to increase the difficulty of adversarial examples features far away from the real category. As shown in Fig. 1, the traditional method uses a fully connected layer to project the image features into prediction scores, where the index corresponding to the maximum score is the prediction result. In comparison, our proposed method learns a feature for each image category during the training phase. In the test phase, backbone network extracts the input image feature and then calculates the distance between the image feature and the category features. These distances build the matching scores, where the size of the score represents the difference between the features. Hence, the minimum score index is the predicted result.

In a nutshell, our contributions can be summarized as follows:

1. We use a feature matching module to replace the fully connected layer, which can significantly improve the model's robustness to adversarial attack without introducing additional parameters.
2. Compared with the methods of regularizing the network by injecting noise, our method loses less accuracy of clean data and eliminates the complex training process.
3. Extensive experiments on the CIFAR10 and SVHN datasets indicate that our method achieves state-of-the-art robustness to adversarial attack in white-box and black-box environments.

The remaining of this article is organized as follows: Section 2 mainly reviews some relative attack and defense methods. Section 3 introduces the proposed feature matching framework. Section 4 demonstrates the experimental results under different setups, as well as our analysis. Qualitative evaluation of our method is presented in Sect. 5. We prove that our method is not relying on gradient obfuscation in Sect. 6. We further discuss our method in Sect. 7, and Sect. 8 concludes this work.

2 Relate work

This section reviews some relative and well-performing attack and defense methods, which will be investigated in this work.

2.1 Adversarial attack

In 2014, Goodfellow et al. [4] explained the existence of adversarial examples; afterwards, many attack algorithms against image classification networks have been proposed. Several typical white-box (i.e., FGSM [4], PGD [10], and C&W [5]) and black-box (i.e., One-Pixel [16] and transferability attack [17]) adversarial attack methods are briefly introduced as follows. As we would evaluate our method under specific parameter settings in the experiment section, we also elaborate the formulas of the attack algorithms.

FGSM Attack: Fast gradient sign method (FGSM) [4] is a simple first-order attack algorithm that uses the symbolized gradient of the input image to generate adversarial examples. When we define a pretrained DNN model f and loss function l_f, FGSM generates an adversarial example using Eq. 1.

$$x^{'} = x + \varepsilon \cdot \text{sign}(\nabla_x l_f(f(x), \text{label})) \tag{1}$$

where the ε refers to the range of attack strength from 0–255, and $sign(\cdot)$ is the gradient sign. In this way, the perturbed image $x^{'}$ will increase the loss function value, causing the network to output a wrong classification result.

PGD Attack: Madry et al. [10] proposes the PGD algorithm, which is a variant of FGSM. With the initialization $x^{(k=1)} = x$, PGD is divided into k steps to generate an adversarial example. The process can be described as:

$$x^{k+1} = \Pi_{x\pm\varepsilon}\left\{x^k + \alpha \cdot \text{sign}\left(\nabla_x l_f(f(x^k), \text{label})\right)\right\} \tag{2}$$

where α is a small step size, adversarial example is within a specified l_p range of the original input x. Madry et al. [10] also experimentally proves that even if $x^{(k=1)}$ is randomly initialized within the l_∞-ball around x, the generated adversarial examples would converge to similar local maximum

loss values. Based on this fact, they claim that PGD is a universal adversary among all the first-order adversaries.

C&W Attack: C&W [5] is another strong white-box attack method and can attack undefended model with almost 100% attack success rate. The attack process of C&W is optimized based on the following objective function formula.

$$\min \left\{ \| \delta \|_p + c \cdot L(x^{'}) \right\} \qquad x^{'} = x + \delta \tag{3}$$

$$L(x^{'}) = \max \left\{ -k, Z(x^{'})_t - \max Z(x^{'})_{i \neq t} \right\} \tag{4}$$

where δ is the perturbation added to the image, t is the true label of x and p is the L_p-norm of the perturbation, and p can be 0, 2, or ∞. $Z(\cdot)$ is the prediction value of each category output by DNN. k is a confidence level greater than 0, and c is used to balance the two parts of the target. Given that C&W achieves optimum performance for p=2, this work evaluates the robustness of a model employing L_2-norm-based C&W attack.

As our model gets a classification result with smallest matching score, we substitute the original $Z(\cdot)$ in Eq. 4 with n-$Z(\cdot)$ and modify the second part loss function as Eq. 5, where n is a large value.

$$\max \left\{ -k, n - Z(x^{'})_t - \max(n - Z(x^{'})_{i \neq t}) \right\} \tag{5}$$

Black-box Attack: Different from the white-box attack, the model information is only partially accessible to the adversary in black-box environment. However, Liu et al. [18] have verified that different models trained with the same training dataset have similar decision boundaries. Therefore, if the training dataset can be obtained, one typically black-box attack method is that training a agent model and attacking the agent model to generate adversarial examples. Several works [17,19] have shown that transferability attack uses the adversarial examples generated by attack agent model that can also attack the black-box model with a high attack success rate. On the other hand, if the training dataset cannot be obtained, One-Pixel [16] attack is an efficient method to conduct black-box attack. One-Pixel attack does not relying on dataset, as it uses a differential evolution algorithm to generate adversarial examples from a randomly initialized perturbed population.

2.2 Adversarial defense

Adversarial training is a common practice to improve the robustness of the model against adversarial attack. Although several works [4,20] have highlighted that adversarial training can be used to regularize the network and improve the model's robustness, Moosavi-Dezfooli [21] proves that no matter how many adversarial examples are added to the training process, there will be new adversarial examples that can fool the model after adversarial training is completed.

