Knowledge-Based Systems 296 (2024) 111909

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Improving the transferability of adversarial examples through neighborhood attribution

Wuping Ke [a,1], Desheng Zheng [a,c,*], Xiaoyu Li [b,c,1], Yuanhang He [d], Tianyu Li [b], Fan Min [a]

[a] School of Computer Science and Software Engineering, Southwest Petroleum University, Chengdu, 610500, China
[b] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, 610054, China
[c] Kash Institute of Electronics and Information Industry, Kash, 844000, China
[d] No. 30 Research Institute of China Electronics Technology Group Corporation, Chengdu, 610200, China

ARTICLE INFO

Keywords:
Adversarial examples
Computer vision
Neural networks
AI security

ABSTRACT

Adversarial examples, which add carefully planned perturbations to images, pose a serious threat to neural network applications. Transferable adversarial attacks, in which adversarial examples generated on the source model can successfully attack the target model, provide a realistic and undetectable method. Existing transfer-based attacks tend to improve the transferability of adversarial examples by destroying their intrinsic features. They destabilized features differentially by assessing their importance, thus rendering the model incapable of inference. However, the existing methods generate feature-importance assessments that are overly dependent on the source model, leading to inaccurate importance guidance and insufficient feature destruction. In this paper, we propose neighborhood expectancy attribution attacks (NEAA) that accurately guide the destruction of deep features, leading to highly transferable adversarial examples. First, we design a highly versatile attribution tool called neighborhood attribution to represent the importance of features that attribute highly similar results to various source models. Specifically, we discard the imputation of a single baseline and adopt the imputed expectation of a baseline within the neighborhood of the image. Subsequently, we generalize the neighborhood attribution to the middle layer of the model and simplify the computation by assuming linear independence. Finally, the attribution result guides the attack to destroy the intrinsic features of the image and obtain highly transferable adversarial examples. Numerous experiments demonstrate the effectiveness of the proposed method. Code is available at Github: https://github.com/KWPCCC/NEAA.

1. Introduction

Deep Neural Networks (DNNs) have achieved remarkable success in the areas such as pattern recognition [1], object detection [2], and autonomous driving [3]. However, they have are highly vulnerable to adversarial attacks. Adversarial attacks disrupt the model inference by adding well-designed perturbations to images that are not easily detected by humans. Adversarial attacks have been widely studied in recent years because of their detrimental effects on DNN applications [4,5]. Thus, there is an urgent need for powerful adversarial attacks to reveal model weaknesses and improve the robustness of DNN applications.

Adversarial attacks, which rely on the extent of knowledge of the model details, are divided into black-box and white-box attacks. In the context of a white-box setup, an attacker obtains information regarding the specifics of the targeted model, such as its structure and parameters [6,7]. Conversely, the black-box setup is devoid of this characteristic. Because of their ability to optimize the utility of model information in a white-box environment, these attacks are characterized by high success rates and minimal human perception. White-box attacks has seen significant advancements with the development of gradient-based [8,9], optimization-based [10,11] and decision-based methods. Conversely, in black-box attacks the attacker lacks access to the target model, making these attacks more realistic and challenging to execute. Query-based black-box offensive strategies [12,13] strive to deduce the adversarial direction by querying the targeted model; however, the large-scale budget required for queries poses a hurdle for their implementation. Transfer-based attacks are considered the most practical among black-box attack schemes, in which adversarial cases are created based on the source model for the successful execution of the targeted model. However, adversarial examples forged

[*] Corresponding author at: School of Computer Science and Software Engineering, Southwest Petroleum University, Chengdu, 610500, China.
E-mail addresses: 202121000503@stu.swpu.edu.cn (W. Ke), ds.zheng@swpu.edu.cn (D. Zheng), xiaoyuuestc@uestc.edu.cn (X. Li), heyh2477@cetcsc.com (Y. He), tianyuli@std.uestc.edu.cn (T. Li), minfan@swpu.edu.cn (F. Min).

[1] Contribute equally.

https://doi.org/10.1016/j.knosys.2024.111909
Received 25 January 2024; Received in revised form 24 April 2024; Accepted 4 May 2024
Available online 8 May 2024
0950-7051/

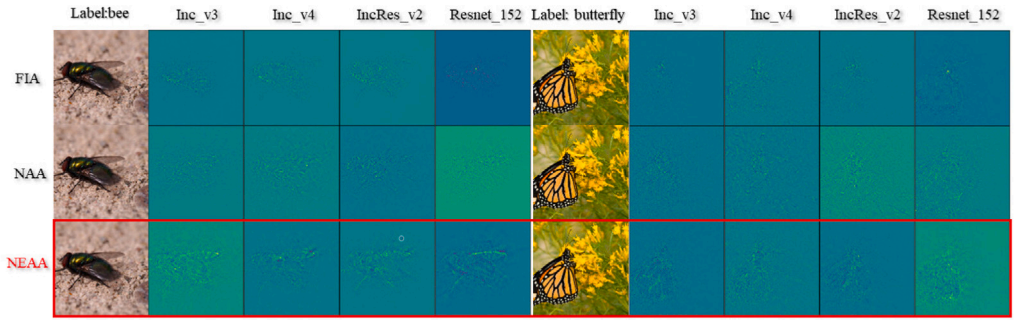

Fig. 1. Visualization of significance assessment. The three importance assessment methods are the aggregated gradient in FIA [22], the integral gradient in NAA [22], and the proposed neighborhood attribution gradient. The four source models are: Inception_v3, Inception_v4, Inception_resnet_v2, Resnet_v2_152.

using traditional methods exhibit poor transferability under black-box conditions.

To mitigate the poor transferability of adversarial examples, a multitude of studies have sought new methodologies to enhance it. Some strategies have pursued the development of in gradient-related techniques, such as the Momentum Iterative Fast Gradient Sign Method (MIM) [14] and Variance Tuning (VT) [15]. A separate body of research has leveraged alterations to the input, as demonstrated by methods such as the Diverse Input Method (DIM) [16], Translation-Invariant Method (TIM) [17], Scale-Invariant Method (SIM) [18], Path-Augmented Method (PAM) [19], and Admix [20]. A novel approach recently gaining momentum impinges on the transferability of adversarial examples through the distortion of intermediate features extracted by the model, which is evident in techniques such as the Attention-Guided Transfer Attack (ATA) [21], Feature Importance-Aware Attack(FIA) [22], and Neuron Attribution-based Attack (NAA) [23]. They accomplish this by inhibiting features at intermediate stages that would otherwise facilitate correct categorization. All these methods follow the idea of finding a precise representation of the importance of a feature. Each method attempts to devise an evaluation method that captures the contribution of features to correct classification, which is similar to the goal of neural network interpretability studies. There is a close connection between current advanced feature-based adversarial attacks and research on neural network interpretability. For example, Grad-Cam [24], input×gradient [25], and neuron attribution [26] have been applied in Attention-Guided Transfer Attack (ATA) [21], Feature Importance-Aware Attack (FIA) [22], and NAA [23], respectively. This is discussed in detail in Section 4.3. However, the current research ignores the fact that neural network interpretability studies aim to explore how a model understands the object, whereas transferable adversarial attacks aim to look for features inherent in the image. The two are similar, but fundamentally different. Interpretability studies are based on the understanding of a single model of an image, whereas the transferability of adversarial examples lies in the generalization of an image's interpretation. Therefore, the key to improving the transferability of adversarial examples is determining an inherent feature of an image and destroying it. In other words, the question is "How should images be understood?" not "How do models understand images?".

Based on this motivation, we propose a feature-level attack framework called the neighborhood expectation attribution attack (NEAA), which search for the intrinsic features of an image and interrupts them, causing the model to lose focus. Specifically, we believe that using the zero baseline (black image) as a starting point for gradient integration will accumulate a large number of strength-modeled, weak-featured interpretations in the early stages of integration. These accumulations can cause the attribution results to favor the model's understanding of the image over the interpretation of image features, contrary to the destruction of the inherent characteristics of the image. We use the image within the neighborhood of the original image as the baseline, the product of the path integral from the baseline to the original image, and the difference between the baseline and the original image as the attribution result at that point. The final attribution result is the expectation of the imputation result within the neighborhood of the original image, which is called neighborhood attribution. We generalize the attribution to the intermediate layers by assuming linear independence between the neural network levels, which ultimately destroys the critical features of the image. We use the Leeman sum to approximate the integral computation. For domain-wide expectations, we randomly sample within the neighborhood to obtain approximate expectations. The results of the neighborhood attribution in Fig. 1. It can be observed that the proposed neighborhood attribution not only summarizes the object-related features, but it is also representationally similar across models. We also quantitatively analyze the ability of neighborhood attribution to extract the intrinsic features of an image in Section 4.2.1. Because the attribution results contain highly generalized feature-importance representations, interrupting these features yields highly transferable adversarial examples. In summary, the contributions of this study are threefold.

