Complex & Intelligent Systems (2025) 11:155
https://doi.org/10.1007/s40747-024-01770-z

ORIGINAL ARTICLE

Mape: defending against transferable adversarial attacks using multi-source adversarial perturbations elimination

Xinlei Liu[1] · Jichao Xie[1] · Tao Hu[1] · Peng Yi[1,2] · Yuxiang Hu[1] · Shumin Huo[1] · Zhen Zhang[1]

Received: 17 October 2024 / Accepted: 29 December 2024 / Published online: 17 January 2025

Abstract

Neural networks are vulnerable to meticulously crafted adversarial examples, leading to high-confidence misclassifications in image classification tasks. Due to their consistency with regular input patterns and the absence of reliance on the target model and its output information, transferable adversarial attacks exhibit a notably high stealthiness and detection difficulty, making them a significant focus of defense. In this work, we propose a deep learning defense known as multi-source adversarial perturbations elimination (MAPE) to counter diverse transferable attacks. MAPE comprises the **single-source adversarial perturbation elimination** (SAPE) mechanism and the pre-trained models probabilistic scheduling algorithm (PPSA). SAPE utilizes a thoughtfully designed channel-attention U-Net as the defense model and employs adversarial examples generated by a pre-trained model (e.g., ResNet) for its training, thereby enabling the elimination of known adversarial perturbations. PPSA introduces model difference quantification and negative momentum to strategically schedule multiple pre-trained models, thereby maximizing the differences among adversarial examples during the defense model's training and enhancing its robustness in eliminating adversarial perturbations. MAPE effectively eliminates adversarial perturbations in various adversarial examples, providing a robust defense against attacks from different substitute models. In a black-box attack scenario utilizing ResNet-34 as the target model, our approach achieves average defense rates of over 95.1% on CIFAR-10 and over 71.5% on Mini-ImageNet, demonstrating state-of-the-art performance.

Keywords Deep learning security · Pattern recognition · Image classification · Adversarial example · Adversarial defense

Introduction

The convolutional neural network (CNN) is a deep and feedforward neural network that incorporates convolution operations, which has been used widely in diverse visual tasks [1], including image recognition [2], object detection [3], and semantic segmentation [4]. However, recent research has shown that there exist adversarial examples [5–8] that do not affect human judgment but can perplex network models. For instance, when a classification model correctly identifies a house finch in Fig. 1, and then a meticulously designed adversarial perturbation is introduced, the model misclassifies it as a catamaran. However, from a human perspective, there is not apparent difference in cognition between the example before and after the inclusion of the adversarial perturbation. Adversarial examples can lead to potential security vulnerabilities, thus affecting the reliability and stability of image classification systems [9, 10]. Therefore, defending against adversarial attacks has become one of the important challenges in protecting deep learning models and ensuring system security.

Transferable adversarial attack [11–15] is a classic black-box attack, which refers to generating adversarial examples on a substitute model and then using them to deceive the target model. In comparison to another typical black-box attack method, query attacks [16–18], transferable attacks are more similar to conventional input patterns. As they do not require sending a large number of query examples to the target model, and are entirely independent of the target model and its output information. Due to their higher stealthiness and greater

Xinlei Liu and Jichao Xie have contributed equally to this work.

✉ Tao Hu
hutaondsc@163.com

✉ Peng Yi
yipengndsc@163.com

[1] Information Engineering University, Zhengzhou, China 450002

[2] Key Laboratory of Cyberspace Security, Ministry of Education, Zhengzhou, China 450002

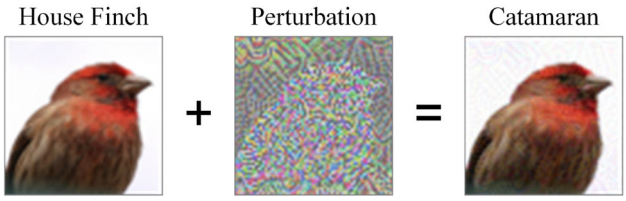

Fig. 1 Clean example and adversarial example. When the adversarial perturbations are added to a house finch, it is misclassified as a catamaran by the classification model

difficulty in detection, along with more relaxed implementation conditions, transferable attacks have become one of the most prevalent adversarial attack methods currently. Consequently, we will designate it as the defensive target of this study. Differing from adversarial training [19, 20] aimed at enhancing the robustness of the target model, input transformations [5, 6, 21] defend against adversarial attacks by applying random transformations to disrupt adversarial perturbations, or by denoising them to eliminate adversarial perturbations. In defending against transferable attacks that are more aligned with real-world scenarios, input transformations demonstrate a more outstanding defense effectiveness [22, 23].

In input transformation, random transformation methods [24, 25] disrupt the overall structural adversarial perturbation by randomly rotating, scaling, and translating, enabling the target model to correctly classify adversarial examples. However, random transformations also obscure the original data distribution in the input examples, inevitably leading to a significant decrease in its classification accuracy. Denoising methods [26, 27] eliminate adversarial perturbations from input examples by introducing denoising blocks in the classification model or deploying denoisers externally to the model, exhibiting strong specificity and mediocre generalizability. Specifically, the defense effectiveness of denoising methods is stronger when the substitute model is structurally similar to the target model, while it significantly decreases when there is a large difference in structure between the substitute model and the target model.

In this article, we propose the *multi-source adversarial perturbations elimination* (MAPE) to assist target models in defending against diverse transferable attacks. At a low level, a channel-attention U-Net (CAU-Net) is utilized as the defense model, reconstructing the adversarial examples by eliminating the perturbations within them. Subsequently, the defense model is trained by computing the label losses between reconstructed examples and clean examples. As the adversarial examples originate from a single classification model, we refer to this low-level mechanism as *single-source adversarial perturbation elimination* (SAPE). At a high level, we introduce several distinct pre-trained models and propose the *pre-trained models probabilistic scheduling algorithm* (PPSA). Based on the pre-trained model's output scores and scheduling records, we define two key components in PPSA: model difference probability and negative momentum probability. The former represents an intrinsic characteristic of the model in the scheduling process, while the latter serves as a regularization factor to adjust the usage of models. PPSA effectively combines these two probabilities to maximize the differences between adjacent scheduled pre-trained models. By integrating SAPE and PPSA, MAPE enhances the defense model's robustness and generalization in eliminating adversarial perturbations. Compared to previous defense strategies, MAPE exhibits superior effectiveness in countering adversarial attacks. These have been verified in "Experiments" section and "Further evaluations" section.

We summarize the main contributions as follows:

- We propose a deep learning defense known as *multisource adversarial perturbation elimination* (MAPE), which utilizes a CAU-Net as the defense model and is capable of eliminating adversarial perturbations in various adversarial examples.
- To the best of our knowledge, we are the first to introduce *negative momentum* as a regularization factor for dynamically adjusting the usage of certain elements, as well as to *quantify the model difference* based on the output scores on the same dataset.
- The evaluation demonstrates that MAPE exhibits strong generalization capability and cross-model defense characteristics, effectively countering transferable adversarial attacks from various substitute models in a black-box attack environment.

Related work

We study related work from three perspectives: adversarial examples, attack methods for generating transferable adversarial examples, and defense methods for resisting this adversarial examples.

Adversarial examples

The generation of adversarial examples can be represented as a constrained optimization problem. Let $\mathcal{C}(\cdot)$ be the pre-trained classification model such that $\mathcal{C}(\boldsymbol{x})$: $\boldsymbol{x} \rightarrow \boldsymbol{\ell}$, where $\boldsymbol{x} \in \mathbb{R}^m$ is a clean example and $\boldsymbol{\ell} \in \mathbb{Z}^+$ is the output of the model. Let $\mathcal{A}(\cdot)$ be the attack method used by the attacker, denoted as $\mathcal{A}(\boldsymbol{\theta}, \boldsymbol{x}) \rightarrow \boldsymbol{\rho}$, where $\boldsymbol{\theta}$ is the parameter of the model and $\boldsymbol{\rho} \in \mathbb{R}^m$ is the adversarial perturbation. To ensure that the semantic information in natural examples used for human recognition is not compromised, the generated adversarial perturbation $\boldsymbol{\rho}$ is often bounded by a norm. For

example, constraining the perturbation $\boldsymbol{\rho}$ within $\|\boldsymbol{\rho}\|_p < \epsilon$, where $\|\boldsymbol{\rho}\|_p$ denotes the L_p norm and ϵ is the adversarial perturbation budget. The adversarial example $\bar{\boldsymbol{x}} \in \mathbb{R}^m$ is obtained by adding the adversarial perturbation $\boldsymbol{\rho}$ to the natural example $\boldsymbol{x}$, represented as $\boldsymbol{x} + \boldsymbol{\rho} \rightarrow \bar{\boldsymbol{x}}$. The generation problem of adversarial examples is essentially the problem of solving adversarial perturbations, which can be represented by the following constrained optimization process:

$$\arg\max_{\boldsymbol{\rho}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x} + \boldsymbol{\rho}, \boldsymbol{\ell}) \text{ s.t. } \|\boldsymbol{\rho}\|_p < \epsilon . \quad (1)$$

In Eq. (1), $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x} + \boldsymbol{\rho}, \boldsymbol{\ell})$ represents the loss of the model with parameter $\boldsymbol{\theta}$ regarding adversarial example $\boldsymbol{x} + \boldsymbol{\rho}$ and label $\boldsymbol{\ell}$, typically computed using the cross-entropy loss function. Therefore, the generation of adversarial examples can be summarized as finding limited adversarial perturbations that maximize the model's loss. Adversarial examples are typically generated through gradient-based methods [9, 28] when the model's parameters and defense strategies are known to the attacker. In cases where the model's parameters and defense strategies are unseen to the attacker, adversarial examples are usually generated from substitute models based on their transferability [10, 11, 13].

Attack methods

Transferable attacks are built on the transferability of adversarial examples [29]. Therefore, the simplest transferable attacks use white-box attack methods as their means of attack, with the difference from regular white-box attacks being that substitute models are treated as target models. Representative methods include the fast gradient sign method (FGSM) [28], the basic iterative method (BIM) [9], and the projected gradient descent (PGD) [19]. The FGSM is a one-step gradient-based method that computes norm-bounded perturbations, while BIM and PGD seek better solutions by optimizing the gradient direction through multiple iterations [6]. The core concept of the three methods mentioned above is to perform gradient ascent on the loss surface of the model to deceive it, which also forms the basis of many adversarial attacks.

