Complex & Intelligent Systems (2023) 9:6051–6063
https://doi.org/10.1007/s40747-023-01060-0

ORIGINAL ARTICLE

SGMA: a novel adversarial attack approach with improved transferability

Peican Zhu[1] · Jinbang Hong[2] · Xingyu Li[3] · Keke Tang[4] · Zhen Wang[1,5]

Received: 6 October 2022 / Accepted: 20 March 2023 / Published online: 24 April 2023

Abstract

Deep learning models are easily deceived by adversarial examples, and transferable attacks are crucial because of the inaccessibility of model information. Existing SOTA attack approaches tend to destroy important features of objects to generate adversarial examples. This paper proposes the split grid mask attack (SGMA), which reduces the intensity of model-specific features by split grid mask transformation, effectively highlighting the important features of the input image. Perturbing these important features can guide the development of adversarial examples in a more transferable direction. Specifically, we introduce the split grid mask transformation into the input image. Due to the vulnerability of model-specific features to image transformations, the intensity of model-specific features decreases after aggregation while the intensities of important features remain. The generated adversarial examples guided by destroying important features have excellent transferability. Extensive experimental results demonstrate the effectiveness of the proposed SGMA. Compared to the SOTA attack approaches, our method improves the black-box attack success rates by an average of 6.4% and 8.2% against the normally trained models and the defense ones respectively.

Keywords Deep neural networks · Feature-level attack · Adversarial examples · Transferable attack

Introduction

The last decade has witnessed the rapid development of deep neural networks (DNN), convolutional neural networks (CNN), and their applications in various vision-related tasks such as pedestrian trajectory prediction, image restoration, face recognition, and so forth [1–6]. Despite the impressive progress, prior studies show that deep learning systems are not always reliable and can be easily misled by carefully designed perturbations. Such malicious perturbations are referred to as adversarial noise, and the resulting data (raw data plus perturbation noise) are called adversarial examples or adversarial attacks [7, 8]. The discovery of adversarial examples poses a huge threat to various security- and safety-sensitive applications [9–12].

Adversarial attacks are usually categorized into two types: white-box attacks and black-box attacks, depending on the access of the target model. Briefly, in the white-box attacks, adversaries obtain knowledge of the target model and fabricate adversarial examples accordingly. Therefore, the attack success rates can be very high. By contrast, in black-box attacks where adversaries have partial or no information regarding the target model, we utilize a surrogate model

✉ Keke Tang
tangbohutbh@gmail.com

Peican Zhu
ericcan@nwpu.edu.cn

Jinbang Hong
962139846@qq.com

Xingyu Li
xingyu@ualberta.ca

Zhen Wang
nkzhenwang@163.com

[1] School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University (NWPU), Xi'an 710072, Shaanxi, China

[2] School of Computer Science, NWPU, Xi'an 710072, Shaanxi, China

[3] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada

[4] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, Guangdong, China

[5] School of Cybersecurity, NWPU, Xi'an 710072, Shaanxi, China

with known information to generate adversarial examples. However, since model structures vary, adversarial examples generated by the surrogate model may be low in transferability [13] and thus have poor attack success rates on the target model [14]. Note that the black-box attack scenario is more common in practice, and thus the investigation of high transferable attacks attracts much attention. In literature, methods to generate adversarial examples with high transferability can be categorized into three paradigms: (1) gradient optimization [15, 16]; (2) input transformation [17–20]; and (3) ensemble-model attack [13, 21, 22]. On the other hand, there are also many researches on methods to combat adversarial attacks, such as studying the stability of neural network systems [23–25] or using methods such as adversarial training [26] to improve the robustness of models. In addition, knowledge distillation, denoising, and random smoothing are also explored to promote deep model's robustness against adversarial attacks [27–30].

In adversarial examples generation, perturbing the intrinsic features in raw data usually leads to adversarial examples with high transferability. Such intrinsic features are model-agnostic and contribute to the final decision-making, regardless of the specific model structure. However, such model-agnostic features usually mingle with model-specific patterns in model optimization; thus, identifying these model-agnostic features is non-trivial. Prior studies show that many adversarial attacks are likely to overfit the surrogate model in adversary generation and modify the model-specific features instead of model-agnostic features, reducing the transferability of adversarial examples. To differentiate the intrinsic features from "noisy" model-specific patterns in adversary generation, Wang et al. [31] introduced a feature importance-aware (FIA) strategy where a random pixel-dropping transformation followed by gradient aggregation is employed. The FIA approach then destroys the intrinsic, model-agnostic features, and thus the transferability of the resulting adversarial examples is significantly improved. Nevertheless, there still exists a drawback in FIA. We find that these noisy, model-specific features are not well filtered out because of the high correlations among adjacent image pixels.

To address the afore-mentioned issues, we propose a novel adversarial attack approach, i.e., Split Grid Mask Attack (being referred to as SGMA for simplicity). To generate adversarial examples with improved transferability, we introduce the split grid masks to identify model-agnostic important features in images; thus, we can significantly improve the transferability of adversarial examples by suppressing these important features. During the recognition process, different models tend to focus on diverse discriminative regions; hence, through removing partial information from the image, CNN can learn information that was not sensitive or important. Thus, the problem of overfitting to the source model can be effectively alleviated through aggregating a number of transformed images. Furthermore, the SGM-transformed images preserve spatial structural and textural information that is related with the principal part of the figure. This ensures the model to fluctuate on noise features while simultaneously learn the important features. Specifically, we divide the image into grids of the same size; then in each grid, we generate masks with random positions and random side lengths. While the important features are identified by computing the aggregate gradient of features in the intermediate layers for a set of transformed images. Then, we are anticipated to suppress the influence of important features on the decision of a model; this can be implemented through adding perturbations which can significantly improve the transferability of adversarial examples.

Summing up, the contributions of our manuscript can be elucidated as:

- We propose an adversarial attack method called SGMA that destroys important features of an image by randomly removing some regions of the image. This method distorts the discriminative region of the model, thereby facilitating the transferability of derived adversarial examples.
- We propose a new image processing method based on information removal—Split Grid Mask, the SGM transformation enables DNN to extract more features to alleviate the overfitting of adversarial examples to the source model. Furthermore, the aggregated gradient of the SGM-transformed images effectively highlights important features, which can lead to higher attack success rates of black-box attacks.
- Extensive experiments are performed on different classification models. Compared with SOTA transferable attack methods, our proposed SGMA is demonstrated to be superior while the derived adversarial examples are of excellent transferability.