- We design a highly generalizable attribution tool called neighborhood expectation attribution, which addresses how an image should be understood. Neighborhood expectation attribution extracts the key features of an image and not just a single model's understanding of the image.
- We propose the Neighborhood Expectation Attribute Attack by extending the neighborhood attribute to the hidden layer of the neural network and simplifying the computation by independent assumptions. The attack sufficiently destroys the intrinsic features of the image and renders the target model incapable of inference.
- Numerous experiments have demonstrated that adversarial examples generated by neighborhood expectation attribution attacks are highly transferable. Moreover, our attacks can be naturally combined with input transformation-based attacks to further improve performance.

2. Related work

2.1. White-box attacks

Given a comprehensive understanding of the target model, an attack can exploit the gradient details related to the appropriate classification to append disturbances, thereby misleading the model. The Fast Gradient Sign Method (FGSM) [27] developed adversarial examples using a one-step gradient. The FGSM serves as the groundwork for the I-FGSM [8] and PGD [9], both of which acquire adversarial examples through the incremental addition of refined disturbances. The Carlini and Wagner Attack (C&W) [10], a technique based on optimization, combines the benefits of high-throughput precision and minimal adversarial disturbances. Although adversarial attacks have yielded commendable results within white-box settings, they fall short in real-world settings because of the hurdles associated with acquiring the structure and parameters of the target model.

2.2. Black-box attacks

Black-box attacks are traditionally categorized as query-based attacks, which require accessibility to the model, and transfer-based attacks, which exploit adversarial examples designed in the original model to effectively assault alternative models. Considering that real-world models are typically confined to access, and the likelihood of query-based attacks arousing suspicion concerning the model, transfer-based attacks are more feasible. Transfer-based attacks are considered one of the most effective attacks currently available, and depend critically on the transferability of adversarial examples. Much research has been proposed for improving the transferability of adversarial examples. Several studies apply transformations to the input, encompassing techniques such as the Diversity Input Method (DIM) [16], Translation Invariant Method (TIM) [17], Channel-Augmented Attack Method (CAAM) [28], and Path-Augmented Method (PAM) [19].

Methods that derive their foundations from superior gradients and input transformations have achieved commendable results in terms of attacking unprotected models. However, models in real-world applications typically encompass defense mechanisms such as adversarial training [8]. The feature-level strategy revisits the primeval input data and aims to obliterate intermediary features to secure transferability between various models, thereby enabling adversarial examples to thwart the defense model. Transferable Adversarial Perturbation (TAP) [29] initially showed that increasing the feature deviation between a natural image and its adversarial replicas in the central layer can escalate transferability. FDA [30] disrupts the features within each layer of the network to obtain impressive transferability. FIA [22] proposed a feature importance-awareness attack, supplementing a feature importance-aware aspect based on the FDA. A recent study [22] proposed neuron attribution-based attacks, which can be attributed to neurons and guide feature destruction.

2.3. Attribution in adversarial attacks

Interpretability is recognized as a key factor in the widespread use of artificial intelligence. A substantial amount of the literature aims to tackle the importance or attribution of features within deep networks, that is, methods to ascertain the significance of each feature for a network's prediction regarding a specific input. Certain techniques accomplish this by tracing the prediction from the results back to the input [31]. Other methods depend on the prediction gradient relative to the input or on a simplified input variant [24,32].

Several methodologies have been used to observe their application in the generation of adversarial examples. Grad-CAM analyzes the gradient in the last convolutional layer to explain the importance of each feature map in a specific category. Formally, the weight of the $k-$th channel is

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k}, \tag{1}$$

where $\partial y^c/\partial A_{ij}^k$ denotes gradients of the backpropagation. The class activation map for category c under feature map A^k is represented as

$$L_{\text{Grad-CAM}}^c = ReLU(\sum_k \alpha_k^c A^k), \tag{2}$$

where k is the number of channels and A^k is the weight of channels. Attention-Guided Transfer Attack (ATA) [21] utilizes Eq. (2) as the model's attention and designs a loss function to destroy it to obtain adversarial examples. In [25], the product of the gradient and input is treated as the contribution score. Input×gradient has been used to perceive the importance of features in the middle layer as a guide map for feature destruction, thus prompting Feature Importance-Aware Attack (FIA) [22]. The attributes of neurons have been explored and the concept of conductance introduced, which is intended to indicate the importance of neuronal nodes [26]. Neuron attribution uses integrated gradients (see Section 3.2) to compute the conductance of intermediate nodes. NAA [23] perturbs the attribution of the image to the source model to destroy image features. Current advanced feature-level attacks assess the importance of features to specifically disrupt crucial features, and abundant insights are provided by studies on the interpretability of neural networks. In the motivation section, we analyze the deficiencies of neural network interpretability in the transferability of adversarial examples and propose an attribution method suitable for transferable adversarial attacks.

2.4. Adversarial defense

Models with defense mechanisms make attackers trickier. The main defenses are divided into modifying the model to improve robustness, and modifying the inputs to remove adversarial perturbations. Models equipped with defense mechanisms present a more challenging scenario for attackers. Primary defensive strategies are categorized as modifying the model to enhance its robustness and altering the inputs to mitigate adversarial perturbations. Adversarial training is considered to be one of the most effective means of bolstering model robustness. This involves exposing the model to adversarial examples during the training phase, thereby endowing it with defensive capabilities. Madry et al. [9] were among the first to theoretically explore the effects of adversarial training on model robustness. Methods based on input transformations aim to eliminate deleterious adversarial perturbations within the examples. For instance, Liu et al. [33] investigated a JPEG-based compression technique in which compressed images substantially lost their adversarial properties. Sun et al. [34] employed convolutional sparse coding to transform input images and project them onto a space with reduced adversarial susceptibility.

3. Methodology

3.1. Motivation

The study of neuronal importance/attribution explains how a model understands an image. Adversarial attacks based on feature destruction use these features to find and destroy important features that misidentify a model. For example, ATA [21] is based on Grad-CAM [24] for attentional representations, FIA [22] is based on the product of the gradient and input, and NAA [23] is based on neuronal attribution [26]. These methods ignore a key issue in the transferability of adversarial examples, namely, the multi-model matching of feature importance. Specifically, these interpretability methods explain how a particular model understands an image, with a focus on the model. In the transferability of adversarial examples, the key point is the destruction of the category features of the image itself. Therefore, previous research has focused on improving a specific model's understanding of an image by ignoring the adaptation of feature representations to multiple models, which is detrimental to the transferability of adversarial examples. Fig. 2 summarizes our motivation. We expect to find a method of feature importance assessment that focuses on the inherent features of an image and can be adapted to the interpretation of different models.

Interpretability or attribution studies explain the contribution of each input variable to the output of a model. However, these studies do not consider how other models produce the same input. The imputation method uses a black image or single image as the starting point for the integration gradient. It expresses the sum of the contributions of the features along the linear path from the baseline to the original image. The singularity of the feature changes leads to attribution results that are heavily model-dependent. We believe that the diversity of feature changes should be enhanced to obtain the attribution results for the concerned images. We use the neighborhood of the original image as the baseline and the expectation of the gradient integral from the images in the neighborhood to the original image as the attribution result, called the expectation attribution. The neighborhood of the original

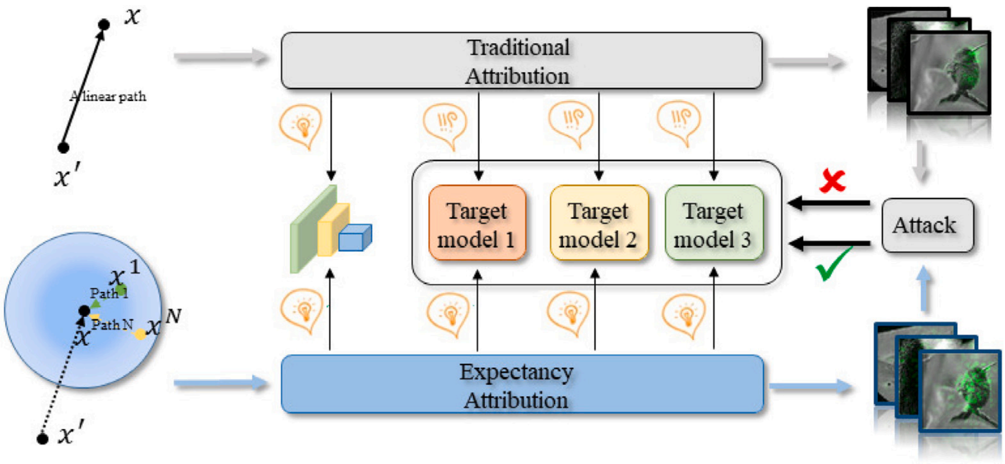

Fig. 2. Motivation for neighborhood expectation attribution attack. x: original image, x': black image, $x^{1\dots N}$: neighborhood image. The proposed attribution method accumulates gradients within the neighborhood of the target. Neighborhood expectation attribution summarizes the inherent features of the image and adapts to various target models.

image provides different feature variations, and the imputation results originate from multiple paths. The imputation result focuses more on the sensitivities due to feature loss than on the superposition of the sensitivities to the model over linear paths. Therefore, EA contributes to the contribution of the response features to the image class rather than the contribution of the response features to the model output.