However, some stronger transferable attacks improve the transferability of adversarial examples by integrating attack techniques, transforming images, optimizing gradient variances, among other strategies. Diverse inputs iterative FGSM (DIM) [11] applies random image transformations, diversifying the input information for each iteration to enhance the transferability of adversarial examples. Built upon momentum iterative FGSM (MI-FGSM) [30] and Nesterov iterative FGSM (NI-FGSM) [12], respectively, variance tuning MI-FGSM (VMIM) and variance tuning NI-FGSM (VNIM) [13] adjust the current gradient by using the gradient variance from the previous iteration to optimize the gradient direction and escape local optima. Additionally, there are also attack methods that aim to enhance the transferability of adversarial examples by integrating gradients from multiple iterations or multiple models, such as large geometric vicinity (LGV) [31], transferable adversarial attack based on integrated gradients (TAIG) [32] and adaptive model ensemble adversarial attack (AdaEA) [33].

Defense methods

Adversarial defense methods are generally divided into two main classes, including adversarial training and input transformation. Adversarial training [19] is the data augmentation technique that enhances the robustness of the target model by adding adversarial examples to the training data. TRADES [20] decomposes the robust error of adversarial examples into the sum of natural error and boundary error, which acts as its design principle in defending against adversarial attacks.

Input transformation methods aim to eliminate the attack nature of adversarial examples, thereby reducing the recognition difficulty for the target model. They are the main force in defending against black-box attacks. These methods can be categorized into random transformation methods and denoising methods. In random transformation methods, total variance minimization (TVM) [34] randomly selects a small group of pixels and reconstructs the "simplest" image that does not include adversarial perturbations. Pixel deflection [35] corrupts adversarial perturbations by redistributing the pixel values and applying adaptive soft-thresholding in the wavelet domain. Mixup inference [23] overlays input examples randomly with other clean examples to reduce the adversarial nature of the input examples. In denoising methods, JPEG compression [36] removes certain high-frequency components and image details through discrete cosine transformation and quantization, thereby enabling defense against adversarial examples with low perturbation budgets. Similarly, Gaussian blurring (smoothing) [37] convolves a Gaussian kernel with adversarial examples, blurring image details to disrupt the adversarial perturbations present in the adversarial examples. Feature denoising method [27] adds denoising blocks in the classification model and combines them with adversarial training to enhance the model's adversarial robustness. High-level representation guided denoiser (HGD) [26] revises the loss function to pull adversarial examples back to the original clean distribution for improving their classification accuracy. Learning defense transformation (LDT) [25] employs parameterizing the affine transformations and the boundary information of neural network as a defense mechanism against adversarial attacks.

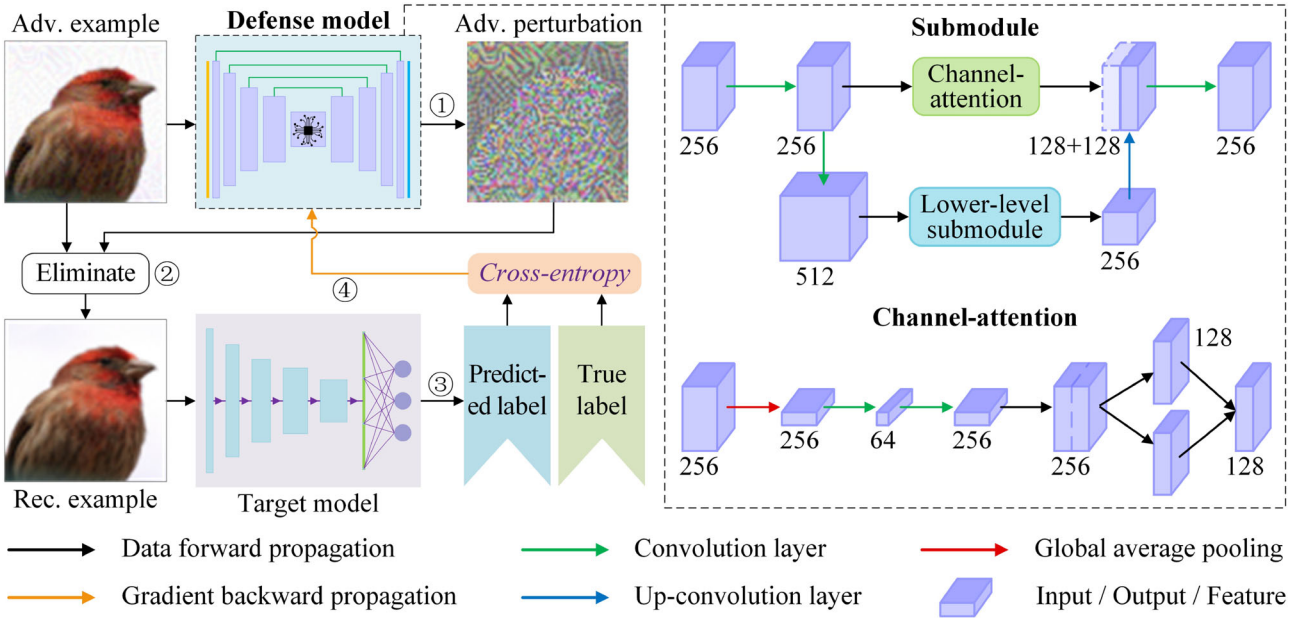

Fig. 2 Single-source adversarial perturbation elimination (SAPE) mechanism

Methodology

Random transformation methods, such as TVM and pixel deflection, do not depend on the target model, resulting in similar defense effectiveness against both known and unknown types of adversarial attacks, although neither is particularly high. Deep learning defense methods, such as HGD and LDT, are closely coupled with the target model. Due to the differences between the target model and the substitute model, their effectiveness in defending against unknown types of adversarial attacks is significantly diminished.

To improve defense effectiveness against unknown types of adversarial attacks, we propose a deep learning defense known as multi-source adversarial perturbations elimination (MAPE). MAPE primarily consists of the single-source adversarial perturbation elimination (SAPE) mechanism and the pre-trained models probabilistic scheduling algorithm (PPSA). SAPE serves as the foundational method for MAPE, aiming to enable the defense model to eliminate known adversarial perturbations. PPSA acts as the organizational framework of MAPE, focusing on achieving the ability to eliminate unknown types of adversarial perturbations and improving its robustness.

Single-source adversarial perturbation elimination

As shown in Fig. 2, SAPE consists mainly of a target model $\mathcal{C}(\cdot)$ with the parameter $\boldsymbol{\theta}$ and a defense model $\mathcal{E}(\cdot)$ with the parameter $\boldsymbol{\zeta}$. The target model $\mathcal{C}(\cdot)$ is the commonly used classification model such as ResNet [2], GoogLeNet [38], MobileNet [39], etc. The defense model $\mathcal{E}(\cdot)$ is typically a neural network model for image-to-image generation. In this paper, we integrate a U-Net [40] with channel-attention mechanism [41] as a defense model, referred to as CAU-Net. Compared to generative adversarial networks (GANs) [42] and diffusion models [43], U-Net requires less computational cost and is easier to train. Additionally, the primary comparison methods, HGD and LDT, also utilize U-Net or its variants as defense models; therefore, selecting U-Net enhances the credibility of the experimental results. We also optimize its structure to enhance its capability in extracting adversarial perturbations. The optimized CAU-Net is constructed with multiple nested submodules. Except for the lowest-level submodules, each submodule nests a lower-level submodule and contains a channel-attention mechanism layer similar to a residual connection. This work utilizes a CAU-Net with five submodules. The right side of Fig. 2 illustrates the third submodule along with its specific components. The number of both input and output data channels is 256.

Deployed outside the target model, the defense model $\mathcal{E}(\cdot)$ is responsible for extracting and eliminating the adversarial perturbation $\hat{\boldsymbol{\rho}}$ from the adversarial example $\bar{\boldsymbol{x}}$, playing a role similar to that of antivirus software. Note that the adversarial perturbation $\hat{\boldsymbol{\rho}}$ extracted from adversarial examples by the defense model is not equivalent to the adversarial perturbation $\boldsymbol{\rho}$ added by an attacker to clean examples. It is desirable to enhance the similarity of the data distributions between the two, which is the goal pursued by the defense model.

The training method of SAPE is as follows. Firstly, the adversarial example generated by the target model is input into the defense model to extract the adversarial perturbation, which is then removed to obtain the reconstructed example

$\hat{\boldsymbol{x}}$. Subsequently, the reconstructed example is fed into the target model to obtain the probability distribution of predicted labels. Finally, the cross-entropy is computed between the probability distribution of the predicted label and that of the true label, after which the defense model's weights are updated with the back-propagation algorithm. The optimization objective of SAPE can be expressed by Eq. (2).

$$\arg\min_{\boldsymbol{\zeta}} \mathcal{L}\left[\boldsymbol{\theta}, \boldsymbol{x}+\boldsymbol{\rho}-\mathcal{E}(\boldsymbol{x}+\boldsymbol{\rho}), \boldsymbol{\ell}\right] . \tag{2}$$

SAPE aims to equip the defense model with the capability to eliminate known adversarial perturbations. This capability serves as the foundation for MAPE. Algorithm 1 summarizes the detailed training method of SAPE.

Algorithm 1 Detailed training method of SAPE.

Require: A defense model $\mathcal{E}$ with the parameter $\boldsymbol{\zeta}$, the target classification model (e.g., ResNet) with the parameter $\boldsymbol{\theta}$, clean examples $\boldsymbol{x}$, attack method $\mathcal{A}$, learning rate η and weight decay λ

Ensure: The well-trained defense model $\mathcal{E}$

1: Initiate the parameter $\boldsymbol{\zeta}$ of the defense model $\mathcal{E}$;
2: Freeze the parameters $\boldsymbol{\theta}$ of the target model;
3: **while** not converged **do**
4: Generate the adversarial example $\bar{\boldsymbol{x}}=\boldsymbol{x}+\mathcal{A}(\boldsymbol{\theta}, \boldsymbol{x})$;
5: Perform stratified sampling from clean examples and adversarial examples to create a mixed example $\ddot{\boldsymbol{x}} \leftarrow[\boldsymbol{x}, \bar{\boldsymbol{x}}]$;
6: Extract and eliminate the adversarial perturbation $\hat{\boldsymbol{x}}=\ddot{\boldsymbol{x}}-\mathcal{E}(\ddot{\boldsymbol{x}})$;
7: Compute the cross-entropy loss $l=\mathcal{L}(\boldsymbol{\theta}, \hat{\boldsymbol{x}}, \boldsymbol{\ell})$;
8: Update the parameter $\boldsymbol{\zeta} \leftarrow \boldsymbol{\zeta}-\eta\{\nabla_{\boldsymbol{\zeta}} l+\lambda \boldsymbol{\zeta}\}$.
9: **end while**

Pre-trained models probabilistic scheduling algorithm

During training SAPE, the adversarial examples used are solely derived from the target model. When the attacker's substitute model differs from the protected target model, the defense effectiveness of SAPE significantly decreases. In practical applications, it is quite common for the substitute model to differ from the target model.