The schematic of this manuscript is provided as follows. The "Related works" section reviews related works. In the "Methodology" section, we illustrate the proposed approach, i.e., SGMA, in detail. Extensive experiments and the corresponding results are provided in the "Experimental analysis" section. "Conclusion" concludes this paper and discusses some future works.

Related works

In [32], Szegedy et al. first proposed the concept of adversarial examples and demonstrated the security issues of deep learning models. Since then, derivation of adversarial examples has received numerous attention. Under the black-box attack setting, inaccessibility of information of the target model encourages investigating transfer-based attacks. In this regard, many methods are proposed for adversarial attacks with high transferability [33–35].

Among these methods, gradient-based black-box attacks is of great importance. They first adopted a well-trained model as the surrogate source model for adversarial examples generation. Then, these adversarial examples are applied to the targeted model for attacks. In [14], Goodfellow et al. introduced the Fast Gradient Sign Method (FGSM), where perturbations overlay onto the original images along the converse direction of the gradient for adversarial examples generation. Later, Kurakin et al. utilize the FGSM iteratively (I-FGSM for short) to promote the success rates of white-box attacks. Nevertheless, as illustrated in [36], the transferability of derived adversarial examples might be limited as these methods are prone to perturbing model-specific features and thus falling into local optimization and low transferable adversarial examples.

To improve the transferability of adversarial attacks, Dong et al. proposed to integrate I-FGSM and the momentum method (MI-FGSM for short) in adversary generation [16]. In this method, the gradient vector from previous iterations is accumulated to guide the calculation of the next gradient, helping escape from the local optimal in adversary optimization and thus improving their transferability. Later, Lin et al. proposed a new approach through replacing momentum with Nesterov Accelerated Gradient, i.e., NI-FGSM [15]. Before calculating the gradient in each iteration, NI-FGSM makes a prediction in the direction of the previously accumulated gradient. Thanks to this looking ahead property, it is much easier and faster to escape from the local optimum. Then, Wang et al. proposed VMI-FGSM through adjusting the iterative gradient according to the gradient variance [37]. When computing the gradient at each iteration, the current gradient is no longer directly used for momentum accumulation. In addition, the gradient variance of previous iteration is also considered for the adjustment of the current gradient.

Input transformation are also explored to promote the transferability of adversarial examples. Inspired by the fact that data augmentation can alleviate the overfitting problem in model optimization, Xie et al. proposed Diverse Input Method (DIM) to improve the transferability through using input transformation [17]. Briefly, gradient are computed on data crafted from random resizing and random padding of the clean image with a certain probability. Later, Dong et al. presented the Translation-Invariant Method (TIM) in which the gradient of the original image is replaced with the average gradient of a set of translated images [18]. Later, Lin et al. considered the scale invariance of CNN and then proposed an adversarial example generation algorithm, i.e., Scale-Invariant Method (SIM) [15]. In SIM, the generation of perturbations is guided by computing the average gradient of a set of scaled images. Moreover, Zhang et al. proposed the Admix Attack Method (Admix) to introduce information from other categories of images [19]. Specifically, this approach first randomly selects images of other classes and then mixes them with the original images; thus, the average gradient of the mixed images is utilized to update the perturbation. On the basis of information deletion, Hong et al. developed an adversarial example algorithm, named Grid Mask Attack (GM-Attack) [38]. Here, proper information removal can ensure the CNN to extract more features; thus, the problem of adversarial examples overfitting the source model can be effectively alleviated by GM-Attack.

The afore-mentioned methods mainly focus on attacking the last layer of CNN. It is intuitive to ask the question: Whether it is also effective to attack the features of middle layers? Various scholars devote their efforts to tackling this question [39, 40]. Zhou et al. presented the Transferable Adversarial Perturbations (TAP) to improve the transferability through increasing the feature distances between the original image and its adversarial example in the intermediate layers [41]. Later, Ganeshan et al. proposed the Feature Disruptive Attack (FDA) [42]. In FDA, the features of each layer in the model are corrupted which eventually leads to wrong classification results. Inkawhich et al. modifies the features of the original image such that the resulting middle layer representation is closer to an image from another class, thereby improving the transferability of adversarial examples [43]. To further promote the transferability, Huang et al. proposed the Intermediate Level Attack (ILA), aiming to fine-tune the previously generated adversarial examples through increasing their perturbations to pre-specified layers in the source model [44].

Recently, differentiation of model-agnostic features and model-specific features in adversarial robustness is proposed. Specifically, a deep learning model might extract noisy features that adapt to its structure. However, these model-specific patterns may not be utilized by other models with different structures. Hence, perturbing these features is of limited significance in improving the transferability of adversarial examples. Therefore, it is very important to identify and attack the important features that can dominate the decisions of different models. For instance, Wang et al. presented the Feature Importance-aware Attack (FIA) [31]. Different from previous methods that modify all features indiscriminately, FIA distinguishes the model-specific and model-agnostic, intrinsic features through gradient aggregating and focuses on perturbation of those important features, which improve adversarial examples' transferability. Similar to FIA, Zhang et al. proposes the Random Patch Attack (RPA) [45], where random patch transformation is used to identify model-agnostic features for enhancing attack transferability. To further improve the transferability of adversarial attacks, this study proposes a novel method namely SGMA. It utilizes aggregate gradient to differentiate features and selects important features to attack. Details of the proposed SGMA will be illustrated explicitly later.

Methodology

In this section, we illustrate the proposed SGMA explicitly with the overall architecture being provided in Fig. 1. Firstly, we propose the SGM-transformed approach to handle the pre-processing of initial input images while a variant, i.e., SGMR-transformed approach, is also provided. Then, details of the adopted gradient aggregating model are provided. After that, we present the overall architecture to illustrate the process of our SGMA.