3.2. Integrated gradients

Recall the definition of an integral gradient. For the image data, the sum of the gradients at all points on a linear path from the baseline to the image when obtain an uninformative baseline (black image) is the integrated gradient. Specifically, the integrated gradient is defined as the path integral of the gradient along the linear path from x' to x. The integral gradient is the sum of all the points on the linear path from the baseline to the image. Formally, the integrated gradient of x with respect to the ith feature of x' is defined as:

$$IG_i(x) ::= (x_i - x'_i) \cdot \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} d\alpha, \tag{3}$$

where $\partial F(x' + \alpha(x - x'))/\partial x_i$ denotes the partial derivative of F up to the ith pixel. Eq. (3) is the path integral of the gradient of F along the line $(x' + a(x - x'))$. The integral gradient accumulates the microscopic interpretation of the model from the baseline to the image.

3.3. Neighborhood baseline

Attribution methods reveal how an object recognition network understands an image. However, our goal is to assess the importance of features inherent in the image. When a black image is used as a baseline, points close to the baseline carry a very small number of features. The initial stage of path integration involves accumulating numerous model interpretations and ignoring intrinsic features of the image. Therefore, we use an image within the neighborhood of the original image as the baseline. Specifically, x is the original image and r is the normal distribution, where $x + r$ is the neighborhood baseline. Formally:

$$x' = x + r, r \sim N(\mu, \sigma), \tag{4}$$

where μ is the mean of the normal distribution, set to 0, and σ is the variance of the normal distribution as a hyperparameter. The neighborhood baseline is closer to the original image than the zero baseline, and the image feature information is more dominant in the gradient. Therefore, the neighborhood baseline can correct the attribution to the image.

3.4. Expectancy attribution

From the uninformative baseline to the original map, features are accumulated from none, attributing the results to feature changes. Using only the neighborhood baseline makes the accumulation of importance for some features insufficient. We use the expectation of the attribution of the neighborhood baseline as the feature importance, called expectancy attribution (EA). The attribution result can be expressed as:

$$EA_i(x) ::= \mathbb{E}_{x'=x+r}[(x_i - x'_i) \cdot \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} d\alpha], \tag{5}$$

where $\mathbb{E}$ denotes expectancy, x' denotes baseline image, i denotes feature pixel points.

However, in practice, expectations cannot be computed directly because of the continuity of the input space. Therefore, we compute $EA_i(x)$ by sampling N samples in a neighborhood of x to approximate its value:

$$EA_i(x) \approx \frac{1}{N} \sum_{j=1}^{N} (x_i - x_i^j) \cdot \int_{\alpha=0}^{1} \frac{\partial F(x^j + \alpha(x - x^j))}{\partial x_i} d\alpha, \tag{6}$$

where N denotes the number of baselines, i denotes i−th pixel of input.

3.5. Neighborhood expectancy attribution attack

Expectation attribution characterizes the importance of the inherent features of an image, and destroying these features causes the object recognition model to lose its ability to recognize or misrecognize them. Several studies have demonstrated that destroying intermediate features yields highly transferable adversarial examples [22,23]. We generalize expectation attribution to the intermediate layer of neural networks. For ease of understanding, we denote $x' + \alpha(x - x')$ as x_α. Eq. (6) represents the attribute of the i−th feature point to the input. When generalized to the intermediate layer, each input contributes to the conductance of the intermediate node. Therefore, it is necessary to sum each input, and we assume that the input image size is $(M\times M)$. The attribution of the intermediate layer can be formulated as

$$EA_{y_k}(x) ::= \frac{1}{N} \sum_{i=1}^{M^2} \sum_{j=1}^{N} (x_i - x_i^j) \cdot \int_{0}^{1} \frac{\partial F}{\partial y_k}(y(x_\alpha)) \frac{\partial y_k}{\partial x_i}(x_\alpha) d\alpha, \tag{7}$$

where y_k denotes the kth feature point of the yth layer, and M^2 denotes the number of input feature points. EA_{y_k} denotes the attributes of neurons in the hidden layer. The integral computation is computationally complex. In practice, we use the Riemann sum of n proxy images to

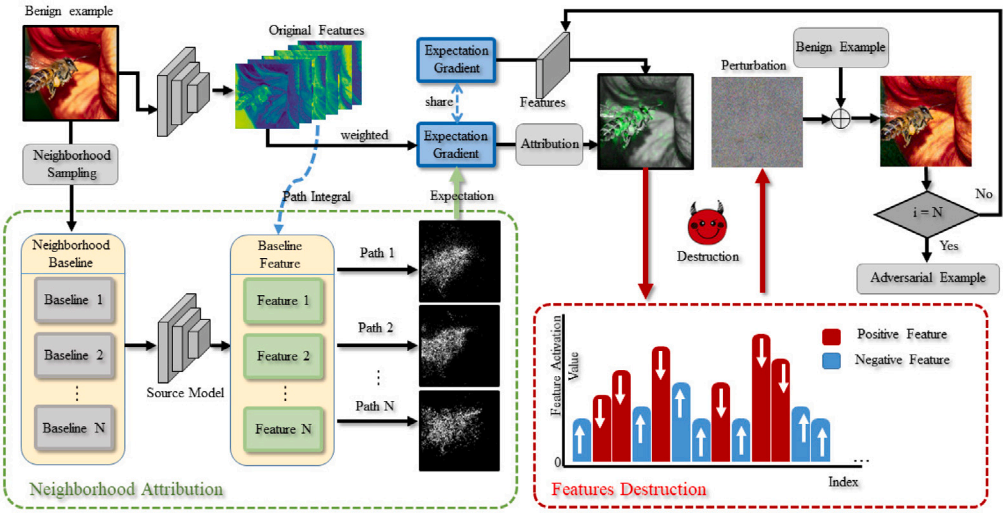

Fig. 3. Overview of neighborhood expectancy attribution attack. First, the neighborhood attribution module gets the feature importance assessment. Then, the neighborhood attribution guides the destruction of the intrinsic features of the image and obtains the perturbation. Finally, features are iteratively destroyed and adversarial examples are obtained based on the control of the number of iterations.

Algorithm 1 Neighborhood Expectancy Attribution Attack

```
Input : A classifier $F$ with loss function $J$ , A benign example $x$ with ground-truth label $y$ ;
Parameters : the maximum perturbation $\epsilon$ , neighborhood boundary $\sigma$, iterations $T$, decay factor $\mu$ , number of baselines $j$, and integrated step $n$;

Output : Adversarial example.
Let $EA = 0$, $g_0 = 0$, $\mu = 1$, $\alpha = \epsilon/\mathrm{T}$
for j = 0 to $N - 1$ do
    Sample $R$ with the same shape as $x$: $R \sim N(\mu, \sigma)$
    Obtain a neighborhood baseline: $x' = x + R$
    Calculate a neighborhood gradients:
    for $m = 1 \leftarrow n$ do
        $EG_y = EG_y + \nabla_{y(x'+\frac{m}{n}(x-x'))} F(x' + \frac{m}{n}(x - x'))$
    end for
    $EG = EG/\|EG\|_2$
    Get neighborhood expectation attribution:
    $EA_y = EA_y + (y - y') \cdot EG_y$
end for
$EA_y = EA_y/N$
for t = 0 to $T - 1$ do
    $g_{t+1} = \mu \cdot g_t + \frac{\nabla_x EA_y}{\|\nabla_x EA_y\|_2}$
    $x^{\mathrm{adv}}_{\mathrm{t+1}} = Clip_\varepsilon\{x^{\mathrm{adv}}_{\mathrm{t}} - \alpha \cdot \mathrm{sign}\left(g_{\mathrm{t+1}}\right)\}$
end for
Return $x^{\mathrm{adv}} = x^{\mathrm{adv}}_t$.
```

approximate the integral value. Thus, the attribution of the importance of the intermediate features of the response is attributed:

$$EA_{y_k}(x) \approx \frac{1}{Nn}\sum_{j=1}^{N}\sum_{m=1}^{n}(\frac{\partial F}{\partial y_k}(y(x^j_m)))(\sum_{i=1}^{M^2}(x_i - x^j_i)\frac{\partial y_k}{\partial x_i}(x^j_m)), \tag{8}$$

where $x_m = x' + \frac{m}{n}(x - x')$ are segmented images.

