Information Sciences 560 (2021) 256–270

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Robust feature learning for adversarial defense via hierarchical feature alignment

Xiaoqin Zhang, Jinxin Wang, Tao Wang, Runhua Jiang, Jiawei Xu, Li Zhao *

College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China

ARTICLE INFO

Article history:
Received 15 July 2020
Received in revised form 18 November 2020
Accepted 15 December 2020
Available online 20 December 2020

Keywords:
Adversarial defense
Domain adaptation
Feature alignment
Optimal transport

ABSTRACT

Deep neural networks have demonstrated excellent performance in most computer vision tasks in recent years. However, they are vulnerable to adversarial perturbations generated by adversarial attacks. These human-imperceptible perturbations often lead to severe distortion in the high-dimensional intermediate feature space, which is one of the major reasons for the vulnerabilities in deep neural networks. Therefore, input images with perturbations can completely change the predictions of the networks in the decision space. To overcome this drawback, we propose to progressively align the intermediate feature representations extracted from the adversarial domain with feature representations extracted from a clean domain through domain adaptation. The difference between two feature distributions can be accurately measured via an optimal transport-based Wasserstein distance. Thus, the deep networks are forced to learn robust and domain-invariant feature representations, so that the gap between the different domains is minimized and that the networks are no longer easily fooled by diverse adversaries. Extensive evaluations are conducted on four classification benchmark datasets in white-box attack scenarios. The evaluation results demonstrate a significant performance improvement over several state-of-the-art defense methods.

1. Introduction

Deep neural networks have demonstrated their performance in most computer vision tasks such as object classification [17,41], object detection [25,15], and image processing [38,40]. However, they can be easily fooled by adversarial examples containing human-imperceptible adversarial perturbations. Generated by adversarial attacks [12,27,29,37,2], such adversarial examples pose a serious threat to safety–critical visual applications including driverless vehicles, medical diagnosis, and surveillance systems. Furthermore, if deep models drastically change their outputs with high confidence given images with slight perturbations as input, these models will fail to capture the task-relevant inherent properties of images and cannot distinctively learn the robust visual concepts. Therefore, it is necessary to design deep neural networks that are robust to adversarial perturbations for safe and reliable computer vision applications.

Numerous adversarial defense mechanisms have been proposed to overcome adversarial attacks. These defense methods can be broadly classified into two categories: reactive defenses [19,8,36,23,14] and proactive defenses [18,20,33,39,7,16]. The former aims at mitigating the effect of varying perturbation levels using image transformation or by modifying the input

* Corresponding author.
E-mail addresses: zhangxiaoqinnan@gmail.com (X. Zhang), jxwang@stu.wzu.edu.cn (J. Wang), taowangzj@gmail.com (T. Wang), ddghjikle1@gmail.com (R. Jiang), jxulincoln@gmail.com (J. Xu), lizhao@wzu.edu.cn (L. Zhao).

https://doi.org/10.1016/j.ins.2020.12.042
0020-0255/

images directly during the inference time. However, these algorithms are easily compromised in the event of advanced attacks because of the uncertainty in their assumptions and explanations for the vulnerability. In comparison, proactive defense algorithms focus on improving the adversarial robustness of models against perturbations by designing different learning models or altering the network architectures, *e.g.*, by employing adversarial or ensemble training, modifying the network layers, and changing the activation functions. For the relatively better adversarial robustness against white-box attacks, proactive defense methods are generally more valued.

This paper introduces an optimal transport-based hierarchical feature alignment method, as a proactive defense mechanism against adversarial attacks, with the objective to progressively increase the similarity between the learned feature representations from the normal image domain and the adversarial image domain. We note that the addition of adversarial perturbations in the clean image domain leads to a considerable domain gap between the distributions of clean and adversarial examples in the high-dimensional intermediate feature space and the output decision space. Moreover, adversarial perturbations can progressively modify the hierarchical features of the deep networks, thereby maximizing the distance between the original and adversarial feature representations.

Inspired by this observation, we propose to progressively minimize the divergence of different distributions, such that the gap between the clean and adversarial domains in the intermediate feature space and decision space is minimum. This ensures that the learned deep representations are domain-invariant and task-relevant on both clean images and adversarial examples. Thus, the classifiers can no longer be easily fooled by existing adversaries. In other words, we build on the intuition that two visually similar images in different domains, despite having significantly diverse deep representations, must be projected into the same decision space. Therefore, we must enforce that their extracted hierarchical feature representations at different scales are similar throughout the network. This is achieved by aligning the feature representations from the adversarial domain to features from the clean domain using a novel distance (Wasserstein or earth mover distance) based on the optimal transport theory [34], where the problem of transforming one probability distribution to another in the most economic way is studied. Minimizing the Wasserstein distance helps the model to identify more robust features instead of non-robust features [32]. Moreover, it provides meaningful measurements even when the supports of the feature distributions do not overlap [5]. Leveraging these properties, we introduce a novel defense mechanism to improve the adversarial robustness in the intermediate feature space, which has not been explored for adversarial defense. Fig. 1 illustrates the details of the proposed approach.

The empirical evaluations, presented in Section 5, demonstrate that the proposed defense approach provides a strong defense against different white-box attacks. The approach outperforms several state-of-the-art defenses by a significant margin. Moreover, the method exhibits a strong generalization ability on examples from the adversarial domain compared with other defense algorithms.

The contributions of this work are summarized as follows:

- We propose a novel hierarchical feature alignment method to learn robust feature representations for adversarial defense from the perspective of domain adaptation.
- While progressively improving the feature similarity along network layers for a robust feature learning, an optimal transport-based Wasserstein distance is employed to measure the difference between the features extracted from clean images and adversarial examples.
- Our methodology shows a robust generalization ability on adversarial examples from different attacks compared with several state-of-the-art defense mechanisms.

The rest of this paper is organised as follows: Section 2 presents the literature review on adversarial attacks and defenses. Section 3 introduces the proposed defense methodology. Subsequently, the training details and experimental analyses are given in Section 5. Finally, we conclude this work in Section 6.

2. Related work

In this section, we first provide a brief overview of diverse advanced attack methods and then describe several classical defense mechanisms.

Generating adversarial examples to cheat deep neural networks and developing defense algorithms against such examples have gained significant attention in recent years. Szegedy et al. [31] first find that certain imperceptible adversarial perturbations generated by maximizing the prediction error of a model can cause the network to misclassify an input image. The fast gradient sign method (FGSM) [12] and its iterative variant [18] are then proposed. DeepFool [21] projects the input examples across their complicated decision boundaries in an iterative manner until these examples are misclassified. The projected gradient descent (PGD) [20] is one of the strongest attack methods; it finds adversarial examples within a l_∞ norm-ball. By formulating the generation process of adversarial examples as an optimization problem, Carlini and Wagner [1] propose an attack approach for white-box attacks. Based on the gradient information at each step, Shi et al. [28] propose an efficient iterative attack method called Ada-FGSM, which adaptively allocates the step size of noises. Other superior attack algorithms include the momentum iterative method [9] and the adaptive iteration fast gradient method [35].