Recent researches [11–14] have proved that employing both noise injection and adversarial training to regularize the model can further improve the model's robustness against adversarial attack. The random self-ensemble (RSE) [11] method adds an additive noise layer before the convolution layer and carries out the forward propagation many times in the test phase, which simulates an ensemble of multiple models. Although RSE significantly improves the model robustness, the variance of the noise is an adjusted hyper-parameter that requires manually selected. In contrast, the parametric noise injection (PNI) [13] learns weights through the network to automatic control noise injection. Learn2Perturb (L2P) [14] is a recent extension of PNI. The noise injection is learned in an end-to-end manner, and the model is trained using a method called alternating back propagation, that is, alternatively training the noise injection module and network layers. Instead of adding additive noise to network layer, the Adv-BNN [12] method assumes all the weights in the network are stochastic and uses the commonly used techniques in Bayesian neural network and adversarial training to train a highly robust model.

Although these noise injection methods improve perturbed data accuracy, the accuracy on clean data is significantly reduced. Moreover, these methods introduce numerous parameters that need to be trained compared to the undefended model. In contrast, our proposed method does not inject noise and additional trainable parameters into network and loses less clean data accuracy than other competitive methods.

3 Proposed method

This section introduces our FM module along with the training and optimization process.

3.1 Feature matching module

We report our model architecture in Fig. 2; it mainly comprises a backbone network f_W and feature matching module. The FM module mainly encodes labels and queries the positive feature and negative feature based on image label. Then, we use the Euclidean distance between image feature and label embedding vector to compute loss and predict classification result. A deeper layer can extract higher-level visual information from an image, which will provide our model higher accuracy of clean data. Therefore, in this work, we select the last convolution layer of a backbone network and global average pooling to extract the feature F_x of input image x. We get label embedding features through word-

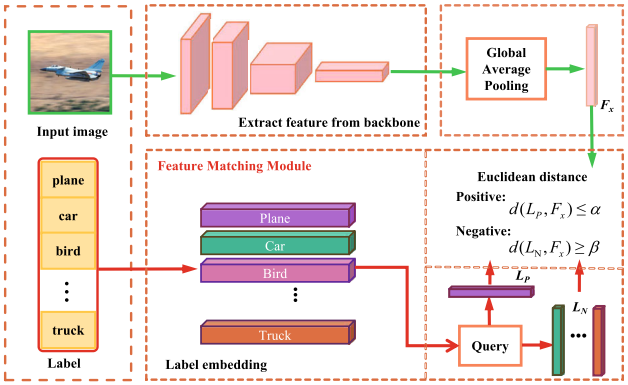

Fig. 2 The architecture of the our FM model. The backbone network can be any known classification model

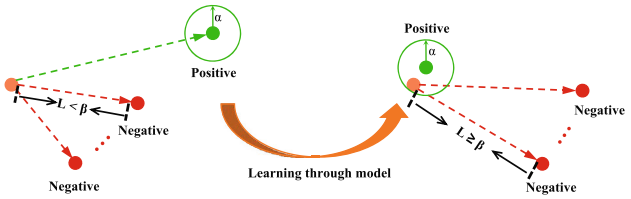

Fig. 3 The fullple loss reduces the distance between an image feature and the positive feature and maximizes the distance between the image feature and all the negative features

embedding technology and set the feature dimension is same with F_x.

We divide the label embedding features into positive feature L_P and negative feature L_N. The two class features are obtained through a query process. For example, for a given image x and its label y, the L_P is the y-th row one-dimensional vector in the embedding matrix, and the L_N represents the remaining vectors. We use these positive and negative features to compute loss value and predict the label of a test image. As presented in Eq. 6, our model defines a new loss function named fullple loss. The fullple loss intends to make the category feature similar to all the same category images but far away from all the different categories images. Accordingly, it can be divided into two parts. The first part refers to the positive sample loss. If the Euclidean distance between the image feature F_x and the positive feature L_P is less than threshold α, the loss of this part is zero. The second half refers to the negative sample loss. If the distance between the image feature F_x and all the negative features L_N exceeds β, the loss of this part is zero. Figure 3 highlights that the fullple loss function encourages images to be close to the true label embedding vector and far away from all the fake labels embedding vector. Moreover, the L_P and L_N are not related to the backbone network parameters. When we require all the $f_W(\widetilde{x})$($\widetilde{x}$ represents all images in a category) to be close to a certain L_P, it is actually a constraint on the parameter W as the images in $\widetilde{x}$ are different. So, our loss function also plays a regularizing role in the model.

$$\mathcal{L}(x, f_W) = \max\{d(F_x, L_P) - \alpha, 0\} + \max\{\beta - d(F_x, L_N), 0\} \tag{6}$$

To avoid repeated parameter adjustments, in the experiment phase, we fix the two parameters α and β as Eq. 7, where len(F_x) represents the length of the image features extracted by the backbone neural network, e.g., 64 for ResNet-V1 and 512 for ResNet-V2.

$$\alpha = \sqrt{\text{len}(F_x) \cdot 2 \cdot 10^{-9}}, \beta = \sqrt{\text{len}(F_x) \cdot 7 \cdot 10^{-4}} \tag{7}$$

As shown in Eq. 8, we calculate the Euclidean distance between the label embedding vector L_i {i=0,$\cdots$,n-1} of each category and the image feature vector F_x to form the network's matching scores, which represent the differences between the image and the n categories. We choose the smallest matching score index as the classification result, as shown in Eq. 9.

$$f_W(x) = [d(F_x, L_0), d(F_x, L_1), \ldots, d(F_x, L_{n-1})] \tag{8}$$

$$\text{pred} = \arg\min(f_W(x)) \tag{9}$$

Moreover, the parameter number of the label embedding module is equal to the element number in the embedding matrix, which is same with the weight number of fully connected layer. As the label embedding module does not use bias, our model has fewer trainable parameters than the undefended model.