In Eq. (8), the attributes of each neuron in the target layer must be computed, which is computationally expensive. In most visual models, the neighboring layers are linearly independent. For Eq. (8), $\partial F/\partial y$ denotes the derivation of the output to the target layer, related to the latter layer of the target layer. $\partial y/\partial x$ denotes the derivation of the target layer to the input, related to the former layer of the target layer. We assume that these two components are linearly independent. For two linearly independent sequences A_i and B_i, their covariance is 0, i.e., $\sum_1^n(A_i - \bar{A}_i)(B_i - \bar{B}_i) = 0$. Then: $\sum_1^n A_i \cdot B_i = 1/n\sum_1^n A_i{\cdot}\sum_1^n B_i$. Thus, Eq. (8) can be written as

$$EA_{y_k}(x) \approx \frac{1}{Nn}\sum_{j=1}^{N}\sum_{m=1}^{n}(\frac{\partial F}{\partial y_k}(y(x^j_m)))\frac{1}{n}\sum_{m=1}^{n}\sum_{i=1}^{M^2}(x_i - x^j_i)\frac{\partial y_k}{\partial x_i}(x^j_m). \tag{9}$$

Continuing Eq. (9), we have $\frac{1}{n}\sum_{m=1}^{n}\sum_{i=1}^{M^2}(x_i - x^j_i)\frac{\partial y_k}{\partial x_i}(x^j_m){=}y_k{-}y^j_k$, y^j_k is the activation of the baseline sampled in the jth neighborhood at the kth level. We write $\frac{1}{n}\sum_{m=1}^{n}(\frac{\partial F}{\partial y_k}(y(x^j_m)))$ as $EG(y^j_k)$, i.e., neighborhood gradients, and $y_k - y^j_k$ as Δy_k. So, $EA_{y_k} \approx \frac{1}{N}\sum_{j=1}^{N} EG(y^j_k)\cdot\Delta y_k$. The EA_{y_k} reacts to the neighborhood attribution of the kth layer of the neural network. We use the product of the relative activation value and neighborhood gradients as the result of the neighborhood attribution. The computational complexity of neuron imputation is $\mathcal{O}(H * W * C)$, where H, W, and C are the height, width, and number of channels in the middle layer, respectively. After simplification, the time complexity of the neighborhood imputation is $\mathcal{O}(N)$, where N is the number of samples in the neighborhood baseline. Our approximation significantly reduces the computational complexity. As a result, the sum of the attribution of all neurons in the target layer is:

$$EA_y \approx \frac{1}{N}\sum_{j=1}^{N}\sum_{y_k\in y} EG(y^j_k)\cdot\Delta y_k = \frac{1}{N}\sum_{j=1}^{N} EG(y^j)\cdot\Delta y \tag{10}$$

Intuitively, we destroy the intermediate features of the image by reducing the positive attribution and enhancing the negative attribution. Fig. 3 shows the framework of the neighborhood expectancy attribution attack. We use EA_y as the optimization objective:

$$\min_{x^{\mathrm{adv}}} EA_y s.t.\left\|x - x^{\mathrm{adv}}\right\|_\infty < \varepsilon. \tag{11}$$

We use momentum [14] to optimize Eq. (11). Algorithm 1 summarizes the neighborhood attribution attacks.

4. Experiments

4.1. Experiment setup

4.1.1. Dataset

We assessed the proposed approach on an ImageNet-compatible dataset [8,35], similar to the benchmarks referred into prior studies [21,22,36]. This specific dataset, incorporated in the NIPS 2017 adversarial competition, comprises a compilation of 1000 images.

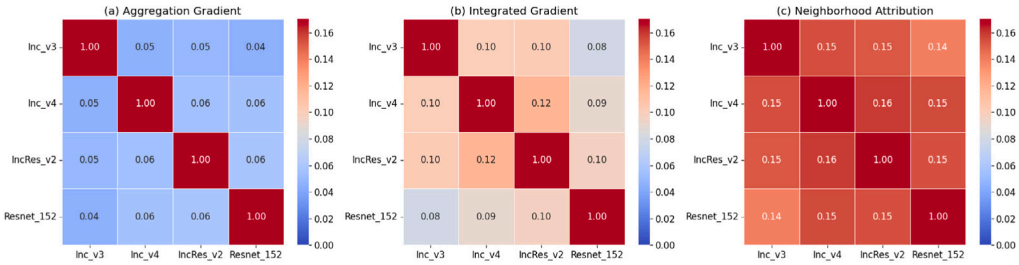

Fig. 4. Heatmap of similarity of importance assessments across models. Four source models: inception_v3, inception_v4, inception_resnet_v2, resnet_v2_152. (a): Aggregate gradients in FIA. (b): Integrated gradients in NAA. (c): Neighborhood attribution.

4.1.2. Baseline attacks

We evaluated the transferable adversarial attack methods based on multiple mechanisms. Gradient-based methods, i.e., MIM [14] and VMI [15], input transformation-based methods, i.e., DIM [16], Admix [20] and PAM [19], and feature-based methods, i.e., FDA [30], FIA [22], RPA [37], and NAA [36]. For model defense, we analyzed highly efficient defense techniques, including TIM [17] and PIM [38]. Furthermore, we conducted a comparative study of variations in these strategies, such as PD, which is a fusion of PIM and DIM.

4.1.3. Models

For evaluation purposes, we cross-validated our proposed approach across a spectrum of models, both unprotected and fortified, through adversarial training. Unprotected models include a variety of architectures,including Inc-v3 [39], Inc-v4 [40], IncRes-v2, Res-50 [41], and Res-v2-152, in addition to Vgg-16 [42] and 19. Fortified models, which went through adversarial training [8,43], also encompass an array such as the Adv-Inc-v3, Adv-IncRes-v2, Ens3-Inc-v3, Ens4-Inc-v3, and Ens-IncRes-v2. For our experiments, we selected Inc-v3, and Res-v2-152, Inc-v4 as the source models from which we initiate our attacks against the remaining models.

Furthermore, in our endeavor to assess the efficacy of attacks on defensive models thoroughly, we expanded our evaluation to include nine additional advanced defenses. This includes the top-three models ranked in the NIPS 2017 competition: HGD [44], R&P [45], and NIPS-r3. In addition, we also consider two sophisticated input transformation-based defenses, JPEG [46] and FD [33], as well as a certified defensive model, RS [47].

4.1.4. Implementation details

The procedure for configuring parameters is executed as follows: We establish a disturbance tolerance, denoted as ϵ, at 16, in conjunction with a limit on iteration, symbolized by T, set at 10, to derive a step rise $\alpha = \epsilon/T$. The NEAA's range of neighborhood as well as its customary momentum remain constant at respective values of $\sigma = 1$ and $\mu = 1$. In terms of the stochastic modification of T, we employ the random pixel concealment approach, elaborated in [22], which consequently results in a probability of pixel exclusion at $p = 0.2$. In accordance with the frameworks recommended by [38], a parameter value assignment for PIM is decreed at $\beta = 10$ and $\gamma = 16$, and for PD, $\beta = 2.5$ with $\gamma = 2$. Meanwhile, the kernel dimension k_w for the projection remained uniformly stipulated at three for both methods. When considering the selection of layers for the targeted strategy, our choice falls on Mixed_5b for Inc-v3, Mixed_5a for Inc-v4, and the final layer of the second block for Res-152.

4.2. Results

4.2.1. Generalizability of neighborhood expectation attribution

Fig. 2 shows the visualization results of the neighborhood attribution. In this section, we quantitatively report the ability of neighborhood attribution to express images. Existing state-of-the-art feature-based attack methods use a guidance map to express features. These guidance maps are almost always gradient-based; therefore, we treat the guidance map as a vector to evaluate their relevance across models. Specifically, we evaluated the current top-performing methods, namely, FIA, NAA, and the proposed NEAA. Fig. 4 shows the cosine similarity of the guidance maps of these methods on different models, which are derived from the mean of 1000 randomly selected images. The proposed neighborhood attribution method generated the highest correlation of bootstrap images on different models, which proving that the attribution results can be expressed by various models. In other words, the neighborhood expectation attribution method is weakly model-based, feature-heavy, and can determine the intrinsic features of an image.

Table 1
All metrics are mean values of 1000 randomly selected images. OLNR and NLOR are the means of all 12 models. Cosine dist and NRT dist is computed at the attacked layer.

	IoU	OLNR	NLOR	Cosine Dist.	NRT Dist.
FDA	0.368	101.19	222.62	0.69	0.093
FIA	0.334	338.11	333.64	0.73	0.094
RPA	0.319	277.24	327.45	0.74	0.107
NAA	0.302	238.87	260.71	0.70	0.094
NEAA (Ours)	**0.283**	**451.31**	**387.14**	**0.78**	**0.129**

4.2.2. Effectiveness of feature disruptions

Fig. 5 shows the attention before and after the NEAA attack, as well as the visualization of the neighborhood expectation attribution. The generated adversarial examples distract the model and demonstrate the effectiveness of the feature destruction. To measure the destruction of feature attributes, we utilize the metric of cosine distance and the Normalized Rank Transformation (NRT) distance, comparing the features pre- and post-attacks, as presented in Table 1. The maximum distance was achieved using the proposed attack. We evaluated the two metrics proposed in [30], named New Label Old Rank (NLOR) and Old Label New Rank (OLNR), which indicate the magnitude of labeling changes before and after destruction. Intuitively, the more severe the feature destruction, the more prominent is the labeling change. Table 1 displays the OLNR and NLOR, which are significantly higher than the additional methods, suggesting that our attacks increase category bias. In addition, we evaluated the distraction of adversarial examples based on the attention of the model in Table 1. We calculated the intersection over union (IoU) of the high-heat regions in Grad-CAM [24] before and after the attack.