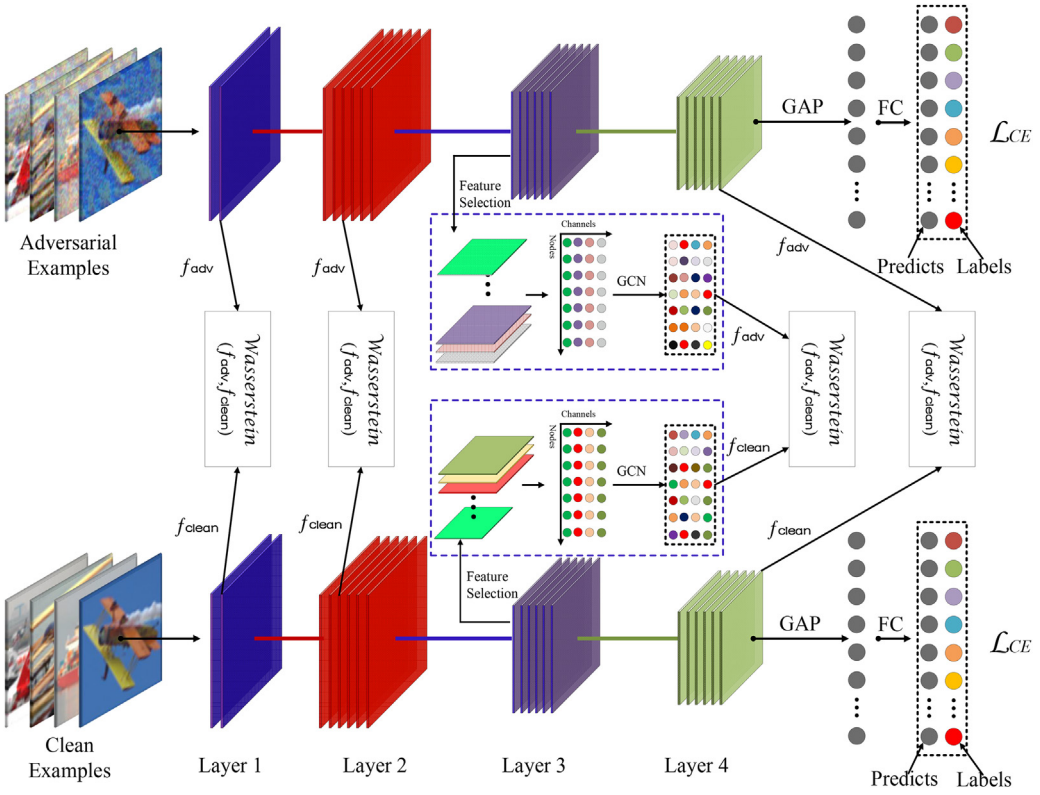


Fig. 1. Illustration of the proposed hierarchical feature alignment method for adversarial defense. After extracting deep features at different levels of the network from clean and adversarial examples, we use the Wasserstein distance to measure the difference between features from adversarial examples f_{adv} and those from clean examples f_{clean}. The four Wasserstein distances in this figure are utilized as loss functions when optimizing the network parameters, along with two cross-entropy losses $\mathcal{L}_{CE}$ to ensure classification accuracy on both clean images and adversarial examples. Before computing the Wasserstein distances, linear combinations used as feature selection and graph convolution network (GCN) are employed here. The extracted or selected features with different colors indicate their different meanings. GAP and FC denote the global average pooling and fully connected operation, respectively. *Zoom in for better visibility.*

The adversarial defense mechanisms in the literature can be broadly classified into two categories. The methods in the first category counter adversarial attacks by employing multiple pre-processing approaches and transformation operations on the given input images. JPEG image compression as a defense mechanism has been proposed in [10,8]. These methods utilize a discrete cosine transform to suppress adversarial noises in the input image domain. However, JPEG image compression is far from being an effective defense method [13]. Leveraging the expressive capability of generative adversarial networks [11], Samangouei et al. [26] propose a new framework termed Defense-GAN to defend deep neural networks. Taking deep super-resolution networks as projecting functions, Mustafa et al. [23] map adversarial examples into the manifold distribution of normal images. Guo et al. [14] propose a transferability prediction difference algorithm to improve the adversarial robustness of deep networks with small degradation in the evaluation accuracy.

Another category of defense methods improves the adversarial robustness of a network through modifying training procedures to handle adversarial attacks. Adversarial training is an effective scheme in this regard, whereby networks with clean images and their adversarial counterparts can by jointly trained. Goodfellow et al. [12] train deep networks on both clean images and adversarial examples generated by the FGSM. The min–max optimization method [20] is one of the strongest defense methods. It augments training data with disturbed images from the first-order attack PGD. Ensemble adversarial training [33] is a novel technique that augments training data with adversarial examples transferred from multiple deep models. As an ensemble method, Dabouei et al. [7] utilize the first-order interactions to improve the robustness of the ensemble classifiers. Through perturbing the feature space and increasing the uncertainty at each layer with novel perturbation-injection modules, a feature perturbation method is proposed by Jeddi et al. [16] to improve the adversarial robustness. To enhance the model generalization ability on adversarial perturbations, a novel adversarial training with a domain adaptation algorithm is introduced by Song et al. [30]. Their method focuses on domain adaptation for adversarial defense without any constraints on the intermediate feature representations, which is fundamentally different from our method. Our experimental results, presented in Section 5, demonstrate that the proposed defense method significantly outperforms some of the existing state-of-the-art techniques.

3. Optimal transport-based hierarchical feature alignment

At the core of our approach lies the proposed hierarchical feature alignment and the optimal transport distance, which ensure feature similarity between clean and adversarial domains. In the following, we first introduce the notations used in this work and then provide a brief overview of the optimal transport-based Wasserstein distance and its iterative approximation algorithm [6]. We should note that the optimal transport-based distance is necessary for the proposed hierarchical feature alignment defense method, which is described in this section.

3.1. Notations

Let $x_{clean} \in \mathbb{R}^m$ and y_{true} denote input data and ground truth label, respectively. We employ a function $F_\theta(\cdot)$ to denote a deep neural network, θ is the learnable network parameter. The deep network output a multi-dimensional feature representation $f^l_{clean} \in \mathbb{R}^{C\times H\times W}$ given a clean image x_{clean} as input, where C, H, W denote the number of channels, and height and width of the features, respectively, extracted at layer l in the network $F_\theta(\cdot)$. Similarly, $f^l_{adv} \in \mathbb{R}^{C\times H\times W}$ denotes the intermediate feature representations extracted on an adversarial example $x_{adv} \in \mathbb{R}^m$. To train the model, we find the optimal value of θ that minimizes a given objective function. Next, we introduce the optimal transport-based distance and its iterative algorithm utilized for the hierarchical feature alignment.

3.2. Optimal transport distance

Let $X = \{x, \ldots, x_N\}$ be a set of N feature vectors extracted from a clean image x_{clean}, and $Y = \{y_1, \ldots, y_M\}$ be a set of M feature vectors extracted from an adversarial example x_{adv}. In this work, these feature vectors can be computed by a series of differentiable operations, which are detailed in Section 3.3.

Optimal transport problems have recently gained significant interest in several research fields; in particular, the optimal transport theory can be used to compute the difference between probability distributions. The existence and uniqueness of the solution can be confirmed theoretically. Given two sets of feature vectors X and Y, the Kantorovich's distance [24] between two probability distributions introduced by the optimal transport problem can be formulated as:

$$W_c(P_X, P_Y) := \inf_{\Gamma\in\mathcal{P}(X\sim P_X, Y\sim P_Y)} \mathbb{E}_{(X,Y)\sim\Gamma}[c(X,Y)], \tag{1}$$

where $\mathcal{P}(X \sim P_X, Y \sim P_Y)$ is a set of joint distributions of (X, Y), and P_X and P_Y are the marginal distributions, respectively. $c(x, y)$ is a measurable cost function, which defines how far a feature vector in X is from another feature vector in Y. For $p \geqslant 1$, different ℓ_p norm functions are usually selected as the cost functions. In this work, the cost function is defined as $c(x, y) = ||x - y||_2^2$ to compute the distance between two feature vectors. Thus, we can obtain the 2-Wasserstein distance:

$$W_2(P_X, P_Y) := \inf_{\Gamma\in\mathcal{P}(X\sim P_X, Y\sim P_Y)} \mathbb{E}_{(X,Y)\sim\Gamma}(||X, Y||_2^2). \tag{2}$$