Therefore, utilizing multiple pre-trained models and enhancing the difference among these models can diversify the adversarial examples used for training, thereby endowing the defense model with higher generalization capability and defensive performance. And the impact of model quantity on defensive performance depends on the differences between the newly added models and the previous models. Furthermore, to fully leverage the difference among these pre-trained models for training the defense model, we propose PPSA to strategically schedule them. PPSA refers to selecting a different pre-trained model based on the scheduling probability after training on current mini-batch. The newly selected pre-trained model will be used to generate adversarial examples of the next mini-batch. If the pre-trained model $C_i(i=1,2,\ldots,N)$ is used to generate adversarial examples in the kth mini-batch, then the probability of the pre-trained model $C_j(j=1,2,\ldots,N, j\neq i)$ being selected in the $(k+1)$th mini-batch can be expressed as

$$P_k^j = h\left\{P_{\text{diff}}^{(i,j)} \circ P_{\text{neg}}^{(k,j)}\right\} . \tag{3}$$

In Eq. (3), $h\{\cdot\}$ is the probability normalization transformation. For the variable $I_s(s=1,2,\ldots,N)$,

$$h\{I_s\} = \frac{I_s}{\sum_{t=1}^{N} I_t} . \tag{4}$$

$P_{\text{diff}}^{(i,j)}$ represents the model difference probability between the pre-trained models C_i and C_j, $P_{\text{neg}}^{(k,j)}$ represents the negative momentum probability of C_j at the kth mini-batch, and "$\circ$" denotes their Hadamard product. Below, we will discuss these two probability distributions separately.

Pre-trained models with greater differences are usually selected to generate adversarial examples during the training of adjacent mini-batches, as this can help the defense model adapt to diverse adversarial inputs and boosts its robustness against adversarial attacks. However, it is difficult to quantify the differences between different models by comparing their structures and parameters. We think that a model's output scores reflect its intrinsic characteristics to some extent; a larger difference in output scores indicates a greater difference between two models. Therefore, we propose a model differences calculating method based on output score. Suppose that pre-trained models have output scores $E_i(i=1,2,\ldots,N)$ on the same dataset, then the difference $W_{i,j}$ between models C_i and C_j can be expressed using L_1 norm Wasserstein distance as:

$$W_{i,j} = \inf_{\gamma\in\Gamma(E_i,E_j)} \int_{\mathbb{R}\times\mathbb{R}} |u-v| \mathrm{d}\gamma(u,v) . \tag{5}$$

In Eq. (5), $\Gamma(E_i, E_j)$ is the set of all joint distributions whose marginal distributions are E_i and E_j, respectively. γ represents a joint distribution that describes how to "transport" or "transfer" probability mass between the two distributions. $W_{i,j}$ has good mathematical properties, such as non-negativity $\left(W_{i,j}\geq 0\right)$ and symmetry $\left(W_{i,j}=W_{j,i}\right)$. Notably, the symmetry property is not present in cross-entropy and KL divergence. A higher value of $W_{i,j}$ indicates a greater difference between models C_i and C_j. After calculating the differences of the pre-trained models, the model difference probability can be expressed as

$$P_{\text{diff}}^{(i,j)} = h\left\{\ln\left(W_{i,j}+1\right)\right\} . \tag{6}$$

The model difference probability is a static attribute of the pre-trained models, remaining unchanged throughout the entire training period. This leads to a stable scheduling ratio among the various pre-trained models as training iterations increase, which is detrimental to the defense model's generalization. To address this issue, we propose the negative momentum probability as a regularization factor to dynamically adjust the model difference probability, expressed as

$$P_{\text{neg}}^{(k,j)} = h\left\{1 - h\left\{M_{k,j}\right\}\right\} . \tag{7}$$

In Eq. (7), $M_{k,j}$ represents the total number of times model C_j has been selected up to the kth mini-batch. It can be observed that $P_{\text{neg}}^{(k,j)}$ is negatively correlated with $M_{k,j}$, meaning that for pre-trained models that are frequently utilized, $P_{\text{neg}}^{(k,j)}$ will decrease the their scheduling probability; conversely, for models that are rarely used, $P_{\text{neg}}^{(k,j)}$ will increase their scheduling probability. Contrary to the effect of traditional "momentum," $P_{\text{neg}}^{(k,j)}$ suppresses the excessive use of high-difference-probability models during training, thereby enhancing the defense model's generalization toward other low-difference-probability models. Hence, $P_{\text{neg}}^{(k,j)}$ is referred to as "negative momentum" probability.

Algorithm 2 Detailed scheduling method of PPSA.

Require: The pre-trained model $C_i (i = 1, 2, \ldots, N)$ used in current mini-batch, output scores E of all pre-trained models on the same dataset, and their total scheduling times M_k up to the kth mini-batch.
Ensure: The scheduled pre-trained model C_r in $(k+1)$th mini-batch
1: Compute the model difference $W_{i,j}$ between the current pre-trained model C_i and the remaining pre-trained models $C_j (j = 1, 2, \ldots, N, j \neq i)$ based on L_1 norm Wasserstein distance;
2: Get the model difference probability $P_{\text{diff}}^{(i,j)}$ according to the model difference $W_{i,j}$;
3: Get the negative momentum probability $P_{\text{neg}}^{(k,j)}$ according to the total scheduling times M_k ;
4: Generate the scheduling probability P_k^j of the pre-trained model C_j by combining $P_{\text{diff}}^{(i,j)}$ and $P_{\text{neg}}^{(k,j)}$;
5: Determine the pre-trained model C_r based on the scheduling probability P_k^j ;
6: Update the total scheduling times of the pre-trained model C_r: $M_{k+1,r} = M_{k,r} + 1$.

PPSA combines model difference probability and negative momentum probability, dynamically scheduling different pre-trained models to generate diverse adversarial examples based on the current state and historical records. This approach helps improve the robustness of the defense model's capability in eliminating unknown types of adversarial perturbations. Algorithm 2 summarizes the detailed scheduling method of the PPSA.

Multi-source adversarial perturbations elimination

As shown in Fig. 3, MAPE consists primarily of PPSA and SAPE, with the training method outlined as follows: First, we calculate the output score between the current pre-trained model and the remaining pre-trained model. Then, utilize PPSA to select one of the remaining pre-trained models based on the model output scores and the model scheduling records. Finally, this scheduled pre-trained model is employed as the target model in SAPE for training the next mini-batch. Notably, to enhance the diversity of the training process, we use "mini-batch" as the switching cycle for pre-trained models instead of "epoch." The model scheduling records are preserved throughout the entire training cycle rather than being reset at the start of a new "epoch." Additionally, compared to SAPE, MAPE incorporates random adversarial perturbation budgets and random attack methods, which contribute to more generalized training for the defense model. Algorithm 3 summarizes the detailed training method of MAPE.

The process of utilizing MAPE to defend against adversarial attacks is illustrated in Fig. 4. The defense model is deployed externally to the target model and is responsible for extracting and eliminating implicit adversarial perturbations from input examples, thereby diminishing the effectiveness of adversarial attacks. The reconstructed examples will then be input to the target model for classification. If the input examples are clean natural images, the extracted adversarial perturbations are meaningless and do not affect normal classification.

Algorithm 3 Detailed training method of MAPE.

Require: A defense model $\mathcal{E}$ with the parameter $\boldsymbol{\zeta}$, N pre-trained classification models with the parameters $\boldsymbol{\theta}_n (n = 1, 2, \ldots, N)$, clean examples $\boldsymbol{x}$, attack methods $\mathcal{A}_{n'} (n' = 1, 2, \ldots, N')$, learning rate η and weight decay λ
Ensure: Robust defense model $\mathcal{E}$
1: Initiate the parameter $\boldsymbol{\zeta}$ of the defense model $\mathcal{E}$;
2: Freeze the parameters $\boldsymbol{\theta}_n$ of all pre-trained models;
3: **while** not converged **do**
4: Schedule a pre-trained model C_r according to **PPSA**;
5: Randomly create a sequence of adversarial perturbation budgets $[\epsilon_1, \epsilon_2, \ldots, \epsilon_{N'}] \leftarrow (4/255, 12/255)$;
6: Randomly select an attack method to generate adversarial examples $\bar{\boldsymbol{x}}_{n'} = \boldsymbol{x} + \mathcal{A}_{n'}(\boldsymbol{\theta}_{r_{n'}}, \boldsymbol{x})$;
7: Perform stratified sampling from clean examples and adversarial examples to create a mixed example $\ddot{\boldsymbol{x}} \leftarrow [\boldsymbol{x}, \bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_2, \ldots, \bar{\boldsymbol{x}}_{N'}]$;
8: Execute steps 6, 7, and 8 of **SAPE**.
9: **end while**

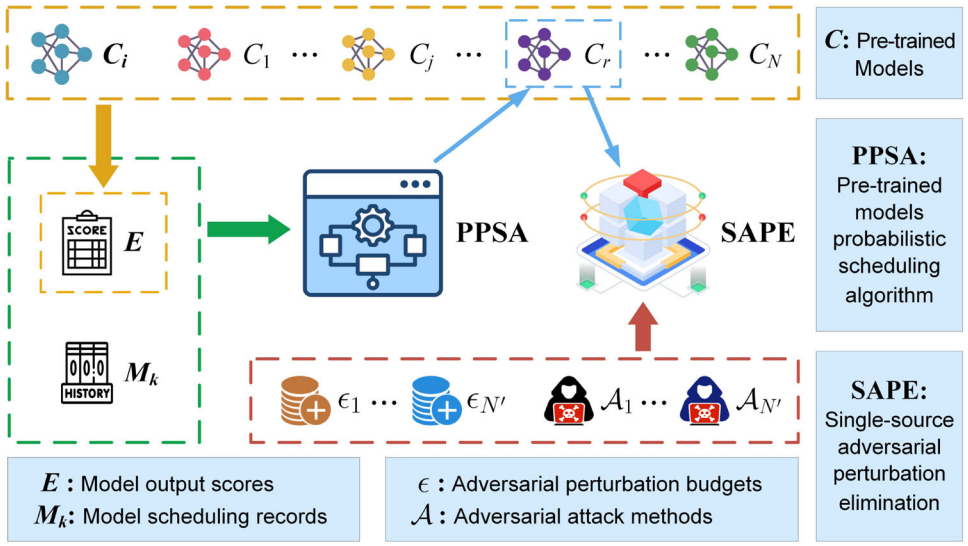

Fig. 3 Deep learning defense known as multi-source adversarial perturbations elimination (MAPE)