4.2.3. Attacking normally trained models

Table 2 illustrates the effectiveness of the attack on the normally trained models. It is evident from the results that our technique surpasses all other tactics across all models. In particular, the impressive attack performance scores of our method utilizing Inc-v3, Res-152 and Inc-v4 averaged 83.4%, 91.9%, and 86.1%, respectively, across all the targeted models. Significantly, Res-152 paired with our process yielded the highest attack efficacy. We infer that the sophistication of the source model contributes to transferability and that simpler

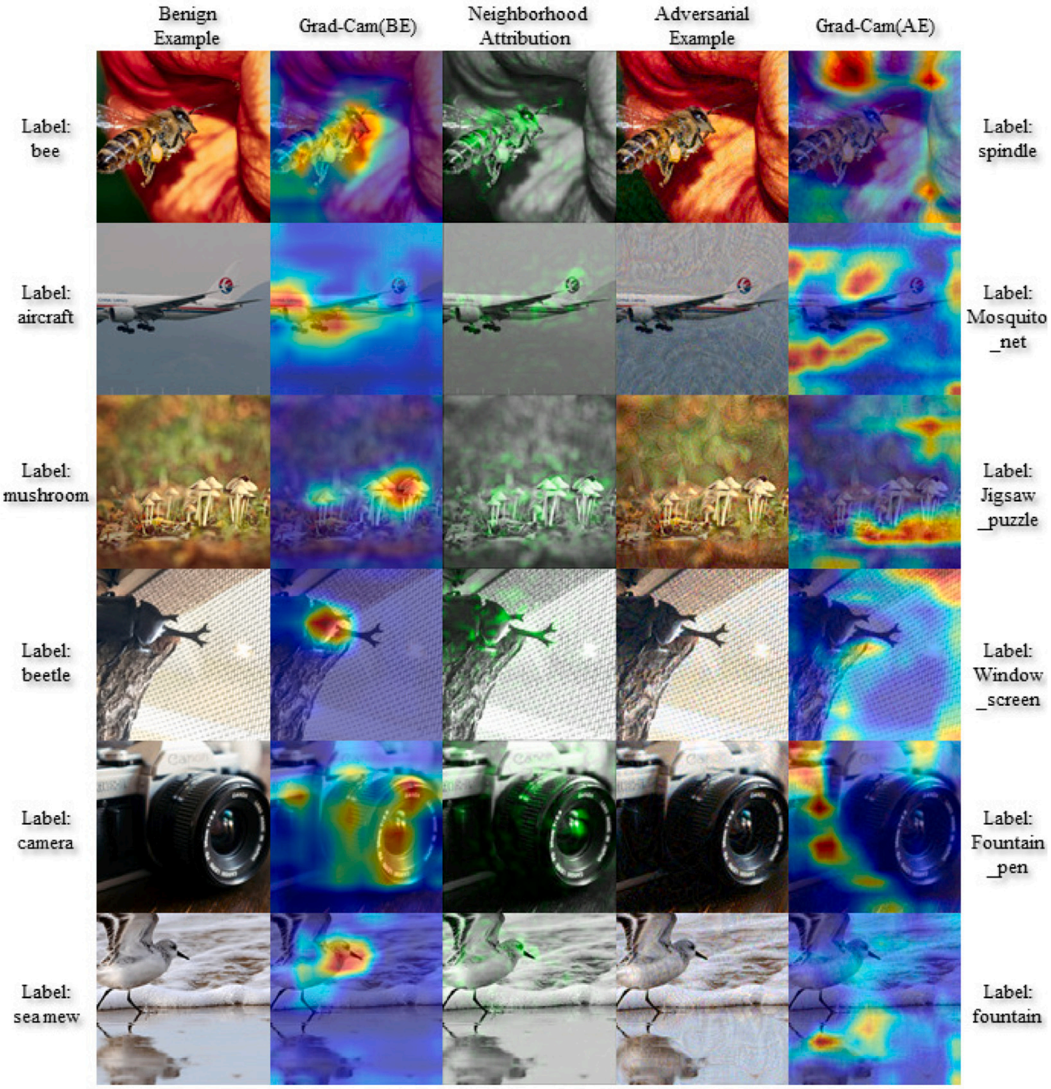

Fig. 5. Visualization of examples. Visualization of neighborhood attribution: the attribution result is linearly interpolated to the original image and overlaid on the original image. Attention visualization using Grad-Cam [24].

models enable superior feature destruction. We obtained slightly lower results in some white-box settings than in other methods based on input transformations. The proposed attribution method targets the image itself, causing the attack to lose effort in the source model.

Furthermore, we assessed the potency of the attacks generated by the combined models. For instance, PD is derived by integrating PIM and DIM, FIA+PD is derived from FIA and PD, and NAA+PD is a blend of NAA and PD. The results of these experiments underscore the enhanced attack proficiency of the NEAA+PD method.

4.2.4. Attacking defense models

The adversarial training process employed with defensive models enhances the resistance against adversarial examples. As is evident in Table 3, we see the superior efficacy levels of our method at 53.4%, 77.8%, and 58.1%, indicating an average improvement of 2.1%, 6.9%, and 10.3%, respectively, compared with NAA. When employing Res-152 as the source model, our method achieved an average success rate of 77.8% compared to adversarially trained models. Moreover, when analyzing the impact of the combined technique on the adversarially trained model, NEAA consistently ranked higher. It is worth noteworthy that PAM, which attacks normally trained models, performs poorly on adversarially trained models. Other gradient or input transformation-based methods suffer from this problem. However, feature-based methods also perform well in defense models. To test the advanced defense models, we explored our method's capacity beyond adversarially trained models, resulting in an increase in the average success rate from 59.9% to 62.0%, as indicated in Table 4. The proposed method is not as effective as black-box when white-box, especially adversarially trained models. This point side-steps our motivation that overfitting to the source model decreases when feature destruction is more focused on the image. The adversarial examples generated by NEAA can be interpreted as exploring a more comprehensive adversarial space. Adding these examples to the adversarial training process helps to improve the robustness of the model.

Transferability can be further improved by ensemble training [48], that is, by applying information from multiple source models. Specifically, during ensemble training, we opted to use Inc-v3, Res-152 and Inv-v4 as source models, resulting in our optimization goal being the aggregation of EG for these four source models. As shown in Table 5, we observed an enhancement in the performance of the proposed method from 84.8% to 93.6%.

4.2.5. Effect of attacking various layers

We explored the impact on the performance when different hidden layers are used as attack targets. We chose Conv_2b, Mixed_5b,

Table 2
Attack success rate of the normally trained models. The source models are Inc-v3, Res-152, and Inc-v4. The first row is the test model. "*" denotes white-box attacks. NEAA is our method. We evaluated the effect of a single attack and combine it with other attacks separately, e.g. NEAA+PD for NEAA combined with PIM and DIM.