In practice, Eq. (2) can be simplified as follows in discrete scenarios:

$$W_2(P_X, P_Y) := \min_{P\in\mathcal{P}(X\sim P_X, Y\sim P_Y)} \langle P, C\rangle := \sum_i\sum_j P_{ij}C_{ij}, \tag{3}$$

where $\langle\cdot,\cdot\rangle$ is the Hadamard product between matrix P and matrix C. However, the computation cost can quickly become prohibitive when the algorithm is extended to large datasets. From a maximum-entropy perspective, the classical optimal transport problem can be smoothed with an entropic regularization term. The distance can then be efficiently computed via simple alternate minimization through the Sinkhorn's matrix scaling algorithm, the computation speed of which is several orders of magnitude higher. The entropic regularization term of a coupling matrix P is defined as:

$$H(P) := -\sum_i\sum_j P_{ij}\log(P_{ij} - 1). \tag{4}$$

Thus, the optimal transport problem Eq. (3) with an additional regularization term Eq. (4) can be summarized as:

$$\min_{P\in\mathcal{P}(X\sim P_X, Y\sim P_Y)} \langle P, C\rangle - \epsilon H(P). \tag{5}$$

With the optimal transport solution $P^\star$ after solving the problem, we can determine the Wasserstein distance using $W^\star(X, Y) = \langle P^\star, C\rangle$. From Eq. (5), we can see that it converges to the original problem Eq. (3) as $\epsilon \to 0$. In this work, we set $\epsilon = 0.1$ as the default value. Moreover, when solving the problem with the Sinkhorn's iterative algorithm, we utilize 100 iterations. We should note that, since the objective is a convex problem, it has a unique optimal solution.

In this work, the Wasserstein distance is employed to measure the difference between intermediate feature representations extracted from clean examples and features extracted from their adversarial examples.

3.3. Hierarchical feature alignment for adversarial defense

In this subsection, we propose a hierarchical feature alignment method to defend against adversarial attacks and ensure that the learned models are robust enough to generalize well for various adversarial examples from the adversarial domain. Considering that deep features extracted from the normal image domain and the adversarial domain differ significantly, our approach is to progressively increase the feature similarity at different depths of the deep models via feature alignment. When deep features extracted from different domains become similar and domain-invariant, the models become insensitive to various adversarial perturbations. Moreover, through minimizing the domain gap in the feature space, we can minimize the gap in the decision space. Hence, the generalization ability on examples from the adversarial domain can be effectively improved. Fig. 1 shows the defense mechanism.

Given a clean image x_{clean} with its label y_{true} from the clean image domain and its adversarial counterpart x_{adv} from the adversarial domain, deep feature maps are extracted by the same deep network F_θ. The intermediate feature maps $\{f^1_{clean}, f^2_{clean}, f^3_{clean}, f^4_{clean}\}$ are extracted from the normal image x_{clean} at different depths of the deep network, whereas the feature representations $\{f^1_{adv}, f^2_{adv}, f^3_{adv}, f^4_{adv}\}$ are extracted from the corresponding adversarial counterpart x_{adv}. Fig. 1 illustrates the process of extracting the feature maps from both the clean and adversarial examples. The feature representations with different scales or colors in the figure indicate their different meanings. Before computing the Wasserstein distance between f^l_{clean} and f^l_{adv}, graph convolution is applied to capture the global relations between different regions in the feature maps, such that the new output feature vectors can aggregate information from multiple regions. The new feature vectors projected from the original feature maps are then used as input to compute the Wasserstein distance.

We first transpose and flatten a feature map $f \in \mathbb{R}^{C\times H\times W}$ as $f^\top \in \mathbb{R}^{L\times C}, L = H \times W$ with respect to the feature dimension C, and then conduct feature selection via a linear combination of the original feature maps. In particular, each new feature can be generated as follows:

$$\tilde{f}_i^\top = w_i * f^\top = \sum_{\forall j} w_{ij} f_j^\top, \tag{6}$$

where $f_j^\top \in \mathbb{R}^{1\times C}$ and $\tilde{f}_i^\top \in \mathbb{R}^{1\times C}$ are flattened feature vectors; $W = [w_i, \cdots, w_{C'}] \in \mathbb{R}^{C'\times L}$ is the weight matrix for the linear combination; the feature dimension is reduced from C to C'. In practice, the weight matric W can be implemented with a convolution layer $\psi(\cdot)$ and $W = \psi(f^\top; W_\psi)$; W_ψ represents the trainable parameters in the convolution layer. Similarly, we can also reduce the dimension of each feature vector $f_j^\top$ from L to L' using a linear combination implemented by a convolution layer $\zeta(.)$, and $f^\top = \zeta(f^\top; W_\zeta)$. Thus, we have $\tilde{f}^\top \in \mathbb{R}^{C'\times L'}$ from Eq. (6). After the aforementioned feature selection and dimension reduction operations, the graph convolution is implemented using two 1D convolutions. We refer readers to a relevant study [3] on the recent developments of graph convolutions for visual tasks. Similar to [3], the graph convolution is given by:

$$GCN(\tilde{f}^\top) = Conv1D(Conv1D(\tilde{f}^\top)^\top)^\top. \tag{7}$$

The two 1D convolution layers sequentially perform channel-wise and node-wise information diffusions, which strengthen the feature representation ability on the relationships between different regions after full end-to-end training. Moreover, aiming at providing a tighter restriction on the relationships in feature representations. At each depth of the network, we utilize one graph convolution for features extracted from both clean and adversarial images. When the graph convolution operation is completed, the output feature vectors are given as input to compute the Wasserstein distance.

Our proposed defense mechanism involves a hierarchical feature alignment. As shown in Fig. 1, we utilize the Wasserstein distance to measure the difference between the features extracted from adversarial examples and those extracted from clean examples. The four distances are taken as loss functions to train the network; all the loss functions employed in this work are formulated as follows:

$$\mathcal{L}_{ALL} = \mathcal{L}_{CE}(F_\theta(x_{clean}), y_{true}) + \mathcal{L}_{CE}(F_\theta(x_{adv}), y_{true}) + \lambda \cdot \sum_l^L W_2(GCN(LC(f^l_{clean})), GCN(LC(f^l_{adv}))) \tag{8}$$

where LC denotes the linear combination operation for feature selection and dimension reduction; $\mathcal{L}_{CE}$ is the cross-entropy loss used to ensure the classification accuracy for examples from both the clean domain and the adversarial domain. L indicates the total number of layers utilized in the networks, $L = 2$ when the LeNet network is utilized, and $L = 4$ when the ResNet-110 architecture [22] is employed. λ is the hyper-parameter used to balance the importance of the different distances. $\lambda = 1.0$ is set as the default value. The training algorithm is summarized in Algorithm 1.

Algorithm 1. Optimal transport-based hierarchical feature alignment for robust feature learning

Input: Classifier $F_\theta(\cdot)$, clean training data $\{x_{clean}\}$, ground truth labels $\{y_{true}\}$, trainable model parameters θ, the adversarial perturbation budget ϵ, the number of attack steps κ, total training epochs T, number of layers L that need to be aligned.
Output: Updated model parameters θ
Model Learning:
for t = 0 to T **do**
  if t $\leqslant T'$ **then**
    Train model with $\mathcal{L}_{CE}$ on clean examples $\{x_{clean}\}$ only,
      θ = arg min$_\theta \mathcal{L}_{CE}(F_\theta(x_{clean}), y_{true})$
  else
    Compute the gradients*w.r.t.* θ and x_{clean},
      $\nabla_\theta \mathcal{L}_{CE}(F_\theta(x_{clean}), y_{true})$ and $\nabla_{x_{clean}} \mathcal{L}_{CE}(F_\theta(x_{clean}), y_{true})$, respectively.
    Generate adversarial examples x_{adv} from x_{clean} using PGD
    Extract features $\{f^1_{clean}, \cdots, f^L_{clean}\}$ and $\{f^1_{adv}, \cdots, f^L_{adv}\}$
    Compute joint loss $\mathcal{L}_{ALL}$ using Eq. (8)
      update model parameters: $\theta = \arg\min_\theta \mathcal{L}_{ALL}$
  end if
end if

4. Adversarial attacks

We evaluate our defense approach on four classification datasets against two typical state-of-the-art adversarial attacks, which are introduced below.