3.2 Training strategy and optimization

Algorithm 1 The train and test process of feature clustering network

Input: Training set D=$\{(x_i, y_i), i = 1, ..., n\}$
Number of training epoch, k
Learning rate, lr
Adversarial examples generator, g
Output: Learned parameters W;

for $t = 1; t < k; t++$ **do**
  update W based on the loss fountion $\mathcal{L}(\cdot)$;
  if $t <= 20$ **then**
    $W^t = W^{t-1} - lr * \nabla_W \mathcal{L}(X; W^{t-1}, Y)$;
  else
    $W^t = W^{t-1} - 0.5 * lr * \nabla_W \mathcal{L}(X; W^{t-1}, Y)$
      $-0.5 * lr * \nabla_W \mathcal{L}(g(X); W^{t-1}, Y)$;
  end if
end for
compute test result: $y^{'} = arg.min(f_W(x_0))$

Algorithm 1 shows the training strategy and optimization of our proposed FM method. As we can see, our model does not require a special training process. The label embedding

module is similar to a fully connected layer, whose input is a category and the output is a vector. During the training process, the label embedding module is set as a trainable network layer and trained together with the backbone network. To further enhance the model's robustness to adversarial attack, we also employ an adversarial training scheme, as adding adversarial examples affords the model to learn more general category features. The ensemble loss $\mathcal{L}_{ens}$ is described in Eq. 10, where $g(\cdot)$ is an adversarial examples generation algorithm. In consistent with L2P [14], we use the PGD attack [10] for the $g(\cdot)$ and add adversarial examples into the training dataset after the model is trained for 20 epochs.

$$\mathcal{L}_{\text{ens}} = w_c \cdot \mathcal{L}(x, f_W) + w_a \cdot \mathcal{L}(g(x), f_W) \tag{10}$$

where w_c is the weight of clean data loss and w_a is the weight of perturbed data loss. In this work, we follow the competitive methods [13,14] and set $w_a = w_c = 0.5$.

4 Experiments

To evaluate the defensive performance of our method, we adopt FM module to train various models and observe their robustness to different attack methods. In addition, we compare our model with typical state-of-the-art methods, including vanilla PGD adversarial training [10], random self-ensemble (RSE) [11], adversarial Bayesian neural network (Adv-BNN) [12], parametric noise injection (PNI) [13], Learn2Perturb (L2P) [14]. We extract the experimental results of the competitive methods from the above papers.

4.1 Dataset and attack

Dataset: The experiments employ two commonly used datasets to evaluate the model's defense capability, CIFAR10 [22] and SVHN [23]. The CIFAR10 dataset involves 10 common types of nature images and consists of 50,000 training and 10,000 test data. Each image has RGB channel setup with a size of 3232 pixels. The SVHN dataset is derived from Google Street View house numbers and is a more challenging version of MNIST. It comprises 99,289 RGB images with a size of 3232, where 73,257 images are used as training data and 26,032 images as test data. For both datasets, we use the same data augmentation strategy (i.e., random crop, random flip) of L2P [14] during training. We do not use normalization in data augmentation, but set it as an non-trainable network layer on the top of the backbone network.

Attack: To evaluate the defensive capability, we compare our method with other defensive methods, in resistance with different white-box and black-box attack settings. The white-box attack includes FGSM [4], PGD [10], C&W [5], and the black-box attack includes One-Pixel [16] and transferability attack [17]. In the evaluate phase, the adversarial examples are generated by adding perturbation to the test images using the attack algorithm. The perturbed data accuracy refers to the proportion of adversarial examples correctly classified by the model. And the clean data accuracy is the classification accuracy on clean testing images.

4.2 Experimental setup

In this work, we utilize VGG [24] and ResNet [25] as the backbone notwork. The classical ResNet (i.e., ResNet-V1) and the new ResNet (i.e., ResNet-V2) are used for evaluation. Compared with ResNet-V1, ResNet-V2 has more stages and kernel numbers and thus has more trainable parameters.

For the attack algorithms, we follow their original configurations [11–14]: For the PGD attack, the attack strength ε in Eq. 2 is set to 8/255, and the iterative step k is set to 7 with the step size α=0.01. The FGSM attack adopts the same attack strength ε setup as PGD. The C&W attack employs the Adam optimizer with a learning rate of $5e^{-4}$. The weight c in Eq. 4 is initially set as 10^{-3} and ranges from 0 to 10^{10}. We use a nine-step binary search to determine it and optimize 1000 times for each search iteration. The confidence parameter k of the C&W attack is set to 5 different values (0,0.1,1,2,5) and set the number n in Eq. 5 to 10. For the One-Pixel attack, we set the number of perturbed pixel is 1. For the transferability attack, we use the PGD algorithm to produce adversarial examples and then use these adversarial examples to attack the target model.

4.3 Evaluation of the FM module

To evaluate the effectiveness of our proposed module, we first compare the accuracy of the model with/without the FM module on clean data and perturbed data. We conduct the experiments on two different datasets (CIFAR10, SVHN). As shown in Table 1, the FM module does not bring in additional parameters. On the contrary, it contains slightly fewer parameters since it eliminates the bias item of the fully connected layer. As expected, the attack algorithms can bring a great drop in accuracy, especially in the undefended model. Take the PGD attack on the ResNet-V2(18) as an example, it has an accuracy of more than 90% on clean data, but the accuracy under PGD attack dramatically drops to less than 0.5%. In contrast, the ResNet-V2(18) with FM module keeps the accuracy of perturbed data more than 80% on both datasets. Similar to the PGD attack, our model also outperforms the undefended model under the FGSM attack. Overall, our FM module makes the backbone network highly robust to adversarial attack.