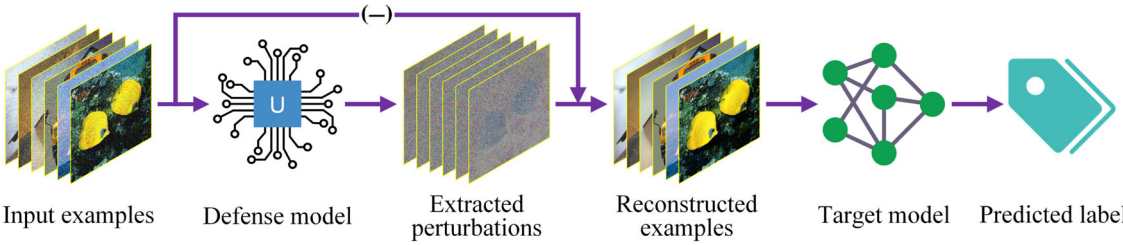

Fig. 4 Utilizing MAPE to defend against adversarial attacks

Experiments

Experimental setup

Attackers. The proposed method focuses on defending against transferable adversarial attacks in image classification. The black-box attack environment represents the most common real-world scenario and is applicable in "Existence of MAPE" section to "Defending against integrated transferable attacks" section and "Further evaluations" section. Additionally, "Defending against strong substitute model attacks and adaptive attacks" section discusses the less prevalent scenarios involving gray-box and white-box attacks. In a gray-box attack environment, attackers can acquire training data and model architecture from benign users, creating substitute models that align with the target model and defense architecture but are initialized differently, thereby enabling strong substitute model attacks. In a white-box attack environment, attackers have complete access to the target model, including all information about the model and its defense strategy, enabling precise adaptive attacks. To emphasize the accuracy of the evaluations, all attack methods in this paper belong to the more potent non-targeted adversarial attacks.

Datasets. We use CIFAR-10, CIFAR-100, and Mini-ImageNet as the evaluation datasets for this work. The resolutions of CIFAR-10 and CIFAR-100 remain unchanged, and the resolution of Mini-ImageNet is set to 64×64. Mini-ImageNet contains 100 classes, all of which are utilized. For each class, 480 randomly selected images are assigned to the training set, while the remaining 120 images are designated for the test set. In the experiments detailed in this paper, all input example values are constrained within the range of 0 to 1, signifying that the entirety of input data comprises image examples.

Classification models. During the training of MAPE, we employ DenseNet [44], DPN [45], GoogLeNet [38], MobileNetV2 [39], PyramidNet [46], RegNet [47], ResNet [2], ResNeXt [48], SENet [41], and WideResNet (WRN) [49] as the pre-trained models for crafting adversarial examples. In the evaluation phase, ResNet [2], ResNetV2 [50], ShuffleNetV2 [51], VGG [52], and Vision Transformer [53] are as substitute models used by attackers and utilized to create adversarial examples for comparing the defense robustness of MAPE and baseline methods. To encompass a wide range of attack sources and achieve more realistic performance evaluations, our selected models include both

Table 1 Classification accuracy rates (%) of SAPE and MAPE in defending against adversarial attacks on CIFAR-10, CIFAR-100 and Mini-ImageNet (*higher is better*)

Datasets	Defenses	Clean	ResNet-34				ShuffleNet-V2-2×			
			FGSM	BIM	DIM	PGD	FGSM	BIM	DIM	PGD
CIFAR-10	Natural Training	**95.82**	27.60	0.11	0.05	0.02	51.24	9.90	6.23	7.16
	SAPE	95.23	**95.28**	95.14	**95.08**	94.15	91.99	91.86	92.13	91.78
	SAPE & Random	95.02	93.66	93.90	93.65	92.90	93.43	93.06	93.21	92.90
	SAPE & PPSA (MAPE)	95.37	95.22	**95.42**	94.98	**94.36**	**95.28**	**94.76**	**94.44**	**94.66**
CIFAR-100	Natural Training	**75.30**	16.43	1.28	1.03	0.88	24.42	6.67	3.98	4.47
	SAPE	74.82	**74.48**	72.75	72.34	**72.25**	70.96	67.27	66.63	66.59
	SAPE & Random	74.54	73.31	71.49	71.36	71.18	72.01	69.65	69.32	69.25
	SAPE & PPSA (MAPE)	74.99	74.23	**72.93**	**72.57**	72.14	**74.65**	**70.64**	**70.76**	**71.40**
Mini- ImageNet	Natural Training	**76.37**	11.64	0.03	0.00	0.01	17.63	1.69	1.02	1.26
	SAPE	74.62	**72.71**	71.18	70.88	**71.29**	65.95	64.78	63.03	63.64
	SAPE & Random	74.02	71.08	69.92	69.76	69.92	68.83	67.82	65.52	67.57
	SAPE & PPSA (MAPE)	74.86	72.64	**71.45**	**71.09**	71.20	**69.13**	**70.28**	**69.85**	**70.35**

ResNet-34 serves as the target model. Simultaneously, it and ShuffleNet-V2-2× serve as substitute models for launching the attacks. For each attack, we show the most successful defense with bold

classic and cutting-edge models, spanning from large-scale to lightweight designs.

Baseline defense approaches. We compare with natural training, adversarial training [19] and the following input transformation defense methods: TVM [34], feature denoising [27], pixel deflection [35], mixup inference [23], HGD [26] and LDT [25]. Except for natural training, all defense methods incorporate adversarial training to enhance their defense performance. To enhance the persuasiveness of the experiments, we set the target model as the ResNet series, similar to other defense methods. Considering the balance between classification performance and computational cost, we decide to use ResNet-34 as the target model of all defense methods. Feature denoising employs a ResNet-34 model with denoising blocks as its target model.

Training details. During the training process of MAPE, attack methods DIM [11] and PGD [19] are used to generate adversarial examples. The perturbation budget is within the range of (4/255, 12/255). The step size is set to 2/255, while the number of steps is set to 20. The training data is a mixture of adversarial examples and clean examples. The defense model $\mathcal{E}$ is optimized using Adam. Their initial learning rate η and weight decay λ are set to 0.01 and 0. The number of training epochs is set to 120, with the learning rate decreasing by a factor of 10 at the 50th, 75th, and 100th epochs. The batch size of the dataset is set to 128. The GPU device used is a NVIDIA Tesla A100 (40GB).

Existence of MAPE

In this section, we conduct a series of controlled experiments on the CIFAR-10, CIFAR-100, and Mini-ImageNet datasets, to validate the effectiveness of MAPE and its components (SAPE and PPSA) in adversarial defense. Each set of controlled experiments uses ResNet-34 as the target model under attack, while both ResNet-34 and unforeseen ShuffleNet-V2-2× serve as substitute models for launching the attacks. The methods compared included natural training, SAPE, SAPE driven by randomly selected pre-trained models (SAPE & Random), and SAPE driven by pre-trained models scheduled with PPSA (SAPE & PPSA, namely MAPE). The evaluation results are presented in Table 1.

It can be observed that, compared to natural training, SAPE has significantly improved adversarial defense performance. However, since the defense model is trained solely on adversarial examples generated by the target model ResNet-34, the effectiveness of SAPE in defending against adversarial attacks from the unforeseen substitute model ShuffleNet-V2-2× significantly decreases. On CIFAR-10, CIFAR-100, and Mini-ImageNet, the average defensive performance of SAPE against ShuffleNet-V2-2× as a substitute model is lower than that against ResNet-34 as a substitute model by 2.97, 5.09, and 7.16%, respectively. This indicates that the defense model trained solely with SAPE exhibits weak generalization capability.

In contrast, when randomly selecting pre-trained models to drive SAPE (SAPE & Random) for training the defense model, it shows relatively close defense performance against attacks from different substitute models. The corresponding data for the aforementioned metrics are 0.38, 1.78, and 2.74%, respectively. When using the PPSA to strategically schedule the pre-trained models to drive SAPE (SAPE & PPSA, namely MAPE), the defense model exhibits stronger generalization capabilities, with the aforementioned metrics

Table 2 Classification accuracy rates (%) of different methods in defending against unknown types of adversarial attacks on CIFAR-10, CIFAR-100 and Mini-ImageNet (*higher is better*)