SGM-transformed image-processing approach

Given an input image m, we randomly generate some grid areas and then remove the corresponding pixels inside the grids similar as in [38]. Specifically, we first specify a matrix M with the elements being filled with 1. The size of M is the same as the input image m. Then, the matrix M will be divided into a number of grids , where the side length of each grid is d. In each grid, we generate a square mask with a side length of $d * r$ where r denotes the side length keep-ratio of the mask; and we set the value of pixels inside the mask to 0. Let the relative positions of each mask to the upper left corner of the grid be represented as ψ_x and ψ_y respectively. Then, the generated masked image can be expressed as:

$$\tilde{m} = m \odot \text{Mask}\left(d, r, \psi_x, \psi_y\right) \tag{1}$$

where $\odot$ denotes the element-wise product.

SGM-transformed approach

In this study, we introduce a novel input-transformation approach, i.e., Split Grid Mask-transformed (being referred to as SGM-transformed for simplicity), to incorporate randomness in adversarial example generation process to improve its transferability. Different from the GM-Attack [38] where the derived masks are uniformly distributed, we propose to assign a random offset (s_x, s_y) to the mask in each grid. s_x and s_y follow the uniform distribution between $(-S, S)$, where S represents the maximum offset. Furthermore, the side length of each mask also changes randomly, following the uniform distribution between $(-C, C)$, where C denotes the maximum varying of the side length. Thus, the final generated masked image can be represented as:

$$\tilde{m} = m \odot \text{Mask}\left(d, r, \psi_x, \psi_y, S, C, p\right), \tag{2}$$

where p represents the occurrence probability of side length varying.

SGMR-transformed approach

We further improve the proposed SGM-transformed approach by incorporating mask random rotation. In this variant of SGM-transformed approach, after the generation of masks in each grid, we introduce a rotating operation to the obtained masks, i.e., rotating all masks clockwise by the same angle of θ. We assume the rotation angle follows a pre-defined distribution such as the uniform distribution between 0 and 90°. The proposed variant is denoted as SGMR-transformed approach with the overall formula being represented as:

$$\tilde{m} = m \odot \text{Mask}\left(d, r, \psi_x, \psi_y, S, C, p, \theta\right). \tag{3}$$

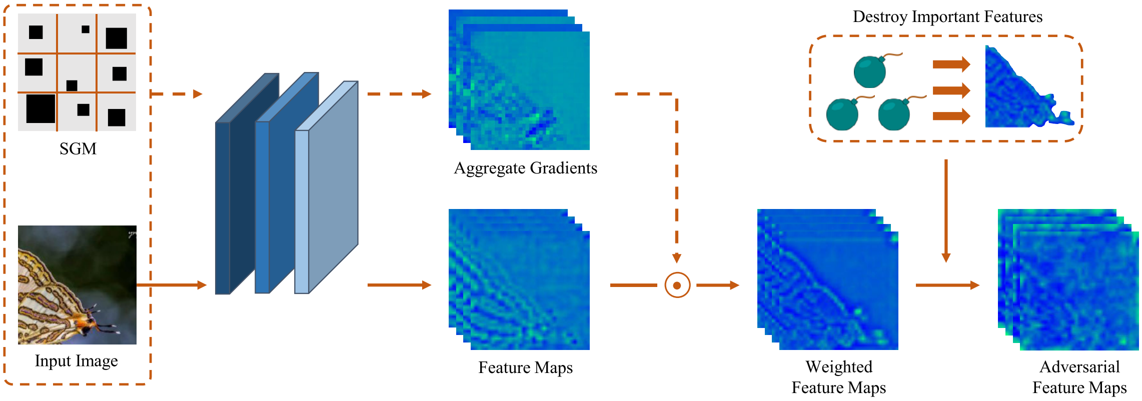


Fig. 1 The overall architecture of our SGMA approach. The input image is pre-processed by our proposed SGM transformation. The set of transformed images are fed to the surrogate model and image features are weighted by the resulting aggregated gradients. The proposed SGMA is able to perturb features with higher weights

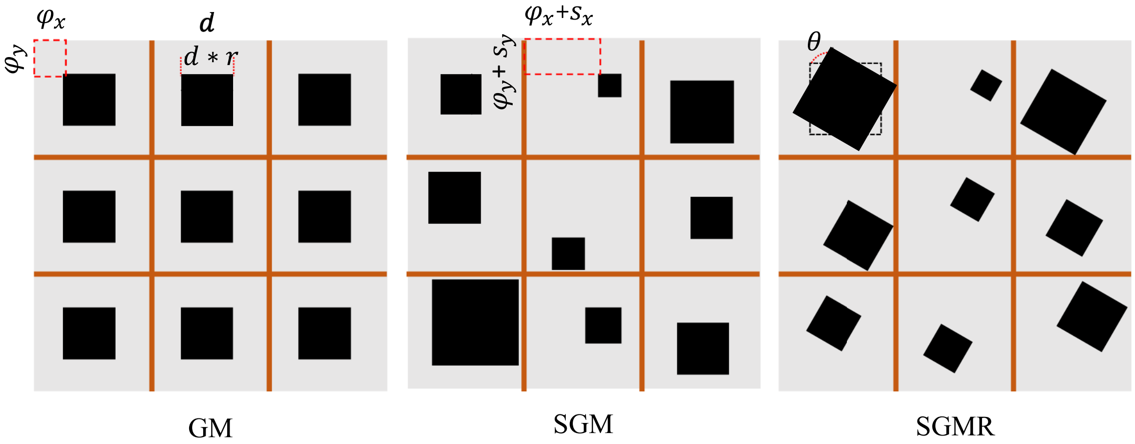


Fig. 2 An illustrative example of the SGM-transformed and SGMR-transformed approaches to pre-process the images. Meanings of the parameters are provided in the main text

After generating masks using Eq. (2) or Eq. (3), the input images are pre-processed, generating a set of masked images for downstream gradient aggregating and feature significance estimation. For better interpretation, an illustrative example of the mask generation process is provided in Fig. 2.

Gradient aggregating

Aggregate gradient upon the SGM- and SGMR- transformed images is capable of highlighting the object-related features meanwhile attenuating the noisy, model-specific features. From this perspective, it facilitates to boost the transferability of generated adversarial examples. We illustrate the gradient aggregating process adopted in this work in Fig. 3.