Note that the FM module suffers from a certain decrease on clean data accuracy, this phenomenon also can be observed in

Table 1 Comparing our FM defense method with the undefended model

Model	#Parameter	CIFAR10			SVHN		
		Clean	FGSM	PGD	Clean	FGSM	PGD
ResNet-V2(18)	11,173,962	95.47	41.82	0.25±0.01	96.51	24.49	0.31±0.01
ResNet-V2(18) with FM	11,173,952	91.66	81.63	80.03±0.15	96.22	81.42	80.09±0.17
VGG19	20,040,522	93.77	26.38	0.05±0.00	96.25	24.52	0.23±0.01
VGG19 with FM	20,040,512	88.48	71.06	68.94±0.16	94.15	66.61	66.00±0.10

Due to the randomness of PGD, the five PGD attacks involved calculated the mean±std% values. Parameter represents the number of all parameters that require training. Clean refers to the model classification accuracy on clean test images

all the competitive methods (as indicated in Table 2). Andrew et al. [26] proves that the features of dataset involve robust features and non-robust features. A undefended model will use all the features in the training phase, so it can achieve a good standard accuracy but a bad robust accuracy. To improve robust accuracy, a defensive model tends to reject these non-robust features, which have only sight correlation to label. Therefore, the model's accuracy will inevitably decline when defended model applied on clean data. Nevertheless, we assume the decrease of FM is acceptable because only a small drop in clean data accuracy is exchanged for a large increase in perturbed data accuracy. For example, on the SVHN dataset, the ResNet-V2(18) with FM only drops by 0.29% (from 96.51 to 96.22%) in clean data accuracy than the undefended model, but the perturbed data accuracy under PGD attack gets significant improvement (from 0.31 to 80.09%).

4.4 White-box attack

4.4.1 Resistance for l_∞-norm based on white-box attack

To further illustrate the effectiveness of the FM method, we challenge it against current state-of-the-art methods, including adversarial training [10], PNI [13], Adv-BNN [12], and L2P [14]. Following the competitive methods, in this section, the experiments are performed on the CIFAR10 dataset using ResNet-V1 (20,32,44,56) and ResNet-V2(18).

Table 2 presents the comparison results under different network setups regarding depth and structure. First, we use ResNet-V1(20,32,44,56) to compare the interplay between the model's depth and robustness. Then, we examine the effectiveness of increasing the number of convolution kernel of ResNet-V1(20) and compare different network widths on the model's robustness. The networks involved are ResNet-V1(20) [1.5], ResNet-V1(20)[2], and ResNet-V1(20)[4], indicating that the input and output channels are expanded to 1.5/2/4. To independently analyze the feature matching module and adversarial training, we also report the test results that do not use adversarial training.

Table 2 shows that the perturbed data accuracy of the Adv-BNN method and L2P method does not increase with the backbone network depth. For example, although the backbone network depth increases from 32 to 56, the perturbed data accuracy keep at 54.62% under PGD attack. The results on Net20(1.5) and Net20 also indicate that the robustness of the Adv-BNN method does not increase with the network width. In contrast, our FM module can improve the model's robust accuracy as the backbone network depth and width increases.

Moreover, compared with the accuracy of the backbone network on clean data shown in the second column, we can find that Adv-BNN and L2P method have more decline on clean data accuracy than our method. For example, when we use Net32 as the backbone network, the accuracy of Adv-BNN on clean data drops 29.68% (92.63%-62.95%) and L2P method drops 8.44 (92.63%-84.19%). In contrast, our FM method only drops 1.11% (92.63%-91.52%) when we do not use adversarial training and 2.05% (92.63%-90.58%) when the adversarial training is used.

Notably, even without using adversarial training, our FM method can also get better accuracy than Adv-BNN and L2P on most evaluated backbone networks. After using adversarial training, the performance of our FM method exceeds Adv-BNN and L2P on all backbone networks, both on clean data and perturbed data. Especially when the Net18 with FM method, the accuracy under FGSM and PGD attack is higher than L2P by 19.2% (81.63%–62.43%) and 23.97% (80.03%–56.06%), respectively. Overall, the experimental results prove the effectiveness of our FM method in improving model's robustness.

Table 3 presents the comparison results between our FM method and other current state-of-the-art methods on the CIFAR10 dataset. Although the existing methods have greatly improved the model's robustness against the PGD attack compared to undefended model, this robustness is at the expense of the accuracy decrease on clean data. In contrast, our proposed FM method provides a robust model and can achieve an appealing performance both on clean and perturbed data.

Table 2 Influence of the network's depth and width on the performance of the feature matching module and current state-of-the-art methods

Model	#Clean	Adv-BNN [12]			L2P [14]			FM without adv.train			FM with adv.train		
		Clean	FGSM	PGD	Clean	FGSM	PGD	Clean	FGSM	PGD	Clean	FGSM	PGD
Net20	91.73	65.76±5.92	51.85±1.49	44.95±1.21	83.62±0.02	58.41±0.07	51.13±0.08	90.07	52.24	39.80±0.25	89.75	**66.68**	**53.12±0.30**
Net32	92.63	62.95±5.63	50.29±2.70	54.62±0.06	84.19±0.06	59.94±0.11	54.62±0.06	91.52	58.82	50.18±0.40	90.58	**69.27**	**57.91±0.20**
Net44	93.10	76.87±0.24	58.55±0.48	54.62±0.06	85.61±0.01	61.32±0.13	54.62±0.06	91.51	64.87	57.13±0.21	90.52	**71.26**	**60.26±0.37**
Net56	93.39	77.20±0.02	57.88±0.02	54.62±0.06	84.82±0.04	61.53±0.04	54.62±0.06	91.38	61.67	60.09±0.31	90.50	**73.33**	**63.60±0.25**
Net20 (1.5×)	92.74	65.58±0.42	36.11±1.29	28.07±1.11	85.40±0.08	61.10±0.06	53.32±0.02	92.05	52.08	44.06±0.32	91.09	**68.54**	**56.32±0.16**
Net20 (2×)	93.43	79.03±0.04	58.30±0.14	53.46±0.06	85.89±0.10	61.61±0.05	54.29±0.02	92.63	55.01	49.43±0.16	91.43	**69.27**	**58.71±0.17**
Net20 (4×)	94.07	82.31±0.03	59.01±0.04	52.61±0.12	86.09±0.05	61.32±0.02	55.75±0.07	93.58	63.17	51.58±0.22	91.87	**73.72**	**64.73±0.21**
Net18	95.47	82.15±0.06	60.04±0.01	53.62±0.06	85.30±0.09	62.43±0.06	56.06±0.08	93.24	78.02	77.62±0.16	91.66	**81.63**	**80.03±0.15**