4.1. Fast gradient sign method (FGSM)

The FGSM [12] is one of the first attack methods. Given a loss function $L(x_{clean} + \rho, y; \theta)$, where θ denotes the trainable network parameter, and ρ is the specially designed perturbation, the objective of the attack method is to maximize the loss as:

$$\arg\max_{\rho \in R^m} L(x_{clean} + \rho, y; \theta). \tag{9}$$

The loss $L(\cdot, \cdot)$ utilized here for image classification is the cross-entropy loss. The FGSM is a single-step attack method aiming to find the adversarial perturbation by moving in the direction opposite to the gradient of the loss function *w.r.t.* the input image:

$$x_{adv} = x_{clean} + \epsilon \cdot sign(\nabla(L(x_{clean}, y; \theta))). \tag{10}$$

Here, ϵ is the allowed perturbation budget, which essentially restricts the ℓ_∞ norm of the perturbation. To visualize the effects of adversarial perturbations, we illustrate the adversarial examples generated using the FGSM in Fig. 2.

4.2. Projected gradient descent (PGD)

Relying on the first-order information of the target model, the PGD perturbs a normal input image x_{clean} for T steps. After each perturbation step, the PGD projects the adversarial example back into the ϵ-ball of x_{clean}, if it goes beyond:

Fig. 2. Adversarial examples under the FGSM attack method with different perturbation budgets ϵ. *Best viewed in color.* (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

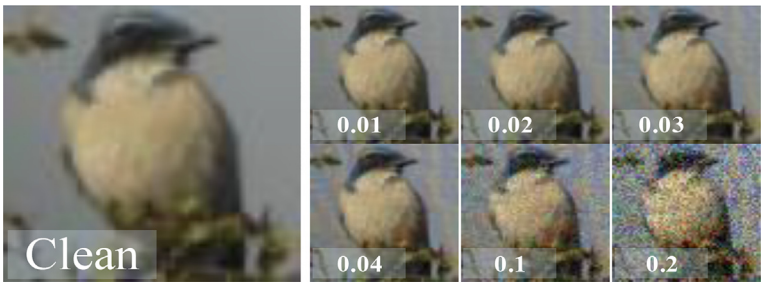


Fig. 3. Adversarial examples under the PGD attack method with different perturbation budgets ϵ. *Best viewed in color.* (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$x_{adv}^{t} = \pi_{\alpha}(x_{adv}^{t-1} + \epsilon \cdot sign(\nabla_{x_{adv}}(L(x_{adv}^{t-1}, y; \theta)))), \tag{11}$$

where ϵ is the perturbation budget, $\pi(\cdot)$ is the projection function, and x_{adv}^{t} is the adversarial example at the t-th step ($x_{adv}^{0} = x_{clean}$). The PGD here utilizes a random initialization for $x_{adv}^{0} = x_{clean} + U^{d}(-\alpha, \alpha)$, where $U^{d}(-\alpha, \alpha)$ is the uniform distribution between $-\alpha$ and α, and x_{adv} has the same dimensions as the input x_{clean}. Fig. 3 shows the adversarial examples generated from the PGD method.

5. Experiments

5.1. Datasets and experimental setup

We extensively evaluate the defense method on four classification datasets: MNIST, fashion-MNIST (F-MNIST), CIFAR-10 and STL-10 [4]. For the MNIST and F-MNIST datasets, the selected CNN model is the famous LeNet network, which has two convolutional layers. For the CIFAR-10 and STL-10 datasets, we utilize the ResNet-110 network. Fig. 4 shows the two network architectures. The deep features for the optimal transport-based hierarchical feature alignment are extracted from different intermediate layers of the two models. We use the first two layers to extract features when training the models on the MNIST and F-MNIST datasets with LeNet. In addition, four intermediate layers of ResNet-110 are employed to extract features from the clean and adversarial examples for their feature alignment. We first train our models for T' epochs with the cross-entropy

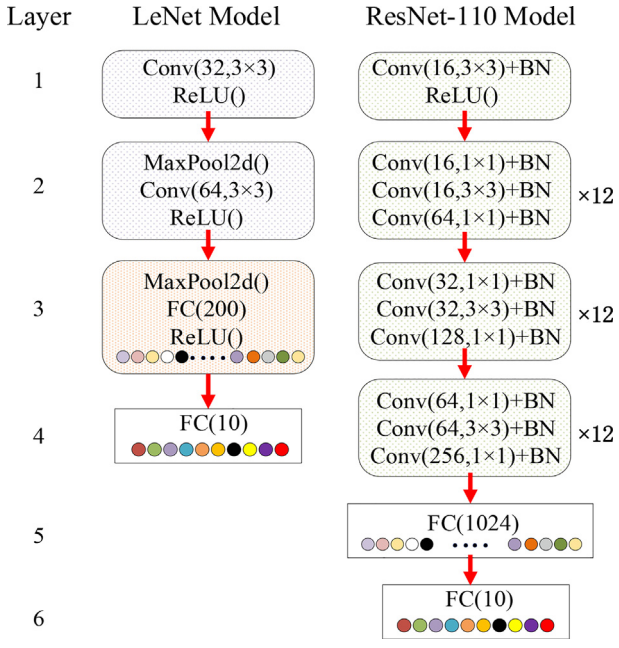


Fig. 4. Two network architectures: LeNet (MNIST, F-MNIST) and ResNet-110 (CIFAR-10, STL-10). Features are extracted from the different layers of LeNet and ResNet-110 for the proposed defense algorithm. *Best viewed in color.* (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

loss on the clean training images ($T' = 20$ for MNIST and F-MNIST datasets; $T' = 40$ for CIFAR-10 and STL-10 datasets). The models are then trained using our proposed method for total T epochs ($T = 60$ for MNIST and F-MNIST datasets, $T = 200$ for CIFAR-10 and STL-10 datasets). A learning rate of 0.01 and a batch size of 128 are used. The stochastic gradient descent (SGD) is utilized to optimize the model parameters; the hyper-parameters momentum and weight decay are set to 0.9 and 0.0001, respectively. After implementing the defense method in the PyTorch framework, we conduct the model training, evaluation and model analyses on the same platform with an Intel-Xeon Silver 4114 CPU, 32 GB of RAM and a RTX 2080 Ti graphics card.

5.2. Hyper-parameters for adversaries

Table 1 lists the details of the hyper-parameters of the PGD adversarial attack employed in our implementations for each benchmark dataset. When training models with adversarial examples generated from the PGD method, $\kappa = 10$ is the default setting; we only use $\kappa = 40$ in model $Ours_{PGD(\kappa=40)}$. For a fair comparison, we use the same ϵ and l_∞-PGD attack method when training the deep models.

5.3. Evaluation comparisons

In this subsection, we evaluate our proposed defense method on four benchmark datasets for image classification to demonstrate the robustness of the model to adversarial attacks and compare its performance with state-of-the-art defenses under two white-box attacks: FGSM and PGD. In the white-box settings, the adversarial attacks can acquire complete knowledge of the architecture of the target model, training method, and model parameters. Tables 2–5 list the test accuracy on the four datasets. We should note that Normal Training means training the models with the cross-entropy loss on the clean images. Goodfellow et al. [12] proposed to augment clean images with adversarial examples generated from the FGSM attack. Madry's adversarial training is one of the most successful defense mechanisms against white-box attacks. To evaluate the generalization ability of the model on adversarial examples, we additionally compute the mean classification accuracy of the classifier given adversarial examples with varying perturbation levels as input.

Evaluation on the MNIST dataset. Table 2 lists the accuracy results on the MNIST dataset. The classifier achieves a satisfactory test accuracy on clean images with normal training; however, its generalization ability on adversarial examples is very poor. When the training is performed as in [12], the classifier presents a better robustness against the FGSM attack but

Table 1
Adversarial settings for training our proposed approach and other defense methods. α, ϵ, κ respectively denote the step size, the perturbation budget, and the number of attack steps for a certain ϵ.