Let f denote the source/surrogate model. For a given input m, the corresponding feature maps of the kth layer can be denoted as $f_k(m)$. Here, we can adopt the gradient to reflect the relative importance of features:

$$\Delta_k^m = \frac{\partial l\left(m, y^{\text{real}}\right)}{\partial f_k(m)}, \tag{4}$$

where $l(.,.)$ represents the logit output corresponding to the true label y^{real}.

Note that the gradient on the original image usually does not highlight the object-related features very well. To identify and highlight significant features, we propose to calculate the aggregate gradient from the set of masked images by the proposed SGM-transformed/SGMR-transformed approach. The corresponding aggregate gradient is calculated as

$$\Delta_k = \frac{1}{D} \sum_{n=1}^{ens} \frac{\partial l\left(\tilde{m}, y^{\text{real}}\right)}{\partial f_k\left(\tilde{m}\right)}, \tag{5}$$

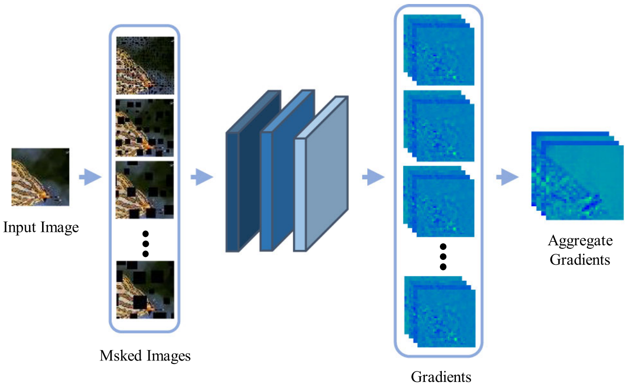


Fig. 3 Presentation of the gradient aggregating process

where *ens* represents the number of masked images, $\tilde{m}$ denotes the masked image generated by the SGM-transformed/SGMR-transformation, and D indicates the l_2-norm of the corresponding summation term.

Proposed SGMA approach

As illustrated in Fig. 1, after obtaining the aggregate gradients from the masked images, we obtain the weighted feature maps of the input image. Based on the weighted feature maps, important features can be highlighted accordingly. During adversarial example generating process, the following loss function is used to represent the important features:

$$L\left(m^{\text{adv}}\right) = \Delta_k \odot f_k\left(m^{\text{adv}}\right), \tag{6}$$

where m^{adv} represents the adversarial example corresponding to the input image m. In Eq. (6), the important features towards the decision-making are highlighted by larger aggregate gradients. Thus, through decreasing the higher intensities, we can effectively suppress the influence of important features on the final decision of a model, thus leading the generation of adversarial examples in a much more transferable direction. The optimization objective of our proposed SGMA can be formulated as:

$$\arg\min_{m^{\mathrm{adv}}} L\left(m^{\mathrm{adv}}\right) \tag{7}$$

$$\text{s.t.,} \quad \left\| m^{\mathrm{adv}} - m \right\|_{\infty} \leq \epsilon, \tag{8}$$

where ϵ represents a pre-specified threshold while the term $\left\| m^{\mathrm{adv}} - m \right\|_{\infty} \leq \epsilon$ indicates that the difference between m^{adv} and m must be smaller than ϵ.

We utilize the momentum optimization to solve Eq. (7) and the detailed processes of the proposed SGMA approach are specified in Algorithm 1 (lines 10–13). Furthermore, a variant SGMRA can be derived accordingly by replacing the SGM-transformed image processing approach with the SGMR-transformed one [i.e., utilizing Eq. (3) instead of Eq. (2) at the 4-th step].

Experimental analysis

Experimental setup

Dataset description

In this manuscript, without loss of generality, we also select the dataset from a widely adopted ImageNet-compatible dataset [46]. The surrogate models are well trained with high classification accuracy.

Models for evaluation

To validate the transferability of adversarial examples derived by different methods, several vanilla trained models are regarded as the target to be attacked including Inception-v3 [47], Inception-v4 [48], Inception-Resnet-v2 [48], ResNet-V1-50 [49], ResNet-v1-152 [49], VGG16 [50], VGG19 [50]; for simplicity, the afore-mentioned models are abbreviated as Inc-v3, Inc-v4, IncRes-v2, Res-50, Res-152, Vgg-16 and Vgg-19 respectively. Moreover, we also consider five adversarially trained models, i.e., Adv-Inc-v3, Adv-IncRes-v2, Inc-v3$_{\mathrm{ens3}}$, Inc-v3$_{\mathrm{ens4}}$ and IncRes-v2$_{\mathrm{ens}}$ [26].

Comparison baselines

To illustrate the performance, we consider the following approaches, including DIM, TIM, PIM, FIA, RPA, as well as their combinations, FIA + PIDIM, FIA + PIDITIM, RPA + PIDIM, RPA + PIDITIM.

Algorithm 1 The SGMA Algorithm.

```
Require: Input image $m$, true label $y^{\mathrm{real}}$, classifier $f$ with loss function $L$;
    Perturbation size $\epsilon$, number of iterations $T$, the decay factor $\mu$;
    Length of the grid $d$, side length keep-ratio $r$, maximum offset $S$;
    Maximum varying of the side length $C$, the number of masks $ens$;
Ensure: An adversarial example $m^{\mathrm{adv}}$;
1: Initialization $\alpha = \epsilon/T$; $m_0^{\mathrm{adv}} = m$; $g_0 = 0$; $\widetilde{g}_0 = 0$;
2: Calculate the aggregate gradients:
3: for $i = 0 \rightarrow ens$ do
4:     Derive the masked image by adopting Eq. (2);
5:     Determine the gradient of the masked image:
           $\Delta_i = \dfrac{\partial l(\tilde{m}, y^{\mathrm{real}})}{\partial f_k(\tilde{m})}$
6:     Update $\Delta$:
           $\Delta = \Delta + \Delta_i$
7: end for
8: Obtain the aggregate gradients:
       $\Delta_k = \Delta / \|\Delta\|_2$
9: Derive the loss through utilizing Eq. (6);
10: for $t = 0 \rightarrow T - 1$ do
11:     Update $g_t$:
            $g_{t+1} = \mu \cdot g_t + \dfrac{\nabla_{\boldsymbol{m}} L\left(\boldsymbol{m}_t^{\mathrm{adv}}\right)}{\left\|\nabla_{\boldsymbol{m}} L\left(\boldsymbol{m}_t^{\mathrm{adv}}\right)\right\|_1}$
12:     Update $m_{t+1}^{\mathrm{adv}}$:
            $m_{t+1}^{\mathrm{adv}} = m_t^{\mathrm{adv}} + \alpha \cdot \mathrm{sign}\left(g_{t+1}\right)$
13: end for
14: return $m^{\mathrm{adv}} = m_T^{\mathrm{adv}}$.
```