Model	Attack	Inc -v3	Inc -v4	IncRes -v2	Res -50	Res -152	Vgg -16	Vgg -19
Inc-v3	MIM	**100***	48.5	48.1	41.2	35.1	40.1	38.3
	DIM	99.5*	63.7	59.6	39.7	45.1	47.5	45.3
	FDA	76.6*	47.1	43.3	40.1	32.2	31.5	31.9
	Admix	**100***	82.2	79.1	71.6	63.0	70.6	64.3
	VMI	**100***	71.7	68.1	57.1	53.9	59.3	55.3
	FIA	98.4*	83.5	80.6	69.5	65.5	71.6	72.9
	RPA	98.6*	85.7	84.0	72.7	68.4	75.3	67.2
	NAA	98.1*	85.0	82.5	71.0	67.0	71.9	71.3
	PAM	**100***	83.8	81.2	76.9	70.5	72.1	65.3
	NEAA (Ours)	99.3*	**88.0**	**87.2**	**78.9**	**73.6**	**78.9**	**78.5**
	FIA+PD	98.9*	87.8	85.7	79.2	74.0	82.0	84.0
	RPA+PD	98.5*	89.6	88.7	83.0	79.8	83.7	80.9
	NAA+PD	98.8*	89.4	88.4	85.5	80.0	85.1	81.3
	PAM+PD	**100***	85.4	87.3	79.6	77.1	80.2	78.8
	NEAA+PD (Ours)	99.3*	**93.1**	**92.5**	**90.1**	**88.2**	**87.2**	**88.1**
Res-152	MIM	57.2	48.2	45.7	90.6	98.8*	72.8	72.9
	DIM	80.3	75.2	76.6	97.0	99.2*	88.4	88.0
	FDA	60.7	52.3	48.0	85.0	95.3*	75.0	75.0
	Admix	84.0	76.5	75.4	96.3	99.9*	89.0	80.1
	VMI	74.3	62.1	69.9	92.1	**100***	76.5	73.3
	FIA	88.7	84.1	83.3	96.5	99.7*	90.2	88.3
	RPA	89.0	85.8	85.7	97.9	99.6*	94.3	89.1
	NAA	85.8	85.0	83.1	94.6	98.0*	86.4	84.1
	PAM	81.7	77.4	76.8	97.1	99.9*	88.3	83.9
	NEAA (Ours)	**89.4**	**85.9**	**86.3**	**98.1**	99.9*	**94.6**	**89.7**
	FIA+PD	90.7	85.8	85.6	97.8	99.6*	94.7	95.7
	RPA+PD	94.3	90.5	91.2	98.5	99.7*	97.6	94.1
	NAA+PD	92.0	90.3	90.0	98.7	98.4*	96.6	94.3
	PAM+PD	89.3	88.5	87.8	98.3	99.9*	89.1	89.6
	NEAA+PD (Ours)	**95.9**	**93.4**	**92.6**	**98.8**	**99.9***	**97.9**	**95.8**
Inc-v4	MIM	60.1	99.6*	47.7	44.0	39.9	60.4	57.5
	DIM	72.6	99.3*	62.1	37.7	40.8	75.1	77.0
	FDA	84.6	99.6*	71.8	70.0	68.8	75.0	75.0
	Admix	87.8	99.4*	83.2	77.3	74.5	78.9	73.8
	VMI	77.9	99.8*	71.2	63.3	61.6	62.7	59.9
	FIA	83.4	95.4*	77.2	73.4	72.0	71.6	68.5
	RPA	82.5	95.7*	79.3	74.7	70.8	70.4	69.2
	NAA	86.0	96.5*	81.0	72.1	71.3	69.7	67.2
	PAM	89.7	**100.0***	84.5	80.6	80.5	77.4	73.5
	NEAA (Ours)	**90.9**	98.8*	**86.3**	**83.4**	**81.1**	**81.9**	**80.5**
	FIA+PD	90.5	97.0*	88.6	85.9	84.8	89.4	87.7
	RPA+PD	91.7	97.7*	88.5	86.2	85.1	88.9	89.4
	NAA+PD	90.4	96.5*	89.0	82.2	86.6	82.5	81.4
	PAM+PD	90.3	97.9*	84.8	85.9	80.6	79.5	76.2
	NEAA+PD (Ours)	**95.9**	**98.6***	**90.7**	**89.1**	**88.8**	**88.9**	**89.8**

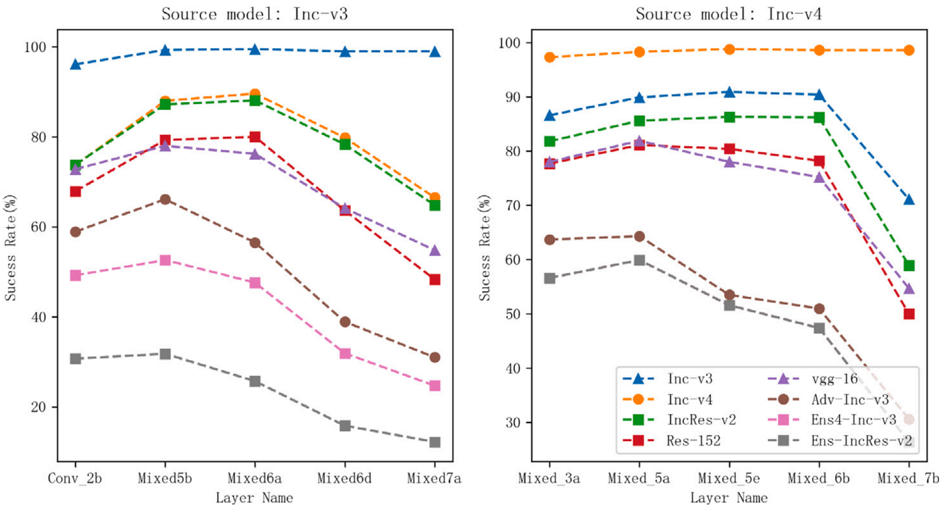

Fig. 6. Effect of selecting different attack layers. The source models are Inc-v3 and Inc-v4. The target models are all the test models.

Table 3
Attack success rate of the adversarially trained models.

Model	Attack	Adv- Inc-v3	Adv- IncRes-v2	Ens3- Inc-v3	Ens4- Inc-v3	Ens- IncRes-v2
Inc-v3	MIM	22.9	17.5	15.4	15.8	7.8
	DIM	26.0	24.5	17.8	14.8	9.0
	TIM	31.9	26.3	30.9	31.2	22.2
	PIM	34.1	30.4	33.3	38.8	25.9
	Admix	47.7	48.0	39.7	39.4	19.2
	VMI	39.7	41.1	32.8	31.2	17.5
	FIA	54.6	54.8	43.7	43.4	24.0
	NAA	61.5	62.7	50.3	50.8	31.5
	RPA	59.0	59.5	45.5	44.1	26.3
	PAM	53.8	54.4	44.7	43.4	23.5
	NEAA (Ours)	**66.1**	**65.1**	**51.4**	**52.6**	**31.8**
	FIA+PD	61.2	57.3	38.9	37.2	22.3
	NAA+PD	67.9	68.6	55.4	55.6	33.8
	RPA+PD	64.8	65.6	49.6	53.2	29.4
	PAM+PD	55.7	56.2	46.3	44.7	25.6
	NEAA+PD (Ours)	**72.5**	**71.7**	**59.1**	**56.9**	**37.6**
Res-152	MIM	36.9	34.8	36.2	37.4	22.0
	DIM	54.3	54.6	35.3	29.4	18.5
	TIM	42.7	38.4	34.3	38.9	27.7
	PIM	40.1	38.8	46.9	50.0	38.1
	FIA	81.3	74.7	76.6	75.6	66.2
	Admix	51.7	54.9	49.7	46.6	29.5
	VMI	43.2	30.1	44.1	39.3	27.1
	NAA	75.3	71.6	73.4	73.2	61.1
	RPA	73.7	71.0	66.9	65.1	46.1
	PAM	38.1	33.7	53.2	47.0	33.3
	NEAA (Ours)	**83.9**	**79.8**	**78.9**	**78.2**	**68.3**
	FIA+PD	66.1	62.6	65.9	68.7	50.2
	NAA+PD	76.0	78.3	70.4	68.0	52.8
	RPA+PD	70.3	73.5	69.4	63.7	50.4
	PAM+PD	44.3	40.6	65.3	51.4	34.9
	NEAA+PD (Ours)	**76.6**	**79.6**	**71.9**	**69.7**	**52.3**
Inc-v4	MIM	23.8	21.2	22.7	21.5	11.9
	DIM	25.1	27.7	23.2	21.1	11.6
	TIM	26.3	27.1	26.2	23.4	20.2
	PIM	45.9	48.8	51.9	50.0	38.1
	FIA	45.3	47.3	38.0	37.2	19.4
	Admix	49.9	55.4	50.6	47.5	31.3
	VMI	37.1	42.6	38.4	39.5	24.0
	NAA	52.4	56.0	50.5	49.4	30.8
	RPA	42.7	50.5	44.3	49.1	29.2
	PAM	50.3	56.7	57.3	54.5	34.7
	NEAA (Ours)	**64.3**	**67.5**	**59.9**	**58.2**	**41.0**
	FIA+PD	55.3	60.6	45.4	42.0	23.5
	NAA+PD	59.3	73.5	54.4	53.3	31.5
	RPA+PD	57.7	60.5	49.1	46.6	30.3
	PAM+PD	50.3	58.4	57.2	52.3	34.9
	NEAA+PD (Ours)	**73.3**	**75.4**	**66.9**	**64.7**	**47.2**

Table 4
Attack success rate for advanced defense models.

Attack	HGD	R&P	NIPS-r3	JPEG	FD	ComDefend	RS	Average
MIM+PD	25.8	22.6	28.3	30.4	62.5	59.9	31.3	37.3
FDA+PD	19.9	16.6	22.7	27.1	37.0	37.5	27.8	26.9
FIA+PD	37.4	36.6	51.3	53.9	76.7	74.5	38.6	52.7
NAA+PD	48.3	46.7	61.4	62.1	80.1	78.9	39.7	59.9
RPA+PD	45.2	43.1	55.7	57.3	77.9	79.1	38.5	56.7
PAM+PD	47.1	38.7	43.6	50.1	73.0	69.7	37.7	58.1
NEAA+PD (Ours)	**52.8**	**50.4**	**61.8**	**64.2**	**81.3**	**80.3**	**43.7**	**62.0**

Table 5
Attack success rate of the defense models when using an ensemble attack that contains Inc-v3, Res-152 and Inc-v4. The target models are five defense models.