Dataset	Defenses	Clean	ResNet-34				ShuffleNet-V2-2×				ResNet-V2-50			
			FGSM	BIM	UPGD	VNIM	FGSM	BIM	UPGD	VNIM	FGSM	BIM	UPGD	VNIM
CIFAR-10	Nat. Tra	**95.82**	27.60	0.11	0.07	0.04	51.24	9.90	6.18	2.93	46.33	8.78	4.97	2.44
	Adv. Tra	88.48	86.94	88.25	87.36	86.42	85.42	86.17	85.62	84.86	85.64	86.23	85.69	85.22
	TVM	89.32	88.05	88.24	87.67	87.04	85.99	86.41	85.95	85.38	86.59	86.88	86.64	86.01
	Feat. Den	88.96	87.47	88.35	87.88	87.12	85.53	86.54	85.79	85.42	86.05	86.94	86.40	86.39
	Pix. Def	90.44	89.61	91.20	90.73	91.27	87.82	89.56	89.21	89.36	88.75	90.33	90.17	90.46
	Mix. Inf	95.30	91.69	94.51	93.33	93.39	88.84	93.72	91.30	91.81	90.55	94.18	92.38	92.43
	HGD	93.49	93.50	93.94	93.96	94.15	91.86	92.07	91.91	93.22	92.21	92.04	93.31	93.69
	LDT	95.31	95.07	95.14	**95.17**	93.69	93.29	93.23	93.19	92.13	93.58	93.85	93.76	93.15
	MAPE	95.37	**95.22**	**95.42**	94.23	**94.48**	**95.28**	**94.76**	**94.95**	**95.18**	**95.19**	**95.29**	**94.97**	**95.02**
CIFAR-100	Nat. Tra	**75.30**	16.43	1.28	0.86	0.73	24.42	6.67	4.29	2.87	29.10	17.49	13.54	7.38
	Adv. Tra	63.88	63.30	63.35	62.50	62.07	59.76	60.05	59.15	58.48	61.31	61.75	61.18	60.64
	TVM	65.43	62.53	62.42	61.01	60.84	59.00	59.13	57.90	57.41	61.91	62.53	61.59	60.94
	Feat. Den	62.07	59.94	60.58	58.94	58.46	56.74	57.04	55.80	55.16	59.56	59.88	59.04	59.13
	Pix. Def	66.83	63.98	65.49	64.62	64.67	60.56	62.12	61.09	61.39	64.08	64.74	64.26	63.91
	Mix. Inf	73.67	65.06	70.98	68.85	68.76	61.94	67.61	65.83	65.72	66.67	69.10	68.02	67.50
	HGD	69.08	71.30	69.45	68.34	69.91	68.28	65.93	65.05	66.50	72.85	69.05	68.36	70.45
	LDT	74.78	**74.59**	72.55	71.41	**72.20**	71.14	67.07	66.30	67.57	72.54	68.66	69.57	70.60
	MAPE	74.99	74.23	**72.93**	**71.80**	72.15	**74.65**	**70.64**	**70.65**	**71.75**	**74.33**	**71.76**	**72.05**	**73.04**
Mini-ImageNet	Nat. Tra	**76.37**	11.64	0.03	0.01	0.00	17.63	1.69	1.18	0.44	17.08	4.67	2.93	0.88
	Adv. Tra	58.39	61.19	61.86	60.93	60.91	56.13	56.28	55.85	55.59	57.05	57.28	57.10	56.99
	TVM	66.21	64.78	66.67	64.07	64.70	59.29	61.28	58.97	59.45	62.88	64.00	62.63	63.33
	Feat. Den	59.00	58.50	58.62	56.02	55.77	53.01	53.14	50.79	50.38	57.77	57.81	57.10	57.41
	Pix. Def	67.28	65.17	67.28	65.43	65.14	59.74	62.01	59.93	59.76	64.54	65.40	64.40	64.48
	Mix. Inf	73.81	68.50	70.00	69.23	69.26	63.41	64.74	63.75	64.08	67.67	69.17	68.58	69.09
	HGD	72.41	72.19	69.68	67.12	69.78	66.79	64.18	62.08	64.43	70.21	67.65	68.63	68.66
	LDT	73.76	70.56	71.19	69.45	**70.69**	63.25	65.14	62.21	63.68	69.25	67.32	66.08	69.39
	MAPE	74.86	**72.64**	**71.45**	**70.14**	70.16	**69.13**	**70.28**	**68.80**	**69.92**	**75.29**	**71.32**	**71.80**	**72.58**

ResNet-34 serves as the target model under attack, while it, along with ShuffleNet-V2-2× and ResNet-V2-50, acts as substitute models for launching the attacks. For each attack, we show the most successful defense with bold and the second one with underline

being 0.21, 1.10, and 1.69%, respectively. Additionally, due to the dynamic adjustment of the negative momentum mechanism, its defense performance has also been effectively enhanced. On CIFAR-10, CIFAR-100, and Mini-ImageNet, SAPE & PPSA (MAPE) outperforms SAPE & Random in average defense effectiveness by 1.55, 1.47, and 1.95%, respectively. These evaluation results indicate that, compared to SAPE, MAPE can effectively enhance both the defense effectiveness and generalization capability of the defense model.

Defending against unknown types of transferable attacks

We evaluate the effectiveness of various defense methods against unknown types of adversarial attacks. These attack methods are not used in the training process of the defense methods. They include the one-step method FGSM, the multi-step method BIM, as well as advanced transferable attack methods such as Ultimate PGD (UPGD) and VNIM. The adversarial perturbation budget is $L_\infty = 8/255$, and the number of steps is set to 50 for iterative attack methods. ResNet-34 serves as the target model under attack, while it, along with ShuffleNet-V2-2× and ResNet-V2-50, acts as substitute models for launching the attacks. The detailed evaluation results on CIFAR-10, CIFAR-100, and Mini-ImageNet are presented in Table 2. For transformer-based models, the selected substitute model is ViT-S/16. The detailed evaluation results on Mini-ImageNet are shown in Table 3. ShuffleNet-V2-2×, ResNet-V2-50, and ViT-S/16 are not used in the training process of any defense methods.

In Table 2, it can be found that regardless of the defense strategy employed, there will be a reduction in the target model's classification accuracy on clean examples. In comparison, MAPE shows the least degradation of clean classification accuracy. When faced with unknown types of adversarial attacks, MAPE consistently demonstrates superior defensive performance compared to other defense methods, always exhibiting optimal performance against each type of attack. Its average defense performance on CIFAR-10, CIFAR-100, and Mini-ImageNet is 95.03, 72.69, and 71.41%, respectively −1.14, 2.01, and 3.49% higher than LDT. Furthermore, it can be observed that compared to defending against known substitute models (ResNet-34), all defense methods except for MAPE exhibit a significant performance decline when defending against unforeseen substitute models (ShuffleNet-V2-2× and ResNet-V2-50). Taking the Mini-ImageNet dataset as an example, the average classification accuracies of HGD, LDT, and MAPE when defending against attacks from ResNet-34 are relatively similar, at 69.69, 70.47, and 71.10%, respectively. However, when defending against attacks from ShuffleNet-V2-2×, their average classification accuracies drop to 64.37, 63.57, and 69.53%, representing declines of 5.32, 6.9, and 1.57%, respectively. This indicates that, compared to other defense methods, MAPE exhibits higher generalization capabilities, effectively defending against transferable adversarial attacks from unforeseen substitute models.

Table 3 Classification accuracy rates (%) in defending against unknown types of adversarial attacks on Mini-ImageNet (*higher is better*)

Defenses	Clean	FGSM	BIM	UPGD	VNIM
Nat. Tra	**76.37**	42.49	39.69	34.06	26.77
Adv. Tra	58.39	56.71	57.07	56.59	56.15
TVM	66.22	60.61	62.90	61.12	59.33
Feat. Den	59.00	54.13	56.23	54.63	53.33
Pix. Def	67.28	61.29	<u>63.50</u>	62.08	59.53
Mix. Inf	73.81	<u>61.89</u>	62.52	61.89	61.04
HGD	72.41	60.73	62.25	62.31	<u>61.20</u>
LDT	73.76	61.76	61.97	<u>62.65</u>	60.27
MAPE	<u>74.86</u>	**64.58**	**65.43**	**65.88**	**64.26**

ViT-S/16 and ResNet-34 serve as the substitute model and target model, respectively. For each attack, we show the most successful defense with bold and the second one with underline

In Table 3, the substitute model ViT-S/16 is based on a transformer architecture, while the target model ResNet-34 relies on convolutional structures. This results in substantial structural differences between them, leading to significant distinctions in their classification boundaries on Mini-ImageNet. Consequently, the effectiveness of adversarial attacks on ViT-S/16 cannot be readily transferred to ResNet-34. Specifically, the natural accuracy drop after experiencing adversarial attacks in Table 3 is not as pronounced as that in Table 2. Similarly, because all defense methods were trained using CNNs as hypothetical substitute models, the effectiveness of defending against adversarial attacks from ViT-S/16 is not as strong as defending against attacks from CNNs. This leads to a curious phenomenon as shown in Table 3: the adversarial attacks from ViT-S/16 are not very strong, yet the defensive effectiveness against them is also not very high. Nonetheless, MAPE still demonstrates the strongest defensive capabilities compared to other methods.

Defending against integrated transferable attacks

We evaluated the effectiveness of different defense methods against integrated adversarial attacks. The integrated adversarial attack methods include LGV [31], TAIG [32], and AdaEA [33]. The basic attack method, adversarial perturbation budget, and number of steps are set to BIM, $L_\infty = 8/255$, and 50, respectively. For LGV, the substitute model is ShuffleNet-V2-2× and the number of weight sets is 10. For TAIG, the substitute model is also ShuffleNet-V2-2×, and the example augmentation factor is 20. For AdaEA, the substitute

Table 4 Classification accuracy rates (%) in defending against integrated adversarial attacks on Mini-ImageNet (*higher is better*)

Defenses	Clean	LGV	TAIG	AdaEA
Nat. Tra	**76.37**	7.03	0.37	0.98
Adv. Tra	58.39	57.14	55.25	56.22
TVM	66.13	62.78	58.99	59.98
Feat. Den	59.00	55.52	50.18	53.66
Pix. Def	67.28	64.28	59.68	60.79
Mix. Inf	73.81	66.08	64.23	61.73
HGD	72.41	66.72	62.58	60.67
LDT	73.76	67.40	64.10	63.41
MAPE	74.86	**69.43**	**69.12**	**69.78**

ResNet-34 serve as the target model under attack. For each attack, we show the most successful defense with bold and the second one with underline

model ensemble consists of ShuffleNet-V2-2×, ResNet-V2-50, and VGG-19. For all defense methods, these substitute models have never been encountered. The detailed evaluation results on Mini-ImageNet are shown in Table 4. It is evident that when confronted with ensemble attacks involving multiple gradients or models, MAPE continues to exhibit the highest natural accuracy and defensive performance compared to other defense methods.

Defending against strong substitute model attacks and adaptive attacks

In this section, we consider the defensive effect of the proposed method in gray-box and white-box attack environments. In a gray-box attack environment, the attackers employ strong substitute model attacks. In a white-box attack environment, attackers use adaptive attacks. The conditions for carrying out white-box attacks are more rigorous than those for gray-box attacks. A detailed overview is provided in "Experimental setup" section.

We evaluated the effectiveness of different defense methods against strong substitute model attacks and adaptive attacks. The base target model under attack is ResNet-34. The adversarial attack methods include FGSM, BIM, UPGD, VMIM, and VNIM. The adversarial perturbation budget is $L_\infty = 8/255$, and the number of steps is set to 50 for iterative attack methods. For defense methods based on obfuscated gradients or random transformation, we respectively employ backward pass differentiable approximation (BPDA) [54] and expectation over transformation (EOT) [54] for gradient correction. The detailed evaluation results on Mini-ImageNet are presented in Table 5.