Hyper-parameters

To be in consistent, we adopt the attack parameter settings in [31]: the maximum perturbation ϵ is set to 16, the number of iteration T is assigned to 10, and the step size α equals to 1.6. For DIM, the transformation probability p is 0.7. While for TIM, the adopted kernel size is 15. For FIA, we assign the ensemble number N to 30. As to the drop probability p_d, the value is set to 0.3 and 0.1 when attacking normally trained models and defense ones respectively. For RPA, we set the ensemble number N to 60, the modify probability p_m is set to 0.3 and 0.2 when attacking normally trained models and defense ones respectively. For the proposed SGMA, the ensemble number ens equals to 30, the length of the grid d will be randomly selected in the range of [3, 105]. The side length keep-ratio r is 0.5 and 0.6 for the normally training model and defense one respectively, the maximum offset S and the maximum varying of the side length C will vary

Table 1 Performance (indicated by attack success rates) of different methods of attacking a normally trained model

Model	Method	Inc-v3	Inc-v4	IncRes-v2	Vgg-16	Vgg-19	Res-50	Res-152
Inc-v3	DIM	0.996*	0.646	0.596	0.476	0.464	0.407	0.363
	PIM	0.979*	0.558	0.515	0.616	0.605	0.533	0.463
	FIA	0.983*	0.835	0.806	0.714	0.733	0.704	0.649
	RPA	0.986*	0.857	0.840	0.753	0.756	0.727	0.684
	SGMA	0.996*	0.929	0.908	0.871	0.875	0.846	0.797
	PIDIM	0.981*	0.705	0.664	0.392	0.192	0.618	0.390
	FIA + PIDIM	0.988*	0.878	0.857	0.824	0.841	0.797	0.744
	RPA + PIDIM	0.985*	0.896	0.887	0.837	0.848	0.830	0.798
	SGMA + PIDIM	**0.997***	**0.936**	**0.934**	**0.922**	**0.930**	**0.898**	**0.858**
IncRes-v2	DIM	0.752	0.713	0.971*	0.515	0.514	0.509	0.437
	PIM	0.668	0.629	**0.996***	0.645	0.635	0.562	0.508
	FIA	0.811	0.775	0.892*	0.714	0.714	0.718	0.689
	RPA	0.705	0.668	0.792*	0.641	0.630	0.643	0.587
	SGMA	0.932	0.892	0.988*	0.866	0.866	0.842	0.801
	PIDIM	0.805	0.780	0.985*	0.625	0.626	0.566	0.501
	FIA + PIDIM	0.842	0.797	0.916*	0.806	0.799	0.790	0.784
	RPA + PIDIM	0.762	0.758	0.825*	0.762	0.753	0.758	0.729
	SGMA + PIDIM	**0.949**	**0.929**	0.992*	**0.914**	**0.937**	**0.913**	**0.882**
Res-152	DIM	0.803	0.722	0.726	0.884	0.880	0.950	0.999*
	PIM	0.660	0.564	0.511	0.832	0.825	0.923	**1.000***
	FIA	0.853	0.811	0.778	0.915	0.915	0.968	0.995*
	RPA	0.896	0.858	0.857	0.943	0.944	0.979	0.996*
	SGMA	0.932	0.873	0.881	0.954	0.961	0.991	**1.000***
	PIDIM	0.822	0.766	0.770	0.912	0.899	0.967	0.998*
	FIA + PIDIM	0.903	0.859	0.856	0.958	0.957	0.982	0.995*
	RPA + PIDIM	0.943	0.905	0.912	0.976	0.977	0.985	0.997*
	SGMA + PIDIM	**0.961**	**0.925**	**0.914**	**0.980**	**0.981**	**0.993**	**1.000***
Vgg-16	DIM	0.872	0.870	0.809	0.998*	0.989	0.920	0.878
	PIM	0.841	0.820	0.756	**1.000***	0.989	0.911	0.859
	FIA	0.957	0.956	0.923	0.998*	0.996	0.973	0.953
	RPA	0.963	0.972	0.951	0.999*	0.999	0.976	0.960
	SGMA	0.981	0.979	0.955	**1.000***	**1.000**	0.984	0.974
	PIDIM	0.891	0.895	0.847	0.999*	0.988	0.938	0.908
	FIA + PIDIM	0.976	0.975	0.938	0.998*	0.998	0.982	0.964
	RPA + PIDIM	0.982	0.982	**0.966**	**1.000***	0.999	**0.991**	0.980
	SGMA + PIDIM	**0.984**	**0.983**	0.961	**1.000***	**1.000**	0.987	**0.983**

Bold values represents the best performance

The first column illustrates source models, and the first row enumerates target models. Our methods are SGMA and SGMA + PIDIM. $*$ represents the values obtained for white-box attacks

within d. Our choice of layers to be attacked is the same as that in [31].

Comparison of transferability

In this section, aiming to illustrate the superiority of SGMA approach, extensive experiments are conducted with the transferability of the obtained adversarial examples being compared with those derived by the considered baseline approaches. Furthermore, we consider applying attacks against two types of models, i.e., vanilla trained models and adversarially trained ones.