The FM without adv.train column is the test result when the feature matching module is used without an adversarial training scheme. Since the network under the FM framework does not have randomness, the accuracy of clean data and FGSM perturbed data is fixed. However, the PGD attack algorithm has random initialization, so the accuracy under the PGD attack is presented as (meanstd)%. #Clean is the classification accuracy of the undefended backbone network on clean test images. Clean refers to the clean test images accuracy in the related defended model. Part of the results are abstracted from [14]. Best results are in bold face

Table 3 Comparison of the proposed FM with the state-of-the-art methods on CIFAR10

Method	Model	Clean	PGD
Vanilla [10]	ResNet-V1(20)[4]	87	46.1±0.1
RSE [11]	ResNext	87.5	40
DP [15]	28-10 Wide ResNet	87	25
PNI [13]	ResNet-V1(20)[4]	87.7±0.1	49.1±0.3
AdvBNN [12]	ResNet-V1(56)	77.20	54.62±0.06
BPFC [27]	ResNet-V2(18)	82.4	50.1
L2P [14]	ResNet-V2(18)	85.3±0.1	56.3±0.1
RoCL [28]	ResNet-V2(18)	91.34	49.66
ASCL [29]	ResNet-V2(50)	78.7	55.8
ACL [30]	34-10 Wide ResNet	85.12	56.7
FM	ResNet-V2(18)	**91.66**	**80.03±0.15**

The reported results are based on the highest accuracy in the literature. For PGD attack, the attack strength ε=8/255. Part of the results are abstracted from [14]. Best results are in bold face

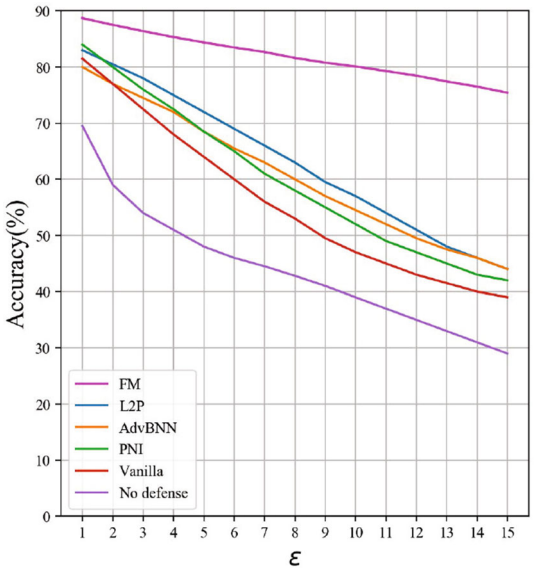

Fig. 4 The comparison of FM and other state-of-the-art methods under different attack strengths of the FGSM

4.4.2 Resistance for different strength attacks

Figure 4 illustrates the robustness of the FM method and the competitive methods under FGSM attack with different strength ε. All results are observed with the ResNet-V2(18) as the backbone network. As can be seen, the robustness of all networks decreases when more and more noise is added to clean data. Nevertheless, our FM method also achieves a superior performance and the advantage over other competitive methods becomes more obvious as the attack strength increases. Specifically, compared with current state-of-the-art L2P, we almost double the accuracy when attack strength ε=15. Next, we conduct the same experiment using PGD

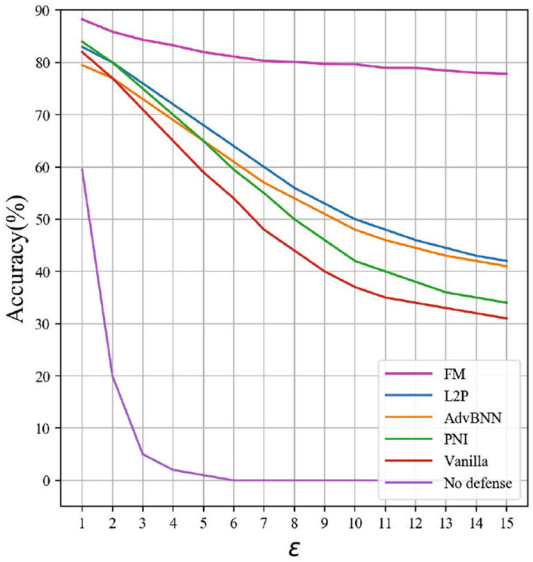

Fig. 5 The comparison of FM and other state-of-the-art methods under different attack strengths of the PGD

Table 4 Comparison with the competitive methods of C&W attack on CIFAR10 dataset when ResNet-V2(18) is used as backbone network

Confident	Adv-BNN [12]	PNI [13]	L2P [14]	FM
k=0	78.9	66.9	83.6	**83.9**
k=0.1	78.1	66.1	84.0	**84.2**
k=1	65.1	34.0	76.4	**81.5**
k=2	49.1	16.0	66.5	**81.5**
k=5	16.0	0.08	34.8	**81.0**

Performance of competitive methods extracted from [14]. Best results are in bold face

attack. As shown in Fig. 5, the accuracy of undefended model quickly dropped to zero under PGD attack. Although the competitive methods obviously improves the model robustness, there is still an apparent gap between them and our method.