From Table 5, it is evident that in strong substitute model attacks under gray-box environments, MAPE demonstrates superior defense effectiveness compared to other methods. This is because, during the training process of MAPE, multiple classification models are introduced to aid in training, alongside random adjustments to adversarial example generation methods and perturbation budgets. Consequently, among all defense approaches, MAPE exhibits the largest sample space and parameter space, significantly reducing the similarity in parameter weights between strong substitute models and MAPE. However, in adaptive attacks under white-box environments, where all model weights and defense strategies are exposed, attackers can develop precise attack strategies, resulting in a significant decrease in the defensive capabilities of all methods. Although adversarial training and TVM can defend against a small number of adversarial examples, their natural accuracy and black-box accuracy are far inferior to those of MAPE.

Failure of defense against adaptive attacks is not a unique issue of the proposed method but rather a common problem associated with black-box defense methods. Such failures can also be found in the original papers on HGD [26] and LDT [25]. Although this paper primarily focuses on the domain of black-box defenses, integrating the proposed method with several advanced adversarial training techniques can still enhance its applicability in the field of white-box defenses. For instance, employing classical TRADES [20] and MART [55] loss functions in the adversarial training process of the proposed method can optimize the classification boundaries of the model. Utilizing data generated by the latest elucidating diffusion model (EDM) as training data [56] can simultaneously improve the performance of the proposed method in both black-box and white-box defenses. Additionally, a specific scaling law [57] allows for more rational allocation of resources such as model and dataset sizes, thereby maximizing adversarial robustness given a fixed computational capacity. Furthermore, combining the fast adversarial training method known as FGSM-PCO [58] can reduce the computational costs associated with the adversarial training process. In summary, integrating adversarial training methods aids in improving the white-box robustness of MAPE, potentially broadening the applicability of the proposed method. This will be our primary research direction moving forward.

Further evaluations

Cross-model defense

We evaluated the transferability of the defensive capabilities across different methods, i.e., their defensive effects on different target models. The adversarial attack methods include FGSM, BIM, UPGD, VMIM, and VNIM. The adversarial perturbation budget is $L_\infty = 8/255$, and the number of steps is set to 50 for iterative attack methods. The selected substi-

Table 5 Classification accuracy rates (%) in defending against strong substitute model attacks and adaptive attacks on Mini-ImageNet (*higher is better*)

Defenses	Clean	Strong substitute model attacks					Adaptive attacks				
		FGSM	BIM	UPGD	VMIM	VNIM	FGSM	BIM	UPGD	VMIM	VNIM
Nat. Tra	**76.37**	13.40	3.39	2.03	1.19	1.07	11.69	0.03	0.03	0.01	0.01
Adv. Tra	58.39	32.27	29.22	29.42	29.87	29.04	21.37	**15.42**	**16.26**	**16.14**	**16.33**
TVM	66.17	50.85	45.62	46.78	40.49	39.48	18.84	4.49	5.16	6.77	7.38
Feat. Den	59.00	21.45	12.17	11.52	11.39	10.57	6.75	0.23	0.31	0.33	0.31
Pix. Def	67.28	41.04	35.88	34.71	34.37	33.67	15.63	0.57	0.73	0.76	0.78
Mix. Inf	73.81	42.73	37.70	35.48	32.45	29.98	**29.46**	0.00	0.00	0.01	0.00
HGD	72.41	51.72	40.81	38.77	35.78	33.38	13.93	0.05	0.06	0.46	0.52
LDT	73.76	55.32	46.77	46.87	44.80	41.58	19.97	0.06	0.10	0.79	1.48
MAPE	74.86	**63.62**	**54.42**	**55.89**	**53.53**	**51.29**	21.98	0.11	0.12	1.68	2.34

For each attack, we show the most successful defense with bold and the second one with underline

Table 6 Classification accuracy rates (%) of different methods for different target models in defending against unknown types of adversarial attacks on CIFAR-10, CIFAR-100 and Mini-ImageNet (*higher is better*)

Dataset	Defenses	Clean	ShuffleNet-V2-2×					ResNet-V2-50				
			FGSM	BIM	UPGD	VMIM	VNIM	FGSM	BIM	UPGD	VMIM	VNIM
VGG-19 serves as the target model under attack												
CIFAR-10	Nat. Tra	**94.69**	51.11	18.55	13.14	5.67	5.96	48.31	22.41	15.90	7.54	7.21
	HGD	90.46	89.75	90.19	89.79	90.06	90.22	92.05	91.35	91.13	91.53	91.64
	LDT	93.03	91.32	90.67	90.56	90.77	90.74	92.21	91.69	91.87	92.56	92.69
	MAPE	94.31	**94.10**	**93.74**	**94.18**	**94.11**	**93.89**	**93.94**	**94.02**	**94.13**	**94.29**	**93.84**
CIFAR-100	Nat. Tra	**75.16**	26.77	14.22	10.57	6.93	6.75	30.13	23.97	19.24	11.26	10.45
	HGD	67.38	63.53	61.92	60.73	62.16	62.59	65.20	62.47	60.82	62.71	63.01
	LDT	71.43	67.48	62.15	60.70	63.14	62.91	69.48	63.11	63.50	64.73	65.11
	MAPE	74.48	**72.82**	**69.08**	**69.81**	**70.31**	**70.28**	**73.89**	**69.71**	**70.92**	**72.24**	**72.46**
Mini-ImageNet	Nat. Tra	**70.33**	21.96	11.10	8.41	4.02	3.65	22.49	29.37	24.43	12.56	10.63
	HGD	62.83	57.73	58.88	56.36	57.37	58.09	61.93	61.00	58.83	60.92	61.32
	LDT	65.82	63.97	59.64	58.25	59.58	59.88	66.71	62.91	63.32	64.93	65.11
	MAPE	69.27	**64.59**	**65.19**	**64.73**	**65.11**	**64.99**	**68.44**	**66.47**	**66.68**	**67.52**	**67.15**
ViT-S/16 serves as the target model under attack												
CIFAR-10	Nat. Tra	**98.67**	85.30	82.77	78.26	67.31	68.07	88.01	85.03	81.21	73.14	73.58
	HGD	95.20	94.07	93.76	93.52	93.48	93.43	95.14	94.30	94.15	94.71	94.32
	LDT	97.22	96.04	95.60	95.41	95.42	95.53	96.89	95.80	95.79	96.04	96.24
	MAPE	98.53	**97.06**	**96.19**	**96.27**	**96.35**	**96.55**	**97.47**	**96.87**	**96.72**	**96.85**	**96.74**
CIFAR-100	Nat. Tra	**90.32**	64.03	60.51	55.15	49.08	49.35	66.84	66.59	61.31	52.70	53.51
	HGD	78.94	74.99	73.97	72.57	73.43	73.47	74.73	72.64	71.34	72.08	72.96
	LDT	82.46	80.01	74.75	74.23	75.76	76.26	79.77	73.77	74.39	76.10	76.12
	MAPE	89.80	**85.94**	**82.20**	**82.47**	**83.34**	**83.63**	**87.76**	**83.13**	**83.82**	**85.75**	**85.53**
Mini- ImageNet	Nat. Tra	**90.53**	68.77	69.18	63.00	52.30	50.86	73.45	76.28	71.03	59.88	60.38
	HGD	84.36	80.04	79.74	78.03	79.57	79.89	82.16	81.55	80.11	81.06	81.43
	LDT	87.19	84.97	83.32	82.42	83.39	83.68	86.80	84.33	84.18	85.30	85.58
	MAPE	89.38	**86.45**	**85.73**	**85.47**	**86.04**	**86.07**	**88.09**	**86.64**	**86.68**	**87.45**	**87.43**

ShuffleNet-V2-2× and ResNet-V2-50 serve as substitute models for generating adversarial examples, VGG-19 and ViT-S/16 serve as other target models not included in the framework. For each attack, we show the most successful defense with bold

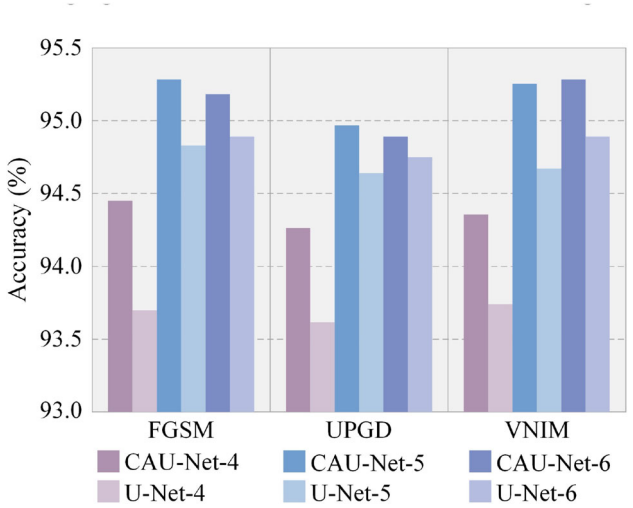

Fig. 5 Classification accuracy rates (%) of MAPEs with different defense models in defending against unknown types of adversarial attacks on CIFAR-10 (*higher is better*). ShuffleNet-V2-2× and ResNet-34 serve as the substitute model and target model, respectively. CA is channel-attention mechanism and numerical suffixes is the quantity of submodules

tute models are ShuffleNet-V2-2× and ResNet-V2-50. We only choose to compare HGD and LDT with MAPE because their defense and target models can be separated, and the target model is independently trained (not jointly trained with the defense model). Therefore, when the original target model is replaced with different target models VGG-19 and ViT-S/16, they can still function properly. At this point, their defensive effects depend on the robustness and generalization of the defense model. The detailed evaluation results on CIFAR-10, CIFAR-100, and Mini-ImageNet are presented in Table 6.

From Table 6, it can be observed that the natural accuracy of MAPE closely aligns with the natural accuracy of the target model. This suggests that MAPE rarely leads to misclassification of input examples by the target model. When compared to HGD and LDT, MAPE achieves the best defensive performance under each type of attack, whether assisting the CNN model VGG-19 or the transformer model ViT-S/16. This indicates that MAPE is capable of training a defense model with strong generalization capabilities. This defense model is independent of the target model, and its performance is not affected by it. After training, it can provide adversarial defense for different target models without the need for retraining for each specific target model.

Ablation study

Based on whether channel-attention mechanism layers have been added, we divide the defense models into CAU-Net and U-Net. The former incorporates channel-attention mechanism layers, while the latter does not. Subsequently, we assign 4, 5, and 6 submodules to each, resulting in a total of 6 defense models participating in the ablation study. The defense model utilized in this paper is CAU-Net-5, which features channel-attention mechanism layers and 5 submodules. Each defense model is trained in a uniform manner and then subjected to adversarial attacks including FGSM, UPGD, and VNIM. The adversarial perturbation budget is $L_\infty = 8/255$, and the number of steps is set to 50. ShuffleNet-V2-2× serves as the substitute model for generating adversarial examples, and ResNet-34 serves as the target model under attack.