Table 2 Performance (indicated by attack success rates) of different methods of attacking a defense model

Model	Method	Adv-Inc-v3	Adv-IncRes-v2	Ens3-Inc-v3	Ens4-Inc-v3	Ens-IncRes-v2
Inc-v3	TIM	0.320	0.264	0.301	0.325	0.224
	PIM	0.343	0.302	0.333	0.384	0.262
	FIA	0.545	0.549	0.439	0.420	0.235
	RPA	0.590	0.595	0.455	0.443	0.263
	SGMA	0.617	0.660	0.482	0.483	0.278
	PIDITIM	0.416	0.339	0.431	0.473	0.314
	FIA + PIDITIM	0.648	0.590	0.625	0.632	0.509
	RPA + PIDITIM	0.721	0.656	0.696	0.706	0.592
	SGMA + PIDITIM	**0.745**	**0.665**	**0.709**	**0.723**	**0.604**
IncRes-v2	TIM	0.400	0.435	0.395	0.415	0.384
	PIM	0.390	0.353	0.394	0.422	0.328
	FIA	0.549	0.568	0.469	0.447	0.374
	RPA	0.605	0.641	0.552	0.543	0.411
	SGMA	0.690	0.746	0.623	0.551	0.457
	PIDITIM	0.538	0.552	0.547	0.545	0.506
	FIA + PIDITIM	0.551	0.529	0.549	0.562	0.506
	RPA + PIDITIM	0.660	0.657	0.646	0.651	0.614
	SGMA + PIDITIM	**0.767**	**0.746**	**0.746**	**0.754**	**0.672**
Res-152	TIM	0.415	0.375	0.431	0.476	0.341
	PIM	0.407	0.389	0.469	0.518	0.388
	FIA	0.701	0.667	0.614	0.603	0.417
	RPA	0.737	0.710	0.669	0.653	0.461
	SGMA	**0.864**	**0.799**	**0.821**	**0.798**	**0.725**
	PIDITIM	0.519	0.490	0.586	0.648	0.479
	FIA + PIDITIM	0.663	0.625	0.696	0.727	0.614
	RPA + PIDITIM	0.711	0.695	0.744	0.756	0.672
	SGMA + PIDITIM	0.721	0.649	0.732	0.768	0.653
Vgg-16	TIM	0.528	0.462	0.551	0.553	0.416
	PIM	0.519	0.432	0.502	0.563	0.399
	FIA	0.878	0.863	0.856	0.860	0.708
	RPA	0.903	0.879	0.877	0.864	0.733
	SGMA	**0.959**	**0.928**	**0.941**	**0.942**	**0.902**
	PIDITIM	0.510	0.446	0.556	0.607	0.431
	FIA + PIDITIM	0.747	0.714	0.773	0.801	0.670
	RPA + PIDITIM	0.780	0.738	0.807	0.837	0.701
	SGMA + PIDITIM	0.820	0.750	0.832	0.851	0.732

Bold values represents the best performance

The first column illustrates source models, and the first row enumerates target models. Our methods are SGMA and SGMA + PIDIM

Attacking vanilla trained models

Here, we incorporate Inc-v3, IncRes-v2, Res-152, Vgg-16 as the source/surrogate models and then attacks are applied to all vanilla trained models. We do not choose TIM for comparison because it is designed for the defense models. The corresponding simulation results are presented in Table 1. As indicated, the attack success rates of our method in the black-box setting are increased by an average of 6.4% over the other state-of-the-art methods. Especially for adversarial examples generated on Inc-v3 and IncRes-v2, our method can improve the transferability by over 10%. Furthermore, our attack approach can also be effectively combined with other attack approaches to further improve the transferability. When our method is combined with PIDIM (i.e., SGMA + PIDIM) craft adversarial examples on Res-152 and Vgg-16, the average attack success rates on all the models are above 95%. Compared with other feature-level attack approaches, the

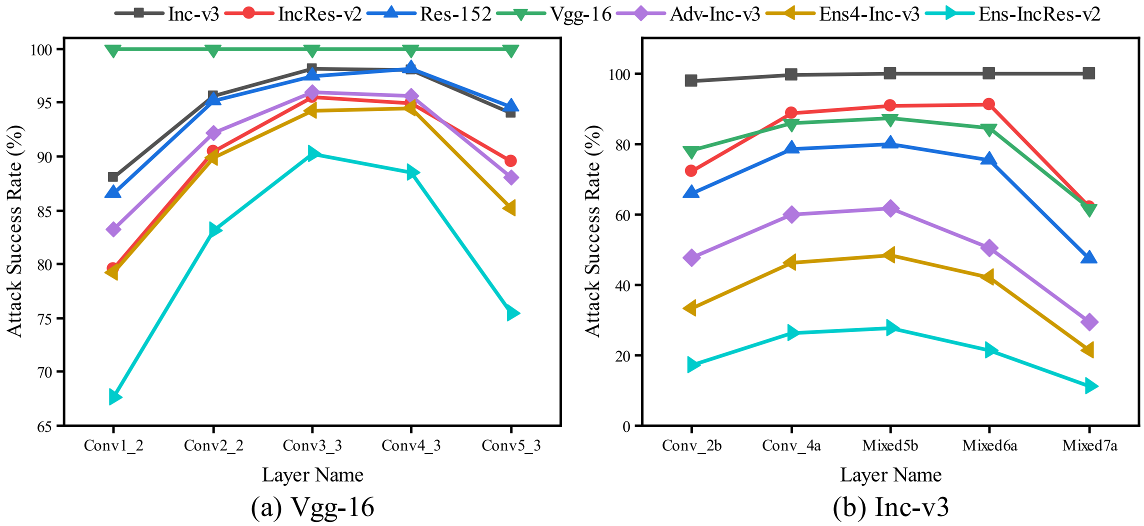


Fig. 4 Effects of layer selection in the source model on the transferability of derived adversarial examples. Here, the generated adversarial examples on Vgg-16 or Inc-v3 are utilized to attack different target models

performance of our method is superior in both white-box and black-box scenarios. To improve the transferability, FIA and RPA sacrifice certain level of performance regarding some white-box attacks; especially in IncRes-v2 model, the corresponding attack success rates are reduced to 89.2% and 80% respectively. However, the white-box success rates of SGMA approach are all higher than 99%.