4.4.3 Resistance for l_2-norm-based white-box attack

The above experiments and analysis are based on l_∞-norm white-box attack. However, Araujo et al. [31] have shown that robust method to attack on l_∞-norm is not necessarily effective on l_2-norm. Thus, to verify our FM method is still effective against l_2-norm attack, we use C&W algorithms with different confidence levels to attack the RecNet-V2(18) with FM module. We set k in Eq. 6 to five different values, which represents different attack strengths. Table 4 presents the defensive effect of our method. As seen, the FM method can maintain high robustness as the k increases and achieves the best results for all k values. Specially when k=5, the robustness is higher than L2P by 46.2% (81.0%–34.8%).

Table 5 Comparison with the competitive methods of One-Pixel attack on CIFAR10 dataset when ResNet-V2(18) and ResNet-V1(20) are used as backbone network

Backbone	AdvBNN [12]	PNI [13]	L2P [14]	FM
ResNet-V1(20)	58.40	67.40	70.15	74.90
ResNet-V2(18)	68.60	50.90	64.45	70.80

Performance of competitive methods extracted from [14]

Table 6 FM method against transferability attack on CIFAR10 dataset, model A is undefended ResNet-V2(18), while model B is ResNet-V2(18) trained with our method

Source model	PGD attack	Transferability attack [17]
A	99.74	A $\Longrightarrow$ B 13.76
B	19.73	B $\Longrightarrow$ A 64.72

The data in the table is the attack success rate

4.5 Black-box attack

In this section, we conduct attack experiments in the black-box environment. First, we follow the L2P [14] and perform the One-pixel attack on ResNet-V1(20) and ResNet-V2(18). The results of the FM and the competitive methods are presented in Table 5. As shown, the FM method achieves the highest robustness on both backbone networks. Next, we use the PGD algorithm to generate the adversarial examples and use the transferability attack to verify the effectiveness of FM. We report the attack success rate in Table 6. While the PGD algorithm attack model A reaches a 99.74% attack success rate, there is only a 13.76% attack success rate when these adversarial examples are used for model B. Although the PGD algorithm attacks model B with a 19.73% success rate, 64.72% of the adversarial examples can attack model A successfully. The above results show that our method still has defense capability against attacks that do not use gradients.

5 Qualitative evaluation

In addition to the quantitative evaluation above, in this section, we conduct the qualitative analysis by using T-SNE tool to visualize the feature distribution of our model and undefended model. Figure 6 shows the distribution diagram of the ResNet-V2(18) employed as the backbone network on the CIFAR10 and SVHN test data. All the features are extracted from the last convolution layer of backbone network. We use global average pooling and PCA (principal component analysis) to project the features into two-dimensional space. We can easily find from the CIFAR10 distribution that our FM method makes clean data features closer than undefended model, which proves our hypothesis that our loss function

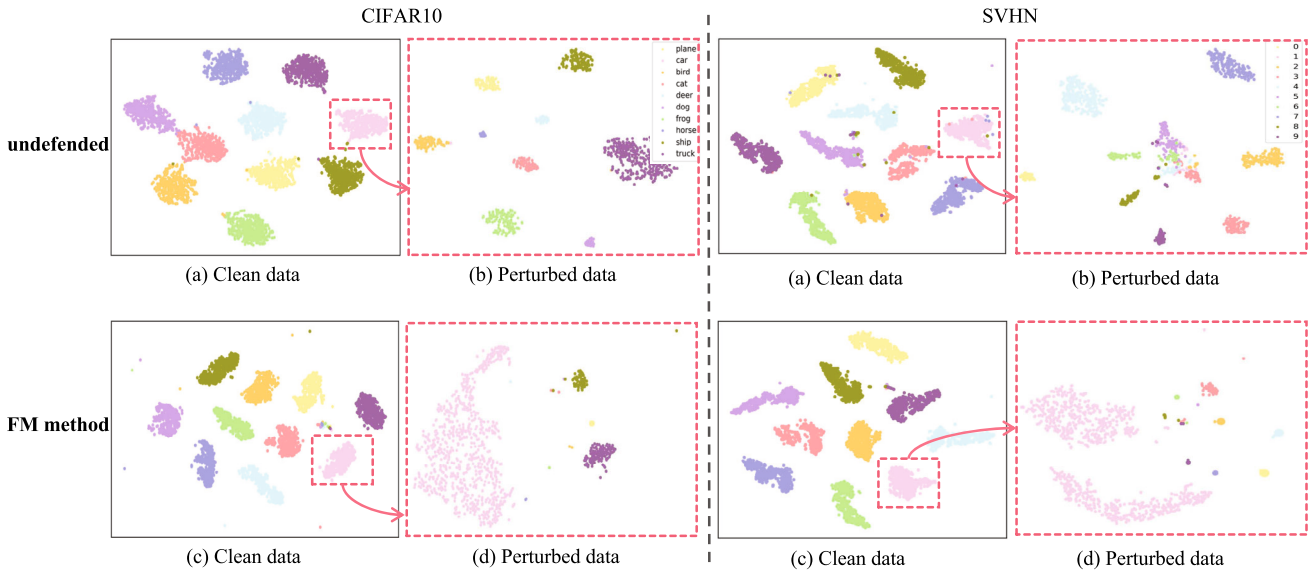

Fig. 6 The visualization of the feature distribution on CIFAR10 and SVHN. The first line is the distribution result of the undefended model ResNet-V2(18). The second line is the result of ResNet-V2(18)with our FM module. We select 500 images from each category in the test data to form the result of clean data and all perturbed data come from the category that selected by the box. We use the PGD attack to generate perturbed data and set the attack strength ε=8

can make all the same category images close to their label embedding feature.