Figure 5 presents the ablation study of the defense models discussed above on CIFAR-10. It can be observed that when the number of submodules (network depth) is the same, CAU-Net with channel-attention mechanism layers exhibits stronger robustness and better performance in defending against adversarial attacks compared to U-Net. The relatively shallow network depth of CAU-Net-4 results in a diminished fitting capability and a lower defense effectiveness. Conversely, excessive network depth in CAU-Net-6 may lead to overfitting issues, resulting in a decline in its defense performance. Therefore, among the three models, CAU-Net-5 exhibits a more suitable fitting capability and the strongest overall defense performance. Additionally, with only 1.66M parameters, CAU-Net-4 comprises merely 24.85% of CAU-Net-5's parameter count (6.68M). Given a modest compromise in defense performance, CAU-Net-4 can be employed for adversarial defense in lightweight classification models.

Robustness against perturbation budgets

We set the L_∞ norm perturbation budget within the range of [4/255, 24/255] and employ FGSM, UPGD, and VNIM to attack the target model, in order to evaluate the robustness of MAPE against perturbation budgets. The detailed evaluation results are shown in Fig. 6. It can be observed that when the adversarial perturbation budget is less than 12/255, LDT is able to maintain a defense effectiveness of over 80%, but as the perturbation budget increases, the defense effectiveness sharply decreases. In comparison, it is only when the adversarial perturbation budget surpasses 16/255 that MAPE's defense effectiveness shows a significant decrease. Furthermore, at an adversarial perturbation budget of 4/255, MAPE exhibits an average defense effectiveness 3.1% higher than LDT. However, at a perturbation budget of 24/255, MAPE's average defense effectiveness surpasses LDT by a remarkable 105.9%. This indicates that the defense based on MAPE exhibits strong robustness against high adversarial perturbation budgets and powerful adversarial attacks.

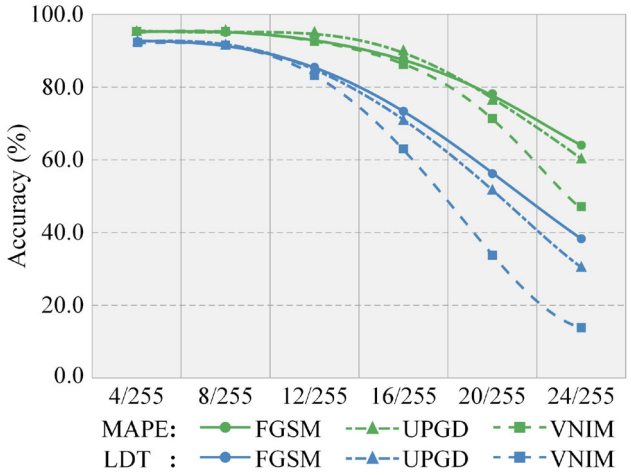

Fig. 6 Classification accuracy rates (%) of MAPE and LDT in defending against unknown types of adversarial attacks with different perturbation budgets on CIFAR-10 (*higher is better*). ShuffleNet-V2-2× and ResNet-34 serve as the substitute model and target model, respectively

Defense costs

We compared the training and evaluation costs of different defense methods on Mini-ImageNet, as shown in Fig. 7. The evaluation cost refers to the actual operational cost. The comparison metrics include memory usage and running time. All defense methods employed a pre-trained ResNet-34 as the target model. When measuring the training cost, the batch size of the dataset was set to 128. For evaluating the cost, defense methods processed a single image at a time, running a total of 10,000 iterations, with the average taken as the processing cost for one image.

During the training process, methods incorporating deep defense models, such as HGD, LDT, and proposed MAPE, require higher training costs compared to others. HGD employs the target model to predict both complete clean examples and adversarial examples separately, so it has the longest running time. In contrast, LDT and MAPE only require predictions for mixed examples composed of clean and adversarial examples, so their running times are considerably shorter than those of HGD. MAPE appears to consume a substantial amount of memory when contrasted with HGD and LDT. However, due to the memory reuse mechanism in PyTorch, the memory usage of MAPE is not the sum of the memory usage of the pre-trained models, but rather their upper bound. This implies that certain heavyweight pre-trained models, such as PyramidNet and WideResNet, are responsible for the increased memory usage. In practical applications, if memory is constrained, these heavier models can be replaced with lighter alternatives.

During the evaluation process, HGD, LDT, and MAPE still incur higher defense costs compared to other methods. However, due to the utilization of a more efficient CAU-Net in MAPE, its memory usage is 2.3% (18 MiB) lower than that of HGD and 4.9% (40 MiB) lower than that of LDT. In terms of runtime for a single image, MAPE takes 9.0% (0.87 ms) less time than LDT, while it takes 7.4% (0.61 ms) more time than HGD. Furthermore, the model parameter counts for HGD, LDT, and MAPE are 32.36M, 33.44M, and 28.01M, respectively, while the parameter counts for other methods are approximately 21.3M. The model parameter count of MAPE is 13.4% (4.35M) lower than that of HGD and 16.3% (5.43M) lower than that of LDT.

In summary, compared to HGD and LDT, MAPE exhibits the lowest memory usage during evaluation while demonstrating the highest memory usage during training. When memory is constrained, the latter issue can be addressed by substituting lighter pre-trained models. In terms of running time, MAPE consistently maintains an intermediate level of performance. Additionally, MAPE has the lowest model parameter count. Combining these findings with previous

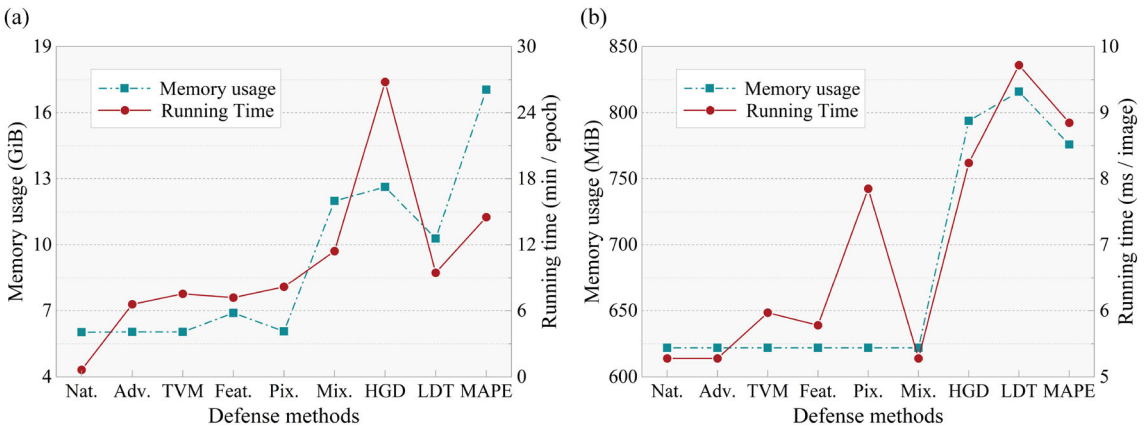

Fig. 7 Comparison of different defense methods in terms of **a** training cost and **b** evaluation cost. The comparison metrics include the memory usage and the running time. The target model and dataset used are ResNet-34 and Mini-ImageNet, respectively

defense experiment results, it is evident that MAPE not only possesses the strongest defense performance but also incurs a defense cost comparable to other methods of similar type.

Conclusion

In this paper, our approach is to deploy a defense model external to the target model to extract and eliminate the adversarial perturbations from input examples. The optimization objective focuses on enhancing the robustness and generalization of the used defense model to effectively defend against a variety of unknown types of adversarial attacks. To achieve this goal, we propose a deep learning defense known as MAPE, which is primarily composed of SAPE and PPSA. SAPE utilizes CAU-Net as its defense model, training it to eliminate adversarial perturbations by using adversarial examples generated from a pre-trained model. Meanwhile, PPSA integrates model difference probability and negative momentum probability to strategically schedule multiple pre-trained models, maximizing the differences among these models during adjacent training cycles, thus enhancing the diversity of the generated adversarial examples. The evaluation results demonstrate that MAPE exhibits strong robustness and can effectively defend against various types of adversarial attacks in a black-box environment. In future work, potential extensions include strengthening its defense capabilities against strong substitute model attacks and adaptive attacks, as well as applying it to adversarial defense in object detection, image segmentation, and other visual tasks.

Acknowledgements This work was supported by the National Key Research and Development Program for Young Scientists of China (No. 2022YFB3102800), and Major Science and Technology Project of Henan Province (No. 221100240100).

Author Contributions Xinlei Liu: Conceptualization, Methodology, Software, Validation, Formal analysis, Data Curation, Writing-Original Draft, Writing- Review & Editing. Jichao Xie: Methodology, Software, Validation, Data Curation, Investigation, Writing-Original Draft. Tao Hu: Resources, Writing-Review & Editing, Supervision, Project administration, Funding acquisition. Peng Yi: Resources, Supervision, Funding acquisition. Yuxiang Hu: Data Curation, Validation, Writing-Review & Editing. Shumin Huo: Formal analysis, Investigation, Writing-Review & Editing. Zhen Zhang: Formal analysis, Project administration.

Data Availibility Data will be made available on request.