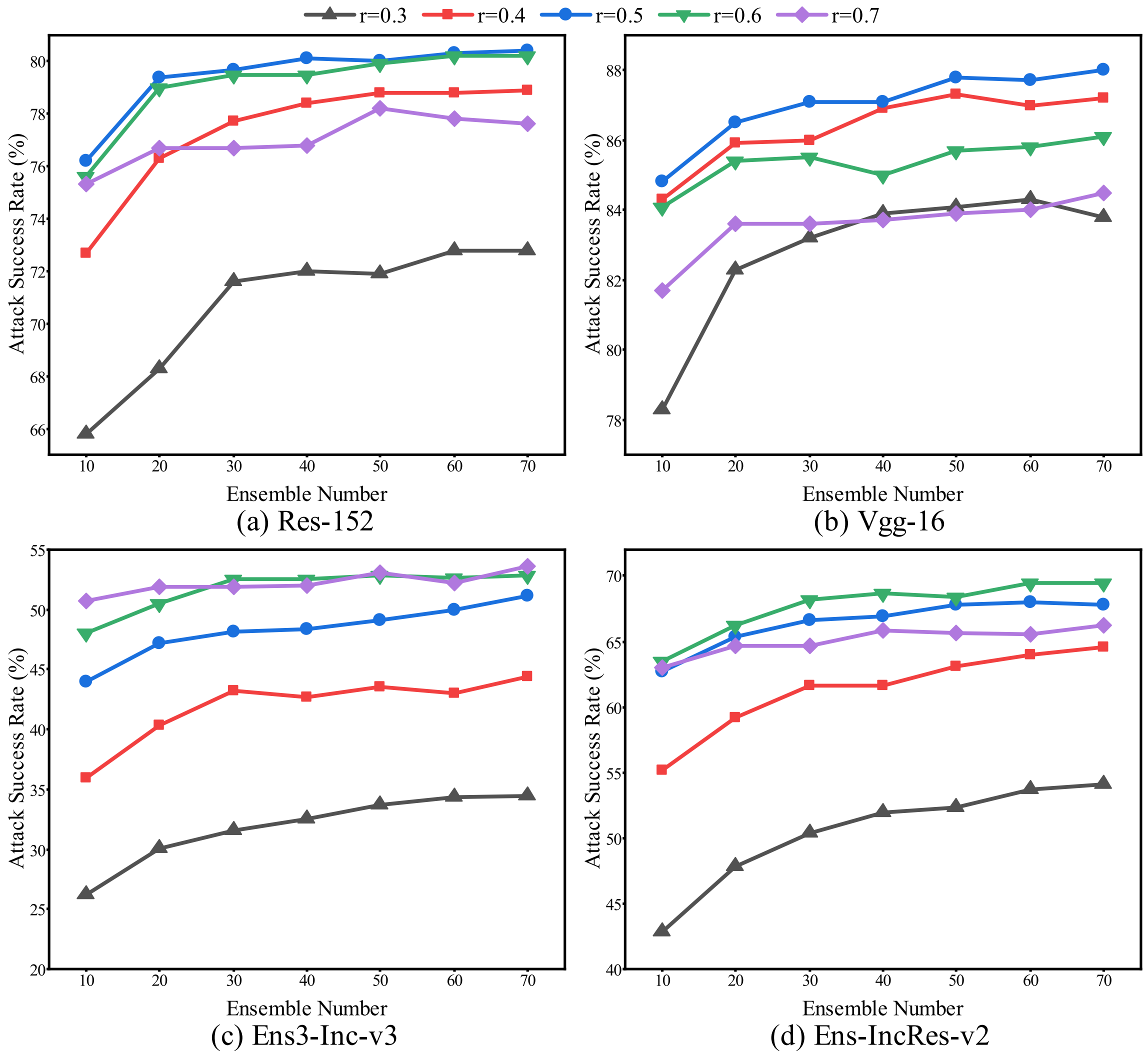
Attacking adversarially trained models

Then, we consider the attacks against defense models. The corresponding results are provided in Table 2 which are also compared with those obtained by considered baselines. Since the defense models are usually robust to adversarial examples, the derived attack success rates are likely to decrease compared with those obtained for normally trained ones. However, as indicated, our approach still significantly outperforms the other SOTA attack approaches. We find that our approach is able to increase the black-box attack success rate by an extent of 8.2% on average. Especially when the adopted source model is Res-152, the attack success rate of SGMA approach on Ens-IncRes-v2 is approximately 26% higher than the previous best result obtained by RPA. When generating adversarial examples on Vgg-16, the attack success rates of SGMA exceed 90% on all the considered defense models; specially, the success rate against Adv-Inc-v3 is increased to 95.9%. As to the combined methods, when attacking Inc-v3 and IncRes-v2, the SGMA + PIDITIM approach can further improve the transferability by 18.52% and 12.36% on average respectively.

Varying layer selection for the source model

The selection of feature layer is a key factor which plays an important role in affecting the performance of feature-level attacks. For DNNs, early layers usually extract only low-level features while data-specific feature sets are still under-constructing. Once enough features are extracted to model the data, later layers will combine and optimize low-level features to form high-level ones aiming to improve the eventually classification accuracy [51]. Therefore, the early layers have not learned the semantic information and important features related with the object, whereas the later layers are likely to extract too many noise features. Thus, we target to find the layer that has learned the semantic information and important features related with the object sufficiently while not too many noise features are incorporated simultaneously. In fact, the features of the middle layer can avoid the shortcomings of insufficient features and high correlation with the model; hence, the selection of appropriate middle layer might lead to promoted transferability.

For simplicity, we just adopt Vgg-16 and Inc-v3 for illustration while the corresponding results are illustrated in Fig. 4a, b, respectively. As revealed, our experiments on Vgg-16 and Inc-v3 prove the afore-conclusion regarding the selection of middle layers. We find that for the considered two models, while there usually exists an optimal layer corresponding to a maximum attack success rate. This indicates that adversarial examples obtained by attacking the middle layer are likely to be of higher transferability compared with those being generated by attacking other layers.



(a) Res-152

(b) Vgg-16

(c) Ens3-Inc-v3

(d) Ens-IncRes-v2

Fig. 5 Effect of varying the side length keep-ratio and the ensemble number on the attacking performance. The adversarial examples are generated by SGMA with different parameter setting and we choose Inc-v3 as the source model. The side length keep-ratio varies from 0.3 to 0.7 with an increment of 0.1, while the ensemble number changes from 10 to 70 with a step of 10

Effects of varying *r* and *ens*

In this section, we explore the effects of varying the other hyper-parameters on the final performance of the proposed SGMA approach. Here, we mainly consider two hyper-parameters, i.e., the keep-ratio of the side length (r) and the ensemble number (ens); while the success rates under the black-box attack for different scenarios are derived with the corresponding results being provided in Fig. 5. For simplicity, we adopt the Inc-v3 as the source model and then adversarial examples are derived by setting different r and ens. Here, we suppose r varies from 0.3 to 0.7, and the increment is 0.1. For each r, ens is iterated from 10 to 70 with a step size of 10. Four models, i.e., Res-152, Vgg-16, Ens3-Inc-v3, and Ens-IncRes-v2, are considered as the target model to be attacked by the generated adversarial examples while the transferability is reflected by the attack success rate.