Attack	Adv- Inc-v3	Adv- IncRes-v2	Ens3- Inc-v3	Ens4- Inc-v3	Ens- IncRes-v2
MIM	69.5	69.3	70.1	71.2	50.8
DIM	80.4	84.7	82.0	79.9	65.5
TIM	66.7	60.9	69.4	71.7	60.1
PIM	73.3	68.1	76.3	77.4	69.1
TIDIM	74.3	68.3	75.7	77.8	67.3
PITIDIM	72.3	66.7	77.1	79.3	67.7
FIA	90.8	89.9	88.4	88.7	75.1
NAA	89.7	90.1	86.3	80.5	77.3
RPA	91.9	88.7	90.1	88.9	77.2
PAM	61.2	60.7	57.4	55.7	42.5
NEAA (Ours)	**96.3**	**92.7**	**94.4**	**94.3**	**90.2**

Mixed_6a, Mixed_6d, Mixed_7a for Inc-v3 and Mixed_3a, Mixed_5a, Mixed_5e, Mixed_6b, Mixed_7b for Inc-v4.

Fig. 6 illustrates the success rate of the attack when different hidden layers are used as targets. We observed that the attack is most effective when the middle layer is used as the target. Different depths have different feature expressiveness; shallow features are close to the input and contain more texture information, whereas deep features are close to the output and contain more semantic information. The middle layer balances both factors. Second, deep feature maps tend to be compact, limiting the comprehensiveness of feature breaking.

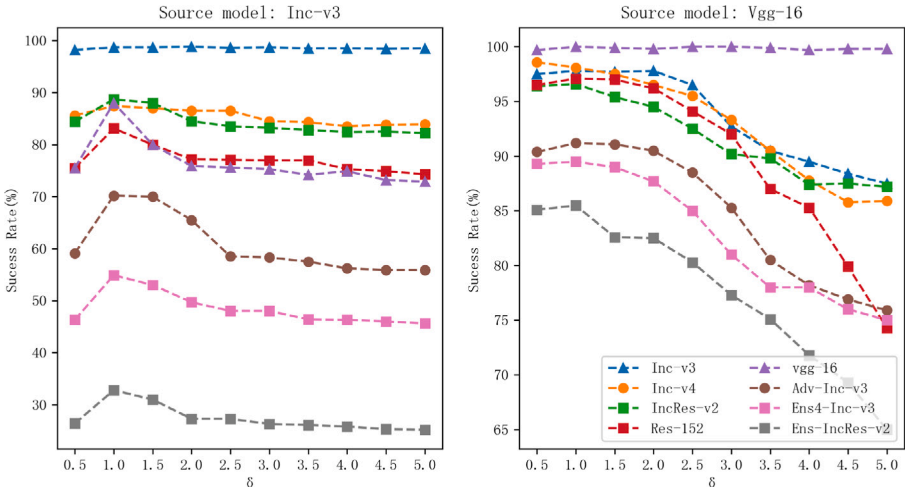

Fig. 7. Effect of neighborhood boundary on attack success. The source models are Inc-v3 and Vgg-16. The horizontal coordinate is the neighborhood boundary and the vertical coordinate is the attack success rate.

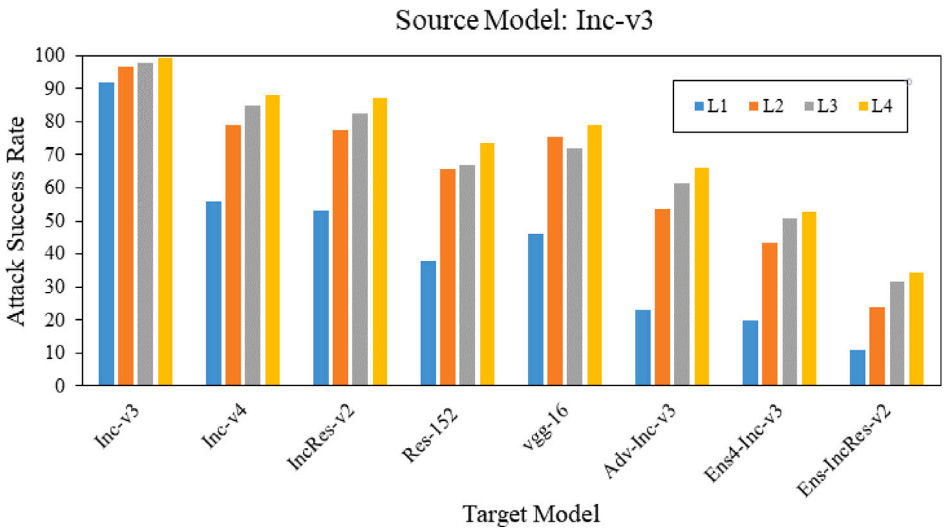

Fig. 8. Effect of NEAA on attack success rate. L_1 does not use importance guidance to optimize feature interference, L_2 uses raw gradient, L_3 adopts aggregate gradient on transformation inputs, and L_4 adopts neighborhood expectation attribution.

4.2.6. Effects of neighborhood boundary

Fig. 6 illustrates the effect of different neighborhood boundaries on the success rate of the attack. Neighborhood boundaries are crucial for attributing neighborhood expectation. Regardless of whether Inc-v3 or Vgg-16 is used as the source model, the effect tends to increase when the neighborhood boundary is in the range of [0.5, 1.0] and decreases thereafter. When the neighborhood boundary is too small, the baseline is close to the original sample, and the gradient accumulation is overly homogeneous. When the neighborhood boundary is too large, the baseline changes the features significantly, and the accumulated gradient introduces a significant amount of noise. Therefore, the neighborhood boundary was taken as 1.0 (see Fig. 7).

4.3. Ablation study

The key to the proposed NEAA method is the representation of neighborhood expectation attribution to features. To emphasize the contribution of the proposed neighborhood attribution method, we conducted ablation experiments to compare the effects of different feature importance estimates on the mobility of adversarial examples. Specifically, we constructed four feature expressions where L_1 uses the raw gradient of the feature Δ_{clean},

$$L_1 = \sum \Delta_{clean} \odot F_i(x), \tag{12}$$

which obtains the gradients directly from the original clean image as feature importance estimates. The formula for the raw gradient is $\Delta_{clean} = \partial y^c / \partial F_i(x)$, where ∂y^c indicates the score for the correct category and $F_i(x)$ denotes the ith feature.

$$L_2 = \sum \Delta_{\mathrm{aggregate}} \odot F_i(x), \tag{13}$$

where $\Delta_{\mathrm{aggregate}}$ denotes the aggregated gradient, which uses the sum of the gradients of multiple copies of the image, which come from randomly dropping pixels from the original image. For the drop probability we use the value in FIA [22].

$$L_3 = \sum \Delta_{IG} \odot (F_i(x) - F'_i(x)), \tag{14}$$

where Δ_{IG} denotes the integral gradient from the black image to the original image.

$$L_4 = \sum \Delta_{EG} \odot (F_i(x) - F'_i(x)) \tag{15}$$

is the proposed neighborhood expectation attribution.

Fig. 8 shows the effectiveness of the four feature estimates for an Inc-v3 attack. The proposed L_4 outperformed the other methods

Table 6
NEAA attacks Transformer-based visual models. The first column is the attacked model. The second column is the recognition rate (%) for clean images.

Model	No-attack	Inc-v3		Resnet-152	
		NEAA	NEAA-PD	NEAA	NEAA-PD
PiT-S [49]	81.9	54.5 (↓ **27.4**)	69.6 (↓ **12.3**)	56.2 (↓ **25.7**)	71.7 (↓ **10.2**)
CaiT-S [50]	85.1	48.8 (↓ **36.3**)	65.4 (↓ **19.7**)	51.8 (↓ **33.3**)	69.0 (↓ **16.1**)
DeiT-B [51]	84.2	42.1 (↓ **42.1**)	55.4 (↓ **28.8**)	42.3 (↓ **41.9**)	59.2 (↓ **25.0**)
Swim-B [52]	86.4	39.0 (↓ **47.4**)	49.6 (↓ **36.8**)	38.8 (↓ **47.6**)	52.9 (↓ **33.5**)
Average loss	–	**−38.3%**	**−24.4%**	**−37.1%**	**−21.2%**

on all models, demonstrating the effectiveness of EG for the accurate representation of important features, and NEAA for improving the transferability of adversarial examples.