Dataset	Attack	Parameters	Norm
MNIST, F-MNIST	PGD	$\alpha = 0.01,\ \epsilon = 0.3,\ \kappa = 10$	ℓ_∞
CIFAR-10, STL-10	PGD	$\alpha = 0.01,\ \epsilon = 0.03,\ \kappa = 10$ or 40	ℓ_∞

Table 2
Evaluation on the MNIST dataset. Testing accuracy (%) under different perturbation budgets ϵ of FGSM and PGD attacks.

Defense	Clean	FGSM (%)		PGD ($\kappa = 20$) (%)		Mean
		$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.2$	$\epsilon = 0.3$	
Normal Training	99.18	47.19	18.00	12.96	2.10	20.06
Goodfellow et al. [12]	97.45	89.71	93.06	13.35	13.35	52.36
Madry et al. [20]	98.08	76.92	27.98	93.63	93.63	73.04
$Ours_{PGD(\kappa=10)}$	**99.23**	94.9	88.92	94.75	92.51	92.77
$Ours_{PGD(\kappa=40)}$	98.63	**97.34**	**97.25**	**97.07**	**96.54**	**97.07**

Table 3
Evaluation on the F-MNIST dataset. Testing accuracy (%) under different perturbation budgets ϵ of FGSM and PGD attacks.

Defense	Clean	FGSM (%)		PGD ($\kappa = 10$) (%)		PGD ($\kappa = 20$) (%)		Mean
		$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.2$	$\epsilon = 0.3$	
Normal Training	91.06	11.31	8.08	5.48	4.4	5.25	4.13	6.41
Goodfellow et al.[12]	87.97	**84.35**	**89.09**	17.61	17.61	6.42	6.42	39.61
Madry et al.[20]	84.84	68.13	54.12	**78.36**	**77.59**	67.23	62.16	67.93
$Ours_{PGD(\kappa=10)}$	91.29	80.06	78.62	74.39	76.78	58.79	65.04	**72.78**
$Ours_{PGD(\kappa=40)}$	**91.59**	77.29	76.18	71.41	72.16	**68.32**	**69.41**	72.46

Table 4
Evaluation on the CIFAR-10 dataset. Testing accuracy (%) under different perturbation budgets ϵ of FGSM and PGD attacks.

Defense	Clean	FGSM (%)		PGD ($\kappa = 10$) (%)		PGD ($\kappa = 20$) (%)		Mean
		$\epsilon = 0.02$	$\epsilon = 0.03$	$\epsilon = 0.02$	$\epsilon = 0.03$	$\epsilon = 0.02$	$\epsilon = 0.03$	
Normal Training	**87.95**	26.8	23.73	7.49	7.17	7.47	6.92	13.26
Goodfellow et al. [12]	78.00	47.00	36.13	38.42	23.01	37.93	21.71	34.01
Madry et al. [20]	75.60	54.57	45.69	50.27	38.03	50.00	37.17	45.96
$Ours_{PGD(\kappa=10)}$	83.16	67.24	55.87	63.88	48.56	63.74	47.96	57.84
$Ours_{PGD(\kappa=40)}$	80.89	**67.72**	**57.34**	**65.05**	**51.62**	**64.99**	**50.92**	**59.61**

Table 5
Evaluation on the STL-10 dataset. Testing accuracy (%) under different perturbation budgets ϵ of FGSM and PGD attacks.

Defense	Clean	FGSM (%)		PGD ($\kappa = 10$) (%)		PGD ($\kappa = 20$) (%)		Mean
		$\epsilon = 0.02$	$\epsilon = 0.03$	$\epsilon = 0.02$	$\epsilon = 0.03$	$\epsilon = 0.02$	$\epsilon = 0.03$	
Normal Training	57.31	21.26	17.8	14.22	12.61	13.78	12.31	15.33
Goodfellow et al. [12]	**60.81**	32.94	25.79	26.24	16.60	25.93	15.99	23.92
Madry et al. [20]	56.69	36.47	29.46	33.09	24.26	32.95	23.86	30.02
$Ours_{PGD(\kappa=10)}$	59.05	47.8	40.06	44.16	35.17	44.07	**35.06**	41.05
$Ours_{PGD(\kappa=40)}$	59.50	**47.92**	**40.80**	**44.91**	**35.58**	**44.79**	**35.06**	**41.51**

is almost completely ineffective against the PGD attack. Madry's adversarial training achieves higher accuracy on adversarial examples generated from the FGSM and PGD attacks; however, its accuracy on adversarial examples generated using the FGSM is still limited. Our method significantly outperforms the two defenses when training the model using adversarial examples generated using the PGD method with an attack step $k = 40$. Moreover, our defense achieves a mean accuracy of 97.07% on the dataset; the mean accuracy demonstrates that our defense method exhibits the strongest adversarial robustness while avoiding any excessive overfitting on the adversarial examples.

Evaluation on the F-MNIST dataset. Table 3 lists the classification accuracy on adversarial examples under different distortion levels ϵ. Similar to the performance on the MNIST dataset, Goodfellow et al. and Madry et al. achieve excellent results against FGSM and PGD attacks, respectively. However, their mean accuracies are limited because of the poor generalization ability. Our approach achieves a relative gain of approximately 20% about mean accuracy compared to the Madry's adversarial training method on the F-MNIST dataset. The results indicate that the proposed method has the strongest generalization ability against attacks under varying distortion levels, even though it does not always achieve the best test accuracy under every attack setting.

Evaluation on the CIFAR-10 dataset. Compared with the MNIST and F-MNIST datasets, the CIFAR-10 dataset is more challenging. Table 4 lists the test results. Choosing ResNet-110 as the classifier trained on CIFAR-10, normal training gives the best accuracy on clean images and all the defense methods tend to exhibit a poor classification performance on clean data. On this dataset, our proposed defense method achieves the highest accuracy on the adversarial examples. Our defense method achieves mean accuracies of 57.84% and 59.61%, thus significantly outperforming the other methods. The results confirm that the generalization ability of neural networks is still guaranteed. We also provide more details regarding the evaluation results on the CIFAR-10 dataset in Fig. 5.

Evaluation on the STL-10 dataset. The STL-10 dataset is similar to the CIFAR-10 dataset; it contains only 5000 training images but has 8000 testing images in ten classes. Thus, it is more challenging to obtain a good prediction accuracy on the

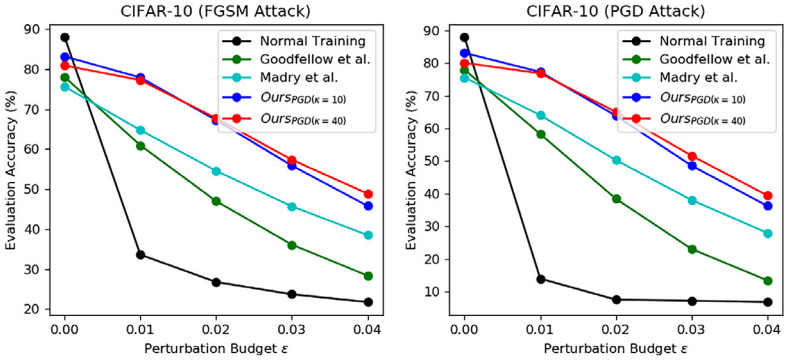


Fig. 5. Evaluation results under FGSM and l_∞-PGD attacks on the CIFAR-10 dataset. Compared with the other defenses, our method exhibits the best performance.

Table 6
Comparison of our approach with Song et al. [30] on the CIFAR-10 dataset.