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article

References

1. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539
2. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. https://doi.org/10.1109/CVPR.2016.90
3. Guo J, Han K, Wang Y, Wu H, Chen X, Xu C, Xu C (2021) Distilling object detectors via decoupled features. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2154–2164. https://doi.org/10.1109/CVPR46437.2021.00219
4. Siam M, Gamal M, Abdel-Razek M, Yogamani S, Jagersand M, Zhang H (2018) A comparative study of real-time semantic segmentation for autonomous driving. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 700–710. https://doi.org/10.1109/CVPRW.2018.00101
5. Zhou Y, Han M, Liu L, He J, Gao X (2019) The adversarial attacks threats on computer vision: A survey. In: 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW), pp. 25–30. https://doi.org/10.1109/MASSW.2019.00012
6. Akhtar N, Mian A, Kardan N, Shah M (2021) Threat of adversarial attacks on deep learning in computer vision: Survey II. CoRR arXiv:2108.00401
7. Gao H, Yang X, Hu Y, Liang Z, Xu H, Wang B, Mu H, Wang Y (2024) Adversarial sample attacks algorithm based on cycle-consistent generative networks. Appl Soft Comput 162:111778. https://doi.org/10.1016/j.asoc.2024.111778
8. Li Q, Wang Z, Zhang X, Li Y (2024) Attack-cosm: attacking the camouflaged object segmentation model through digital world adversarial examples. Complex Intell Syst 10:5445–5457. https://doi.org/10.1007/s40747-024-01455-7
9. Kurakin A, Goodfellow IJ, Bengio S (2017) Adversarial examples in the physical world. In: 2017 International Conference on Learning Representations (ICLR), OpenReview.net
10. Croce F, Hein M (2020) Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: 2020 International Conference on Machine Learning (ICML), PMLR. pp. 2206–2216
11. Xie C, Zhang Z, Zhou Y, Bai S, Wang J, Ren Z, Yuille AL (2019) Improving transferability of adversarial examples with input diversity. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2725–2734. https://doi.org/10.1109/CVPR.2019.00284
12. Lin J, Song C, He K, Wang L, Hopcroft JE (2020) Nesterov accelerated gradient and scale invariance for adversarial attacks. In: 2020 International Conference on Learning Representations (ICLR), OpenReview.net
13. Wang X, He K (2021) Enhancing the transferability of adversarial attacks through variance tuning. In: 2021 IEEE/CVF Conference

on Computer Vision and Pattern Recognition (CVPR), pp. 1924–1933. https://doi.org/10.1109/CVPR46437.2021.00196
14. Zhu H, Ren Y, Liu C, Sui X, Zhang L (2024) Frequency-based methods for improving the imperceptibility and transferability of adversarial examples. Appl Soft Comput 150:111088. https://doi.org/10.1016/j.asoc.2023.111088
15. Chen Z, Luo W, Naseem ML, Kong L, Yang X (2024) Comprehensive comparisons of gradient-based multi-label adversarial attacks. Complex Intell Syst 10:6667–6692. https://doi.org/10.1007/s40747-024-01506-z
16. Bhagoji AN, He W, Li B, Song D (2018) Practical black-box attacks on deep neural networks using efficient query mechanisms. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y (eds), Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part XII, Springer. pp. 158–174. https://doi.org/10.1007/978-3-030-01258-8_10
17. Li H, Xu X, Zhang X, Yang S, Li B (2020) QEBA: query-efficient boundary-based blackbox attack. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020, Computer Vision Foundation / IEEE. pp. 1218–1227. https://openaccess.thecvf.com/content_CVPR_2020/html/Li_QEBA_Query-Efficient_Boundary-Based_Blackbox_Attack_CVPR_2020_paper.html, https://doi.org/10.1109/CVPR42600.2020.00130
18. Shi Y, Han Y, Hu Q, Yang Y, Tian Q (2023) Query-efficient black-box adversarial attack with customized iteration and sampling. IEEE Trans Pattern Anal Mach Intell 45:2226–2245. https://doi.org/10.1109/TPAMI.2022.3169802
19. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A (2018) Towards deep learning models resistant to adversarial attacks. In: 2018 International Conference on Learning Representations (ICLR), OpenReview.net
20. Zhang H, Yu Y, Jiao J, Xing EP, Ghaoui LE, Jordan MI (2019) Theoretically principled trade-off between robustness and accuracy. In: Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA, PMLR. pp. 7472–7482
21. Ozdag M (2018) Adversarial attacks and defenses against deep neural networks: a survey. Procedia Comput Sci 140:152–161. https://doi.org/10.1016/j.procs.2018.10.315
22. Raff E, Sylvester J, Forsyth S, McLean M (2019) Barrage of random transforms for adversarially robust defense. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6521–6530. https://doi.org/10.1109/CVPR.2019.00669
23. Pang T, Xu K, Zhu J (2020) Mixup inference: Better exploiting mixup to defend adversarial attacks. In: 2020 International Conference on Learning Representations (ICLR), OpenReview.net
24. Bahat Y, Irani M, Shakhnarovich G (2019) Natural and adversarial error detection using invariance to image transformations. CoRR arXiv:1902.00236
25. Li J, Zhang S, Cao J, Tan M (2023) Learning defense transformations for counterattacking adversarial examples. Neural Netw 164:177–185. https://doi.org/10.1016/j.neunet.2023.03.008
26. Liao F, Liang M, Dong Y, Pang T, Hu X, Zhu J (2018) Defense against adversarial attacks using high-level representation guided denoiser. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1778–1787. https://doi.org/10.1109/CVPR.2018.00191
27. Xie C, Wu Y, Maaten Lvd, Yuille AL, He K (2019) Feature denoising for improving adversarial robustness. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 501–509. https://doi.org/10.1109/CVPR.2019.00059
28. Goodfellow IJ, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples. In: Bengio Y, LeCun Y (eds.), 2015 International Conference on Learning Representations (ICLR)
29. Guo Y, Li Q, Chen H (2020) Backpropagating linearly improves transferability of adversarial examples. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H (eds.), 2020 Neural Information Processing Systems (NeurIPS)
30. Dong Y, Liao F, Pang T, Su H, Zhu J, Hu X, Li J (2018) Boosting adversarial attacks with momentum. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018, Computer Vision Foundation / IEEE Computer Society. pp. 9185–9193. https://doi.org/10.1109/CVPR.2018.00957
31. Gubri M, Cordy M, Papadakis M, Traon YL, Sen K (2022) LGV: boosting adversarial example transferability from large geometric vicinity. In: Avidan S, Brostow GJ, Cissé M, Farinella GM, Hassner T (eds.), 2022 European Conference on Computer Vision (ECCV), Springer. pp. 603–618. https://doi.org/10.1007/978-3-031-19772-7_35
32. Huang Y, Kong AW (2022) Transferable adversarial attack based on integrated gradients. In: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25–29, 2022, OpenReview.net
33. Chen B, Yin J, Chen S, Chen B, Liu X (2023) An adaptive model ensemble adversarial attack for boosting adversarial transferability. In: IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1–6, 2023, IEEE. pp. 4466–4475. https://doi.org/10.1109/ICCV51070.2023.00414
34. Guo C, Rana M, Cissé M, van der Maaten L (2018) Countering adversarial images using input transformations. In: 2018 International Conference on Learning Representations (ICLR), OpenReview.net
35. Prakash A, Moran N, Garber S, DiLillo A, Storer JA (2018) Deflecting adversarial attacks with pixel deflection. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Computer Vision Foundation / IEEE Computer Society. pp. 8571–8580. https://doi.org/10.1109/CVPR.2018.00894
36. Dziugaite GK, Ghahramani Z, Roy DM (2016) A study of the effect of JPG compression on adversarial images. CoRR abs/1608.00853. arXiv:1608.00853
37. Wang L (2021) Adversarial perturbation suppression using adaptive Gaussian smoothing and color reduction. In: IEEE International Symposium on Multimedia, ISM 2021, Naple, Italy, November 29–Dec. 1, 2021, IEEE. pp. 158–165. https://doi.org/10.1109/ISM52913.2021.00033
38. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: 2015 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9. https://doi.org/10.1109/CVPR.2015.7298594
39. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4510–4520. https://doi.org/10.1109/CVPR.2018.00474
40. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, III, WMW, Frangi AF (eds.), 2015 Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer. pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4
41. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Computer Vision Foundation / IEEE Computer Society. pp. 7132–7141. https://doi.org/10.1109/CVPR.2018.00745
42. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville AC, Bengio Y(2014) Generative adversarial nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND,

Weinberger KQ (eds), 2014 Neural Information Processing Systems(NeurIPS), pp. 2672–2680
43. Ho J, Jain A, Abbeel P (2020) Denoising diffusion probabilistic models. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H (eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, virtual. https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html
44. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261–2269. https://doi.org/10.1109/CVPR.2017.243
45. Chen Y, Li J, Xiao H, Jin X, Yan S, Feng J (2017) Dual path networks. In: Guyon I, von Luxburg U, Bengio S, Wallach HM, Fergus R, Vishwanathan SVN, Garnett R (Eds.), 2017 Neural Information Processing Systems (NeurIPS), pp. 4467–4475
46. Han D, Kim J, Kim J (2017) Deep pyramidal residual networks. In: 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6307–6315. https://doi.org/10.1109/CVPR.2017.668
47. Radosavovic I, Kosaraju RP, Girshick R, He K, Dollár P (2020) Designing network design spaces. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10425–10433. https://doi.org/10.1109/CVPR42600.2020.01044
48. Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5987–5995. https://doi.org/10.1109/CVPR.2017.634
49. Zagoruyko S, Komodakis N (2016) Wide residual networks. In: Wilson RC, Hancock ER, Smith WAP (eds) 2016 British Machine Vision Conference (BMVC). BMVA Press
50. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: Leibe B, Matas J, Sebe N, Welling M (eds) 2016 European Conference on Computer Vision (ECCV), Springer. pp. 630–645. https://doi.org/10.1007/978-3-319-46493-0_38
51. Ma N, Zhang X, Zheng H, Sun J (2018) Shufflenet V2: practical guidelines for efficient CNN architecture design. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y (eds.), 2018 European Conference on Computer Vision (ECCV), Springer. pp. 122–138. https://doi.org/10.1007/978-3-030-01264-9_8
52. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: Bengio Y, LeCun Y (eds), 2015 International Conference on Learning Representations (ICLR)
53. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N (2021) An image is worth 16x16 words: Transformers for image recognition at scale. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021, OpenReview.net
54. Athalye A, Carlini N, Wagner DA (2018) Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: Dy JG, Krause A (eds.), Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10–15, 2018, PMLR. pp. 274–283
55. Wang Y, Zou D, Yi J, Bailey J, Ma X, Gu Q (2020) Improving adversarial robustness requires revisiting misclassified examples. In: 2020 International Conference on Learning Representations (ICLR), OpenReview.net
56. Wang Z, Pang T, Du C, Lin M, Liu W, Yan S (2023) Better diffusion models further improve adversarial training. In: International Conference on Machine Learning, ICML 2023, 23–29 July 2023, Honolulu, Hawaii, USA, PMLR. pp. 36246–36263. https://proceedings.mlr.press/v202/wang23ad.html
57. Bartoldson BR, Diffenderfer J, Parasyris K, Kailkhura B (2024) Adversarial robustness limits via scaling-law and human-alignment studies. In: Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21–27, 2024, OpenReview.net. https://openreview.net/forum?id=HQtTg1try7
58. Wang Z, Wang H, Tian C, Jin Y (2024) Preventing catastrophic overfitting in fast adversarial training: A bi-level optimization perspective. In: Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXVIII, Springer. pp. 144–160. https://doi.org/10.1007/978-3-031-73390-1_9