4.4. Method generalization ability

We experimented with NEAA and its variants attacking transformer-based visual models, including PiT-S [49], CaiT-S [50], DeiT-B [51] and Swim-B [52]. Table 6 shows the rate of correct recognition on the four models for adversarial examples generated on the source models Inc-v3 and Resnet-v2-152. The proposed NEAA and NEAA-PD remain aggressive in the face of the transformer-based models. The loss of average recognition rate has a maximum of 38.3%. It is worth noting that this experiment revealed an interesting phenomenon, where the enhanced version of NEAA was even less aggressive. The attention mechanism of the transformer-based model captures the global information of the image and introduces positional encoding of the image patches. The PIM and DIM attacks are image transformation-based attacks, which are insensitive to the transformer-based model, instead making NEAA's attack power dispersed. This phenomenon also reflects the weakness of convolution-based models.

5. Conclusion and future work

In this study, we propose the NEAA framework, which destroys key image features to dramatically improve the transferability of generated adversarial examples. NEAA utilizes information about the gradient of an image's neighborhood to detect highly generalized features and extends this transferable insight to intermediate nodes in the neural network. Destroying the key conductance eliminates features that contribute to the category to obtain adversarial examples, and NEAA attacks significantly better on a range of models, with and without defenses. Additionally, the NEAA can be used as a metric to assess model robustness.

Future Work. NEAA provides highly generalized attribution tools designed to capture the key features of an image and new ways of thinking about how the model understands the image. The proposed neighborhood attribution emphasizes image features over source model understanding, transforming the question from "How do models understand images?" to "How should images be understood?". Neighborhood attribution has great potential as an interpretable tool for feature visualization, owing to its powerful ability to describe features. In the field of adversarial attacks, obtaining a more model-light and feature-heavy importance assessment method is feasible for reducing model overfitting and improving the transferability of adversarial examples.

CRediT authorship contribution statement

Wuping Ke: Software, Methodology, Conceptualization. **Desheng Zheng:** Writing – original draft, Data curation. **Xiaoyu Li:** Visualization, Investigation. **Yuanhang He:** Supervision. **Tianyu Li:** Validation, Software. **Fan Min:** Writing – review & editing.

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.

Data availability

I have shared my code in the article in a link.

Acknowledgments

This work was supported in part by the National Key R&D Program of China [No. J2019-V-0001-0092], in part by NSFC, China [U21B2019], in part by Nanchong Municipal Government Universities Scientific Cooperation Project, China [23XNSYSX0062].

References

[1] Y. Xiang, Y. Zhao, S. Deng, Asset-return momentum prediction through pattern recognition, Knowl.-Based Syst. 268 (2023) 110443.
[2] S. Wu, W. Zhang, S. Jin, W. Liu, C.C. Loy, Aligning bag of regions for open-vocabulary object detection, in: CVPR, 2023, pp. 15254–15264.
[3] A.H. Khan, M.S. Nawaz, A. Dengel, Localized semantic feature mixers for efficient pedestrian detection in autonomous driving, in: CVPR, 2023, pp. 5476–5485.
[4] A. Osama, S.I. Gadallah, L.A. Said, A.G. Radwan, M.E. Fouda, Chaotic neural network quantization and its robustness against adversarial attacks, Knowl.-Based Syst. (2024) 111319.
[5] S. Guo, X. Li, P. Zhu, Z. Mu, ADS-detector: An attention-based dual stream adversarial example detection method, Knowl.-Based Syst. 265 (2023) 110388.
[6] C. Guo, J. Gardner, Y. You, A.G. Wilson, K. Weinberger, Simple black-box adversarial attacks, in: ICML, 2019.
[7] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z.B. Celik, A. Swami, Practical black-box attacks against machine learning, in: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, 2017.
[8] A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial examples in the physical world, in: Artificial Intelligence Safety and Security, 2018.
[9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, 2017, arXiv preprint arXiv:1706.06083.
[10] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy, SP, 2017.
[11] 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.
[12] X. Wang, Z. Zhang, K. Tong, D. Gong, K. He, Z. Li, W. Liu, Triangle attack: A query-efficient decision-based adversarial attack, in: ECCV, 2022.
[13] T. Maho, T. Furon, E. Le Merrer, Surfree: A fast surrogate-free black-box attack, in: CVPR, 2021.
[14] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, J. Li, Boosting adversarial attacks with momentum, in: CVPR, 2018.
[15] X. Wang, K. He, Enhancing the transferability of adversarial attacks through variance tuning, in: CVPR, 2021.
[16] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, A.L. Yuille, Improving transferability of adversarial examples with input diversity, in: CVPR, 2019.
[17] Y. Dong, T. Pang, H. Su, J. Zhu, Evading defenses to transferable adversarial examples by translation-invariant attacks, in: CVPR, 2019.
[18] J. Lin, C. Song, K. He, L. Wang, J.E. Hopcroft, Nesterov accelerated gradient and scale invariance for adversarial attacks, in: ICLR, 2019.
[19] J. Zhang, J.-t. Huang, W. Wang, Y. Li, W. Wu, X. Wang, Y. Su, M.R. Lyu, Improving the transferability of adversarial samples by path-augmented method, in: CVPR, 2023, pp. 8173–8182.
[20] X. Wang, X. He, J. Wang, K. He, Admix: Enhancing the transferability of adversarial attacks, in: ICCV, 2021, pp. 16158–16167.
[21] W. Wu, Y. Su, X. Chen, S. Zhao, I. King, M.R. Lyu, Y.-W. Tai, Boosting the transferability of adversarial samples via attention, in 2020 IEEE, in: CVPR, 2020.
[22] Z. Wang, H. Guo, Z. Zhang, W. Liu, Z. Qin, K. Ren, Feature importance-aware transferable adversarial attacks, in: ICCV, 2021.
[23] J. Zhang, W. Wu, J.-t. Huang, Y. Huang, W. Wang, Y. Su, M.R. Lyu, Improving adversarial transferability via neuron attribution-based attacks, in: CVPR, 2022, pp. 14993–15002.
[24] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: ICCV, 2017.
[25] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: ICML, 2017.
[26] K. Dhamdhere, M. Sundararajan, Q. Yan, How important is a neuron, in: ICLR, 2018.

[27] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, 2014, arXiv preprint arXiv:1412.6572.

[28] D. Zheng, W. Ke, X. Li, S. Zhang, G. Yin, W. Qian, Y. Zhou, F. Min, S. Yang, Channel-augmented joint transformation for transferable adversarial attacks, Appl. Intell. (2023) 1–15.

[29] W. Zhou, X. Hou, Y. Chen, M. Tang, X. Huang, X. Gan, Y. Yang, Transferable adversarial perturbations, in: ECCV, 2018.

[30] A. Ganeshan, V. BS, R.V. Babu, Fda: Feature disruptive attack, in: ICCV, 2019.

[31] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: visualising image classification models and saliency maps, in: ICLR, 2014.

[32] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: ICML, PMLR, 2017, pp. 3319–3328.

[33] Z. Liu, Q. Liu, T. Liu, N. Xu, X. Lin, Y. Wang, W. Wen, Feature distillation: Dnn-oriented jpeg compression against adversarial examples, in: CVPR, 2019.

[34] B. Sun, N.-h. Tsai, F. Liu, R. Yu, H. Su, Adversarial defense by stratified convolutional sparse coding, in: CVPR, 2019.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: CVPR, 2009.

[36] J. Zhang, W. Wu, J.-t. Huang, Y. Huang, W. Wang, Y. Su, M.R. Lyu, Improving adversarial transferability via neuron attribution-based attacks, in: CVPR, 2022.

[37] Y. Zhang, Y.-a. Tan, T. Chen, X. Liu, Q. Zhang, Y. Li, Enhancing the transferability of adversarial examples with random patch, in: IJCAI, 2022.

[38] L. Gao, Q. Zhang, J. Song, X. Liu, H.T. Shen, Patch-wise attack for fooling deep neural network, in: ECCV, 2020.

[39] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: CVPR, 2016.

[40] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: AAAI, 2017.

[41] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016.

[42] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.

[43] F. Tramr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, P. McDaniel, Ensemble adversarial training: Attacks and defenses, in: ICLR, 2018.

[44] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, J. Zhu, Defense against adversarial attacks using high-level representation guided denoiser, in: CVPR, 2018.

[45] C. Xie, J. Wang, Z. Zhang, Z. Ren, A. Yuille, Mitigating adversarial effects through randomization, in: ICLR, 2018.

[46] C. Guo, M. Rana, M. Cisse, L. van der Maaten, Countering adversarial images using input transformations, in: ICLR, 2018.

[47] J. Cohen, E. Rosenfeld, Z. Kolter, Certified adversarial robustness via randomized smoothing, in: ICML, 2019.

[48] Y. Liu, X. Chen, C. Liu, D. Song, Delving into transferable adversarial examples and black-box attacks, in: ICLR, 2016.

[49] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, S.J. Oh, Rethinking spatial dimensions of vision transformers, in: ICCV, 2021.

[50] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, H. Jégou, Going deeper with image transformers, in: ICCV, 2021.

[51] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in: ICML, 2021.

[52] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: ICCV, 2021.