Defense	Clean	FGSM (%)				PGD ($\kappa = 10$) (%)				Mean
		$\epsilon = 0.01$	$\epsilon = 0.02$	$\epsilon = 0.03$	$\epsilon = 0.04$	$\epsilon = 0.01$	$\epsilon = 0.02$	$\epsilon = 0.03$	$\epsilon = 0.04$	
Song et al.[30]	79.8	68.96	57.91	47.56	39.01	68.75	56.83	44.28	33.70	52.13
$Ours_{PGD(\kappa=10)}$	**82.79**	**78.04**	**67.42**	56.19	46.37	**77.42**	63.94	49.41	36.29	59.44
$Ours_{PGD(\kappa=40)}$	81.64	77.36	67.17	**57.22**	**47.55**	76.78	**64.37**	**50.50**	38.43	**59.92**

test set. Table 5 lists the evaluation results. The proposed defense method still achieves a state-of-the-art defense performance. Compared with the methods proposed by Goodfellow et al. and Madry et al., our method achieves relative gains of approximately 17% and 11%, respectively. Furthermore, the performance drop under white-box attack settings is negligible.

Comparison with a similar method. We compare the performance of our model with a similar method proposed in [30], as listed in Table 6. Our method is based on the hierarchical feature alignment that progressively increases the similarity of the intermediate feature representations extracted from both clean and adversarial examples. This paradigm is the main contributing factor behind our improved performance. In comparison, the similar method concentrates on domain adaption between clean and adversarial examples without any constraint on their extracted deep features. For a fair comparison, we retrain the defense model employed in [30] on clean images and their adversarial examples generated from the PGD attack with perturbation budget $\epsilon = 0.03$, step size $\alpha = 0.01$, and attack step $\kappa = 10$. All the deep models here are trained for a total of $T = 150$ epochs on the CIFAR-10 dataset, just as the settings in [30]. The evaluation results in Table 6 show that our proposed approach outperforms this method with an approximately 7% better classification accuracy on adversarial examples.

In conclusion, the accuracy results on several benchmarks demonstrate that our proposed approach has a better generalization ability than other competing defense mechanisms.

5.4. Effects of λ for adversarial robustness

To investigate the effects of λ on the joint loss function Eq. (5), we evaluate the model performance on the four datasets under FGSM and PGD attacks with different perturbation budgets ϵ. Fig. 6 illustrates the detailed comparison results. We train all the models here with adversarial examples generated from PGD for $T = 60$ epochs on the MNIST and F-MNIST datasets. When evaluating the model accuracy on MNIST, $\kappa = 20$ is set for PGD, which is a stronger attack setting. On CIFAR-10 and STL-10, all the models are trained for $T = 200$ epochs and the adversarial examples are generated with the PGD attack method as well.

From the comparison results on MNIST, F-MNIST, and CIFAR-10, we find that choosing an appropriate λ value is a promising way to improve the adversarial robustness of the models. The performance gain is more pronounced for attacks with a high-level perturbation budget without noticeable performance drop under other smaller perturbation budgets. This further indicates that our proposed hierarchical feature alignment method can efficiently improve the generalization ability of the model on adversarial examples generated using FGSM and PGD with a high distortion level. Fig. 6 shows that $\lambda = 60$ and $\lambda = 40$ are the best parameter settings for MNIST and F-MNIST datasets, respectively. Moreover, we find that it is a good choice to select the hyper-parameter in $[0.1, 1.0]$ on CIFAR-10, otherwise the performance will be slightly degraded. On the STL-10 dataset, $\lambda = 20$ yields the best performance.

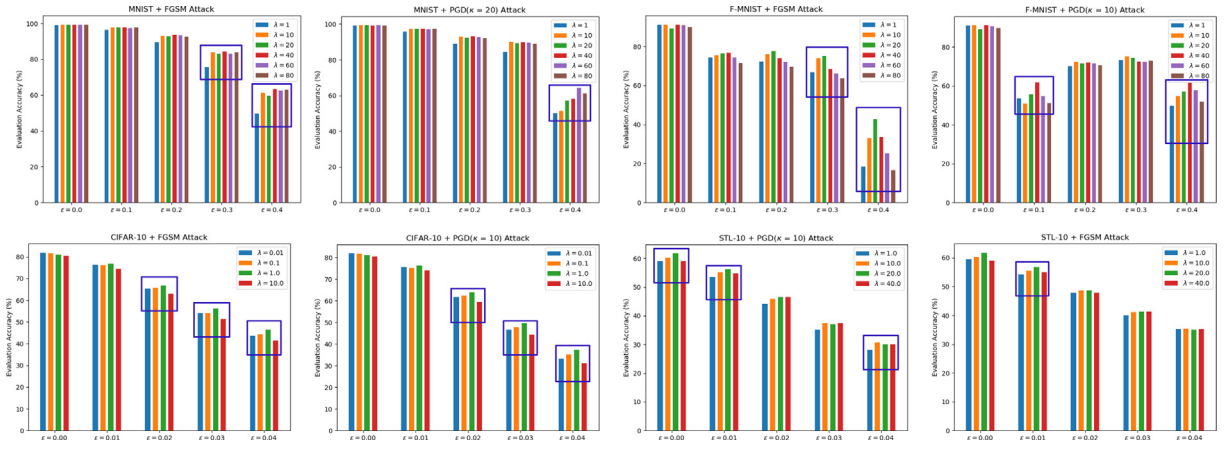


Fig. 6. Effects of λ for adversarial robustness on the MNIST, F-MNIST, CIFAR-10, and STL-10 datasets. *Zoom in for better visibility.*.

5.5. Visualizations of robust feature learning

Fig. 7 illustrates the gradual progression of our robust feature learning process, by visualizing the intermediate features extracted from adversarial examples when the deep network is trained for different epochs. From the first column of the figure, we find that the adversarial perturbations in the images progressively modify the features extracted from the different depths of the deep network. Adversarial perturbations enlarge the difference of input examples in the feature space, and then the gap in the feature space leads to a distortion in the decision space; this causes deep networks to make ill-advised predictions under different adversaries. We first train the network for 100 epochs on clean examples, and then optimize the model with our proposed defense mechanism for another 200 epochs with an appropriate initial learning rate of 0.01. We should note that the dataset employed here is CIFAR-10, and the network is the ResNet-110 as shown in Fig. 4. Fig. 7 shows the feature maps extracted when the network is trained with the proposed method for $10, 50, 100$ and 200 epochs. Through aligning intermediate features extracted from adversarial examples with features extracted from normal images, the model can capture meaningful visual concepts from images and learn robust features for image classification. Thus, the similarity between features extracted from different domains is improved, and the generalization ability under various adversaries is enhanced as well.

5.6. Embedding visualization of defense effects

To visualize the high-dimensional data in the decision space for classification tasks, we apply the t-distributed stochastic neighbor embedding (t-SNE) tool to project high-dimensional data into a low-dimensional embedding space. Fig. 8 illustrates the embedding points projected into a 2D space on three test datasets. From the first column of the figure, we see that a fully optimized classifier can learn a good embedding on clean images from the MNIST, F-MNIST, and CIFAR-10 datasets, and the embedding points for different classes are clearly separated. In the figure, the points with different colors indicate images in different classes. However, without an efficient defense mechanism, the classifier can easily be distorted by adversarial perturbations. As shown in the middle column of the figure, embedding points for different classes are mixed in the embedding space under the PGD attack. The last column of the figure shows that our defense approach can make good adjustment to the adversarial examples and generate well-separated embeddings in the decision space, which visually indicates that the proposed method can improve the adversarial robustness of deep networks.