On the other hand, we believe that the features extract by a robust model would be closer to true category than the undefended model under the same attack strength. Unsurprisingly, as can be seen in the undefended model, the PGD(ε=8) attack spreads the features extracted by the boxed category into nine different clusters, so the perturbed data are completely identified as the other nine categories. In contrast, the feature distribution of the perturbed data generated by attacking our model is relatively concentrated, and most of them are still remained as the correct category. In other word, our proposed defense is strong that it is difficult to generate adversarial examples to fool a model with FM module.

Last but not least, it is well known that images in the CIFAR10 can be generally categorized into machines (including plane,car,ship,truck) and animals (including bird,cat,deer,dog,frog,horse). As shown in the distribution result of CIFAR10, the perturbed data of *car* are misclassified as other nine categories in the undefended model, but the perturbed data are mainly misclassified as *ship* and *truck* in our model. So we can conclude that gradient-based attack finds the correct perturb direction to conduct an attack in our model, because our model uses the similarity between image features and category features as the basis for classification. This conclusion is related to the stability of our defense method, and we will further discuss the stability in the next section.

6 Inspection of gradient obfuscation

According to Athalye et al. [32], the defense method based on gradient obfuscation is unreliable. Gradient obfuscation is considered to be unable to correctly obtain the true gradient from a defended model. We try to prove that the robustness provided of our method is not relying on gradient obfuscation from two perspectives: (1) In the above section, we have proved that the gradient-based attack successfully finds the correct perturb direction to complete an attack in our model. In other words, the gradient-based attack successfully finds the correct gradient of our model. (2) Our proposed FM method does not have the five phenomena, which will appear in a defense strategy based on gradient obfuscation according to [32]. In the following, we will give the relevant phenomena and the refutation evidence to prove those phenomena do not exist in our method.

Phenomenon 1: One-step attacks perform better than iterative attacks.

Refutation: From the results in Table 2, we can see that our FM method performs better against FGSM than PGD. In other words, the iterative attack PGD performs better than one-step attack FGSM.

Phenomenon 2: Black-box attacks are better than white-box attacks.

Refutation: From the results of the transferability attack in Table 6, the attack success rate of model B is 13.76% under black-box attack, but the success rate of the white-box

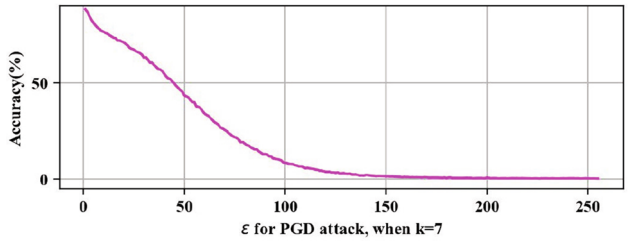

Fig. 7 The robustness of ResNet-V2(18) based on FM when it is attacked by PGD algorithms with different attack strengths. The x-axis represents the attack strength of the PGD algorithm, and the y-axis represents the accuracy of all the adversarial examples in the model

PGD attack is 19.73%. Therefore, white-box attack performs better than black-box attack in our model.

Phenomenon 3: Unbounded attacks do not reach 100% success.

Refutation: We report the our model accuracy under PGD attack in Fig. 7. We set the PGD attack iterative step k=7 and make the attack strength ε change from 0 to 255. As shown, the unbounded PGD attack can reach 100% attack success rate.

Phenomenon 4: Random sampling finds adversarial examples.

Refutation: This phenomenon involves that if the gradient-based attack method cannot find an adversarial example, there will be no adversarial example that can be found even if randomly sampling 10^5 times within the clean image ε-ball. Therefore, we randomly sample 1000 test images from the CIFAR10 test data. These test images are correctly classified by the our model but cannot be successfully perturbed by the PGD algorithm (ε=8/255). Then we conduct 10^5 times random sampling noises within each test image ε-ball, and finally, the classification accuracy of the 10^8 perturbed images in our model is 100%.

Phenomenon 5: Increasing the distortion bound does not increase success.

Refutation: As shown in Figs. 4 and 5, increasing the distortion bound can increase attack success rate.

7 Discussion

Our methods has three main strengths than the competitive methods. (1) Our FM method essentially optimizes the extraction process of image features and does not modify the related network layers. So, the proposed FM method can be integrated into any neural network. As we can see in Tables 1 and 2, our method gets great performance on several different networks and does not need to design carefully for each network architecture. (2) Our method does not increase the parameters of the network, while the competitive methods need additional parameters and complex training processes. (3) We can achieve high accuracy on both clean data and perturbed data, while the other defense methods improve the perturbed data accuracy at the cost of lowering the clean data accuracy.

For the adversarial training process, it normally consists of numerous training epochs, and in each epoch, it will generate the same number of adversarial examples with the training set and double training data (adversarial examples and clean training data) will be learned by the network. Therefore, training a model on the large-scale dataset with the adversarial training is rather time-consuming, and most methods are experimented on the small datasets, typically CIFAR10, to validate their effectiveness. Actually, CIFAR10 has already been employed as the primary dataset for the evaluation of the robustness. To facilitate the comparison with these methods, we also conduct most of our experiments on the CIFAR10 dataset. However, we assume it is more practically useful to ensure the model robustness on large-scale natural images; therefore, it is meaningful to facilitate more efficient training strategies on large-scale datasets.