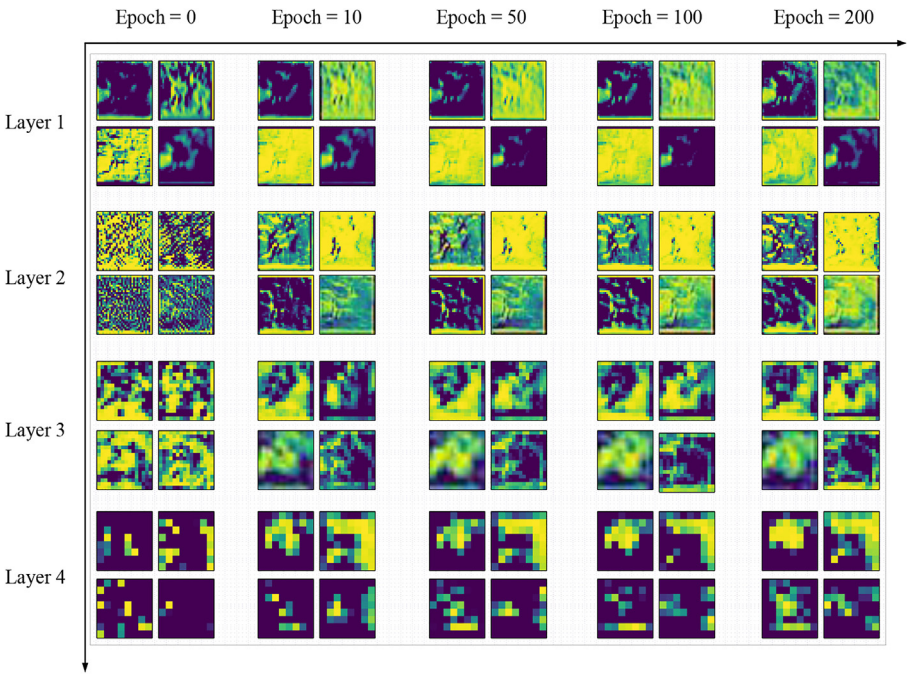


Fig. 7. Visualizations of robust feature learning. We visualize the feature maps extracted from the different layers of the ResNet-110 network on the CIFAR-10 dataset. With the aid of the proposed learning methodology, the deep network can capture robust feature representations from adversarial examples. *Best viewed in color.* (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

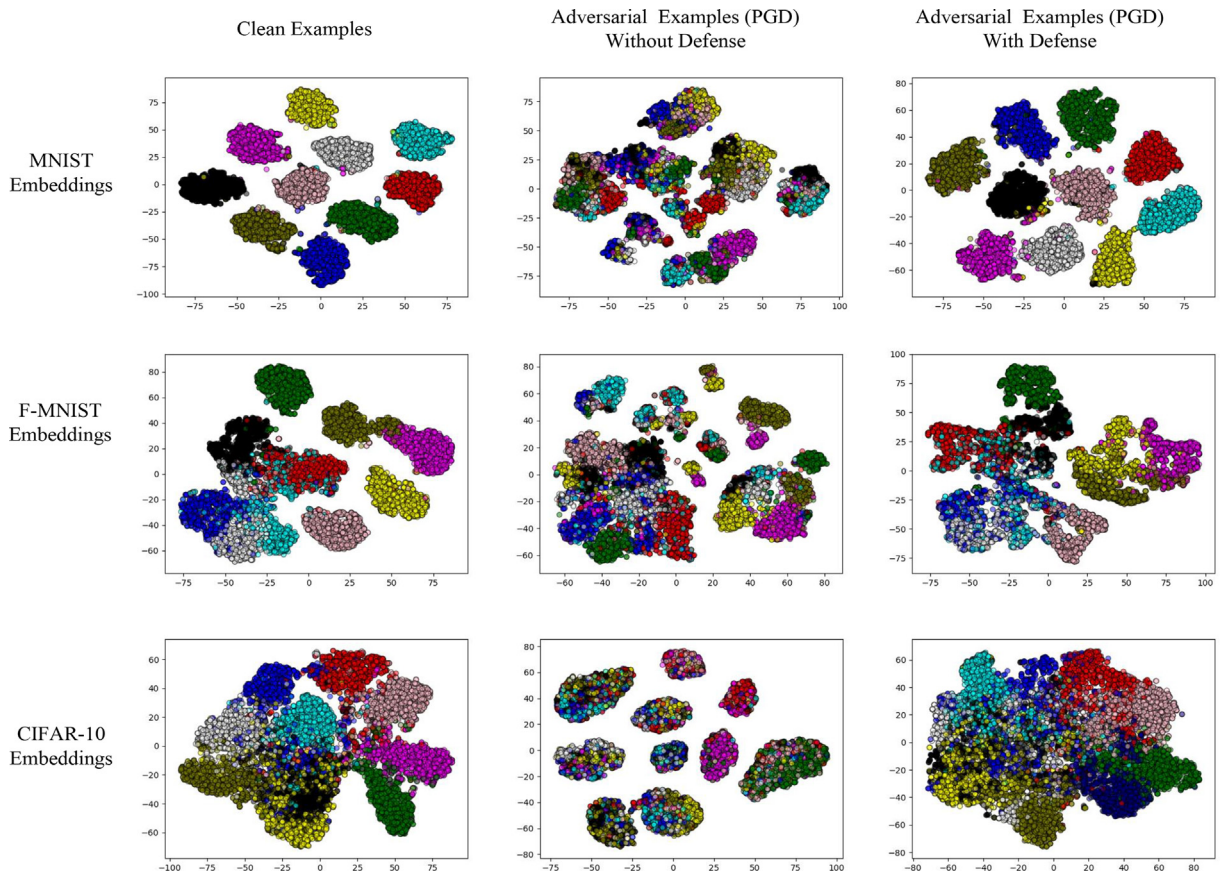


Fig. 8. t-SNE visualizations for the embeddings of clean test data and adversarial testing examples generated by the PGD method on classification datasets. Embedding points of different classes are separated on a 2D space; however, under adversarial attacks, some embedding points of different classes are mixed. Nevertheless, our proposed method ensures that these deep networks can generate well-separated embeddings on both clean and adversarial examples. *Best viewed in color.* (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6. Conclusion

Recent studies suggest that adversarial attacks pose an ever-increasing threat to deep-learning-based safety and security-critical applications. In this study, we investigated adversarial defense from the perspective of domain adaptation in the intermediate feature space. Our findings provide evidence that the adversarial robustness of models can be improved by progressively aligning the features extracted from adversarial examples with those extracted from legitimate images. We extensively evaluated the proposed method against both single-step and iterative attacks in white-box settings and showed that our defense method exhibits a higher robustness than existing state-of-the-art methods on four benchmark datasets. Moreover, empirical evaluations demonstrated the excellent generalization ability on diverse adversaries. Thus, the proposed defense mechanism can provide good reliability and security against adversarial perturbations in deep neural networks.

CRediT authorship contribution statement

Xiaoqin Zhang: Conceptualization, Data curation, Writing - review & editing, Supervision, Resources. **Jinxin Wang:** Methodology, Investigation, Software, Writing - original draft. **Tao Wang:** Writing - review & editing, Software, Visualization. **Runhua Jiang:** Writing - review & editing, Software. **Li Zhao:** Funding acquisition, Project administration, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

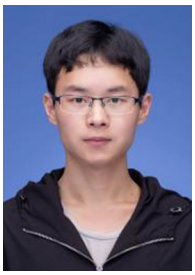
This work was supported in part by the National Natural Science Foundation of China [Grant Nos. 61922064, U2033210], in part by the Zhejiang Provincial Natural Science Foundation [Grant Nos. LR17F030001, LQ19F020005], in part by the Project of science and technology plans of Wenzhou City [Grant Nos. C20170008, ZG2017016].