8 Conclusion and prospect

In this paper, we propose a feature matching module, which enhances the robustness of the model without increasing additional parameters. The module can be easily integrated into any neural network. Extensive white-box and black-box attack experiments verify the effectiveness of the suggested FM method and get state-of-the-art performance both on clean data and perturbed data. We further prove that its high performance does not originate from gradient obfuscation. In the future, we strive to combine our FM module with the randomness methods to further enhance model robustness and apply our method to large-scale dataset.

References

1. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25:1097–1105

2. Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: ICLR
3. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2014) Intriguing properties of neural networks. In: ICLR
4. Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. In: ICLR
5. Carlini N, Wagner D (2017) Towards evaluating the robustness of neural networks. IEEE Symp Sec Privacy (SP). https://doi.org/10.1109/SP.2017.49
6. Dziugaite GK, Ghahramani Z, Roy DM (2016) A study of the effect of jpg compression on adversarial images. arXiv preprint arXiv:1608.00853. [Online]. https://arxiv.org/abs/1608.00853
7. Bhagoji AN, Cullina D, Sitawarin C, Mittal P (2018) Enhancing robustness of machine learning systems via data transformations. Ann Conf Inform Sci Syst (CISS). https://doi.org/10.1109/CISS.2018.8362326
8. Ross A, Doshi-Velez F (2018) Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In: Proceedings of the AAAI conference on artificial intelligence. [Online]. https://ojs.aaai.org/index.php/AAAI/article/view/11504
9. Lyu C, Huang K, Liang H-N (2015) A unified gradient regularization family for adversarial examples. In: Proceedings of the 2015 IEEE international conference on data mining. https://doi.org/10.1109/ICDM.2015.84
10. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A (2018) Towards deep learning models resistant to adversarial attacks. In: ICLR. [Online]. https://openreview.net/forum?id=rJzIBfZAb
11. Liu X, Cheng M, Zhang H, Hsieh C-J (2018) Towards robust neural networks via random self-ensemble. In: ECCV. https://doi.org/10.1007/978-3-030-01234-2-23
12. Liu X, Li Y, Wu C, Hsieh C-J (2019) Adv-BNN: improved adversarial defense through robust Bayesian neural network. In: ICLR . [Online]. https://openreview.net/forum?id=rk4Qso0cKm
13. He Z, Rakin AS, Fan D (2019) Parametric noise injection: trainable randomness to improve deep neural network robustness against adversarial attack. In: CVPR. https://doi.org/10.1109/CVPR.2019.00068
14. Jeddi A, Shafiee MJ, Karg M, Scharfenberger C, Wong A (2020)Learn2perturb: an end-to-end feature perturbation learning to improve adversarial robustness. In: CVPR. https://doi.org/10.1109/CVPR42600.2020.00132
15. Lecuyer M, Atlidakis V, Geambasu R, Hsu D, Jana S (2019) Certified robustness to adversarial examples with differential privacy. In: IEEE symposium on security and privacy (SP). https://doi.org/10.1109/SP.2019.00044
16. Su J, Vargas DV, Sakurai K (2019) One pixel attack for fooling deep neural networks. IEEE Trans Evol Comput 23(5):828–841. https://doi.org/10.1109/TEVC.2019.2890858
17. Papernot N, McDaniel P, Goodfellow I (2016) Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07277. [Online]. https://arxiv.org/abs/1605.07277
18. Liu Y, Chen X, Liu C, Song D (2016) Delving into transferable adversarial examples and black-box attacks. arXiv preprint arXiv:1611.02770. [Online]. https://arxiv.org/abs/1611.02770
19. Papernot N, McDaniel P, Goodfellow I, Jha S, Celik ZB, Swami A (2017) Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia conference on computer and communications security. https://doi.org/10.1145/3052973.3053009
20. Sankaranarayanan S, Jain A, Chellappa R, Lim SN (2018) Regularizing deep networks using efficient layerwise adversarial training. In: Proceedings of the AAAI conference on artificial intelligence. [Online]. https://ojs.aaai.org/index.php/AAAI/article/view/11688
21. Moosavi-Dezfooli S-M, Fawzi A, Fawzi O, Frossard P (2017) Universal adversarial perturbations. In: CVPR. https://doi.org/10.1109/CVPR.2017.17
22. Krizhevsky A, Hinton G, et al (2009) Learning multiple layers of features from tiny images
23. Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng AY (2011) Reading digits in natural images with unsupervised feature learning
24. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. [Online]. https://arxiv.org/abs/1409.1556
25. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: CVPR. https://doi.org/10.1109/CVPR.2016.90
26. Ilyas A, Santurkar S, Engstrom L, Tran B, Madry A (2019) Adversarial examples are not bugs, they are features. Adv Neural Inf Process Syst 32:125–136
27. Addepalli S, Baburaj A, Sriramanan G, Babu RV (2020) Towards achieving adversarial robustness by enforcing feature consistency across bit planes. In: CVPR. https://doi.org/10.1109/CVPR42600.2020.00110
28. Kim M, Tack J, Hwang SJ (2020) Adversarial self-supervised contrastive learning. In: Neural information processing systems
29. Bui A, Le T, Zhao H, Montague P, Camtepe S, Phung D (2021) Understanding and achieving efficient robustness with adversarial supervised contrastive learning. arXiv preprint arXiv:2101.10027
30. Jiang Z, Chen T, Chen T, Wang Z (2020) Robust pre-training by adversarial contrastive learning. In: Neural information processing systems
31. Araujo A, Meunier L, Pinot R, Negrevergne B (2019) Robust neural networks using randomized adversarial training. arXiv preprint arXiv:1903.10219. [Online]. https://arxiv.org/abs/1903.10219
32. Athalye A, Carlini N, Wagner D (2018)Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. In: International conference on machine learning