References

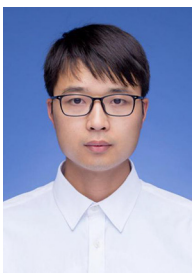
[1] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: IEEE Symposium on Security and Privacy, IEEE, 2017, pp. 39–57.
[2] S. Chen, Z. He, C. Sun, J. Yang, X. Huang, Universal adversarial attack on attention and the resulting dataset damagenet, IEEE Transactions on Pattern Analysis and Machine (2020), Intelligence.
[3] Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, Y. Kalantidis, Graph-based global reasoning networks, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 433–442.
[4] A. Coates, A. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, in: Proceedings of Conference on Artificial Intelligence and Statistics, 2011, pp. 215–223.
[5] N. Courty, R. Flamary, D. Tuia, A. Rakotomamonjy, Optimal transport for domain adaptation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (2017) 1853–1865.
[6] M. Cuturi, Sinkhorn distances: Lightspeed computation of optimal transport, in: Advances in Neural Information Processing Systems, 2013, pp. 2292–2300.
[7] A. Dabouei, S. Soleymani, F. Taherkhani, J. Dawson, N.M. Nasrabadi, Exploiting joint robustness to adversarial perturbations, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 1122–1131.
[8] Das N., Shanbhogue M., Chen S.T., Hohman F., Chen L., Kounavis M.E., Chau D.H., Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression, 2017, arXiv preprint arXiv:1705.02900..
[9] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, J. Li, Boosting adversarial attacks with momentum, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9185–9193.
[10] Dziugaite G.K., Ghahramani Z., Roy D.M., A study of the effect of jpg compression on adversarial images, 2016, arXiv preprint arXiv:1608.00853..
[11] Goodfellow I., Nips 2016 tutorial: Generative adversarial networks, 2016, arXiv preprint arXiv:1701.00160..
[12] Goodfellow I.J., Shlens J., Szegedy C., Explaining and harnessing adversarial examples, 2014, arXiv preprint arXiv:1412.6572..
[13] C. Guo, M. Rana, M. Cisse, L. van der Maaten, Countering adversarial images using input transformations, in: Proceedings of International Conference on Learning Representations, 2018.
[14] F. Guo, Q. Zhao, X. Li, X. Kuang, J. Zhang, Y. Han, Y.A. Tan, Detecting adversarial examples via prediction difference for deep neural networks, Information Sciences 501 (2019) 182–192.
[15] Z. Huang, J. Wang, X. Fu, T. Yu, Y. Guo, R. Wang, Dc-spp-yolo: Dense connection and spatial pyramid pooling based yolo for object detection, Information Sciences (2020).
[16] A. Jeddi, M.J. Shafiee, M. Karg, C. Scharfenberger, A. Wong, Learn2perturb: an end-to-end feature perturbation learning to improve adversarial robustness, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 1241–1250.
[17] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[18] Kurakin A., Goodfellow I., Bengio S., Adversarial examples in the physical world, 2016, arXiv preprint arXiv:1607.02533..
[19] Luo Y., Boix X., Roig G., Poggio T., Zhao Q., Foveation-based mechanisms alleviate adversarial examples, 2015, arXiv preprint arXiv:1511.06292..
[20] Madry A., Makelov A., Schmidt L., Tsipras D., Vladu A., Towards deep learning models resistant to adversarial attacks, 2017, arXiv preprint arXiv:1706.06083..
[21] S.M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2574–2582.
[22] A. Mustafa, S.H. Khan, M. Hayat, R. Goecke, J. Shen, L. Shao, Deeply supervised discriminative learning for adversarial defense, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020).
[23] A. Mustafa, S.H. Khan, M. Hayat, J. Shen, L. Shao, Image super-resolution as a defense against adversarial attacks, IEEE Transactions on Image Processing 29 (2019) 1711–1724.
[24] Peyré G., Cuturi M., Computational optimal transport, 2018, arXiv preprint arXiv:1803.00567..
[25] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[26] Samangouei P., Kabkab M., Chellappa R., Defense-gan: Protecting classifiers against adversarial attacks using generative models, 2018, arXiv preprint arXiv:1805.06605.
[27] Y. Shi, Y. Han, Q. Tian, Polishing decision-based adversarial noise with a customized sampling, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2020.
[28] Y. Shi, Y. Han, Q. Zhang, X. Kuang, Adaptive iterative attack towards explainable adversarial robustness, Pattern Recognition 107309 (2020).
[29] Y. Shi, S. Wang, Y. Han, Curls & whey: Boosting black-box adversarial attacks, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6519–6527.
[30] Song C., He K., Wang L., Hopcroft J.E., Improving the generalization of adversarial training with domain adaptation, 2018, arXiv preprint arXiv:1810.00740..
[31] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, 2013, arXiv preprint arXiv:1312.6199.
[32] Tolstikhin I., Bousquet O., Gelly S., Schoelkopf B., Wasserstein auto-encoders, 2017, arXiv preprint arXiv:1711.01558..
[33] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, P. McDaniel, Ensemble adversarial training: attacks and defenses, 2017, arXiv preprint arXiv:1705.07204.
[34] C. Villani, Optimal Transport: Old and New, vol. 338, Springer Science & Business Media, 2008.
[35] Y. Xiao, C.M. Pun, B. Liu, Adversarial example generation with adaptive gradient search for single and ensemble deep neural network, Information Sciences (2020).
[36] Xie C., Wang J., Zhang Z., Ren Z., Yuille A., Mitigating adversarial effects through randomization, 2017, arXiv preprint arXiv:1711.01991.
[37] J. Xu, H. Liu, D. Wu, F. Zhou, C.Z. Gao, L. Jiang, Generating universal adversarial perturbation with resnet, Information Sciences (2020).
[38] X. Zhang, R. Jiang, T. Wang, P. Huang, L. Zhao, Attention-based interpolation network for video deblurring, Neurocomputing (2020).
[39] X. Zhang, D. Wang, Z. Zhou, Y. Ma, Robust low-rank tensor recovery with rectification and alignment, IEEE Transactions on Pattern Analysis and Machine Intelligence (2019).

[40] X. Zhang, T. Wang, J. Wang, G. Tang, L. Zhao, Pyramid channel-based feature attention network for image dehazing, Computer Vision and Image Understanding 103003 (2020).
[41] X. Zhu, Z. Li, X. Li, S. Li, F. Dai, Attention-aware perceptual enhancement nets for low-resolution image classification, Information Sciences 515 (2020) 233–247.

Xiaoqin Zhang received the B.Sc. degree in electronic information science and technology from Central South University, China, in 2005 and Ph.D. degree in pattern recognition and intelligent system from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China, in 2010. He is currently a professor in Wenzhou University, China. His research interests are in pattern recognition, computer vision and machine learning. He has published more than 100 papers in international and national journals, and international conferences, including IEEE T-PAMI, IJCV, IEEE T-IP, IEEE T-IE, IEEE T-C, ICCV, CVPR, NIPS, IJCAI, AAAI, and among others.



Jinxin Wang is currently a graduate student at College of Computer Science and Artificial Intelligence, Wenzhou University, China. He received his bachelor's degree in information and computing science at Wenzhou University. His research interests include reinforcement learning, image generation and deep learning.



Tao Wang is currently a graduate student at College of Computer Science and Artificial Intelligence, Wenzhou University, China. He received the B.Sc. degree in information and computing science from Hainan Normal University, China, in 2018. His research interests include several topics in computer vision and machine learning, such as object tracking, image/video quality restoration, adversarial learning, image-to-image translation and reinforcement learning.

Runhua Jiang is currently a graduate student majoring in computer software and theory at College of Computer Science and Artificial Intelligence, Wenzhou University, China. He received his B.Sc. degree in department of information science at Tianjin University of Finance and Economy, China. His research interests include several computer vision tasks, such as image/video restoration, crowd counting, visual understanding, and video question answering.

Jiawei Xu is currently working in College of Computer Science and Artificial Intelligence, Wenzhou University, China from 2020. He was working in School of Computing, Newcastle University from 2017 to 2019. He was a visiting postdoc (2016) in National Institutes of Health, United States. He received Ph.D. (2012–2015) in School of Computer Science, University of Lincoln, United Kingdom, M.Sc. degree (2009–2011) from Department of Electronic Engineering, Hallym University, Korea and B.S. (2003–2007) from Department of Automotive Engineering, University of Shanghai for Science and Technology, Shanghai, China. His research interests include human vision modeling and its industrial application.

Li Zhao received the B.Sc. degree in automation in 2005 and MEng degree in control theory and control engineering in 2008 from Central South University, China. She is currently an assistant researcher in Wenzhou University. Her research interests are in pattern recognition, computer vision, and machine learning.