In practice, a small r will remove too much information from the input image, which may result in the failure of extracting object-related features; whereas a large r retains too much information which might limit the performance of the method. This can be reflected by the obtained results

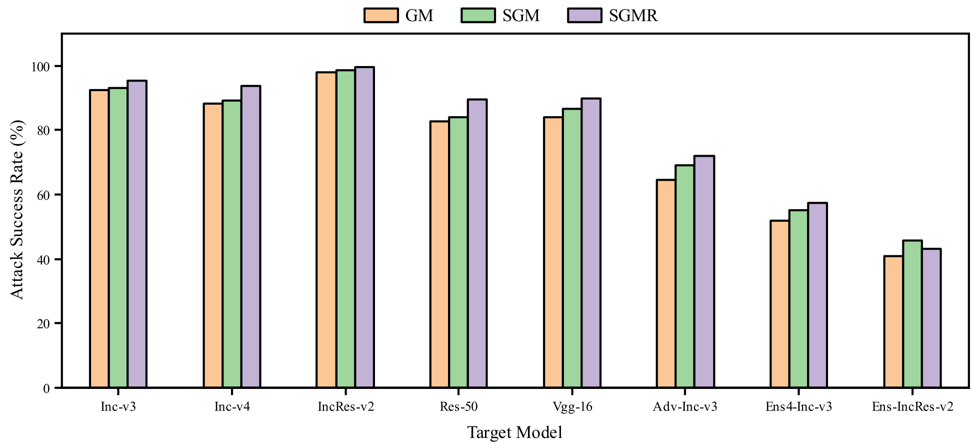


Fig. 6 Attack success rates (%) derived when attacking eight models with adversarial examples being crafted by GM-Attack, SGMA, and SGMRA on IncRes-v2 model

being presented in Fig. 5. As reflected, the optimal r is around 0.5 when attacking the considered normally trained models (Res-152 and Vgg-16 are considered for simplicity); while the value is around 0.6 if attacking the considered defense models (Ens3-Inc-v3 and Ens-IncRes-v2 are incorporated). Generally speaking, the attack success rates increase with the increase of ensemble number ens. Nevertheless, when ens is large enough, the success rate tends to saturate and may even decline. Hence, for the parameter setting for the SGMA approach, we set $ens = 30$, while r is assigned to be 0.5 and 0.6 for the normally training model and defense one respectively.

Comparison of GM-attack versus SGMA versus SGMRA

The aggregate gradients reduce the intensity of the noise feature by taking advantage of its vulnerability to image transformation. Therefore, increasing the sampling space of the transformed images can help highlight important features. Similar as GM-Attack, SGMA and SGMRA are subject to the image transformation category based on information deletion. All these approaches are aiming to achieve a good balance between deleting information and retaining information, which can help DNNs to extract sufficient features. Whereas the difference lies in that SGMA and SGMRA can efficiently increase the randomness of image transformation; thus, we can efficiently expand the example space of transformed images. Therefore, they further highlight the important features of the image which plays an important role in guiding the generation of adversarial examples.

Aiming to valid this inherently, we perform experiments through attacking different models while the IncRes-v2 is adopted as the source model; the corresponding results are provided in Fig. 6. As indicated, we find that the attack success rates of GM-Attack, SGMA and SGMRA increase gradually while the performances of SGMA and SGMRA are always much better than that of GM-Attack. For the majority of considered scenarios, the transferability of the adversarial examples crafted by SGMRA is higher than that obtained through SGMA being indicated by larger attack success rates.

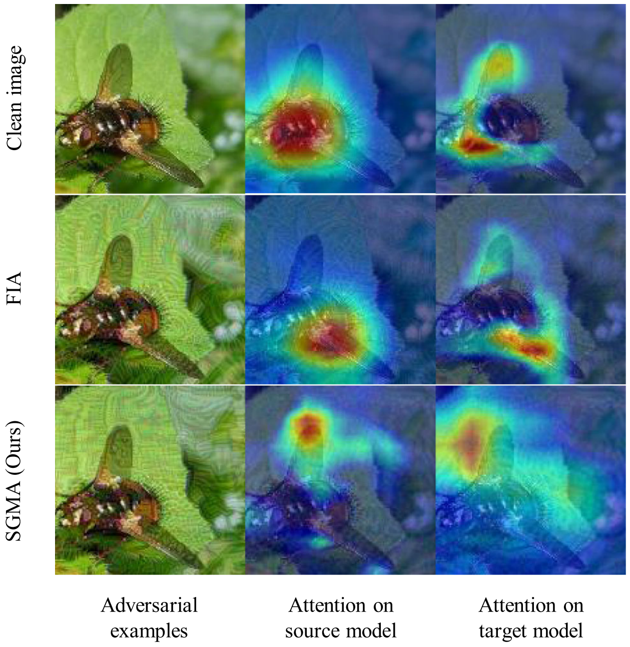


Fig. 7 Comparison of attention regions derived by FIA and SGMA. Adversarial images are generated on the source model (VGG16 is considered) and utilized to attack the target model (Inception-V3 is adopted). Our SGMA reduces the model's ability to capture important features of objects and focuses on completely irrelevant regions instead; whereas the model's attention to adversarial example generated by FIA partially overlaps with that on the clean image

Capability of distorting attention region

To visualize the capability of adversarial examples generated by our proposed SGMA in distorting the model's attention, we present the corresponding results in Fig. 7; while the corresponding results for the adversarial examples derived by FIA are also provided.

We find that the corresponding attention region of the model given the adversarial example is completely different from that on the clean image. In contrast, the model's attention to adversarial examples generated by FIA partially overlaps with that on clean image. Whereas for the SGMA, the two regions are totally different; this illustrates the effectiveness of the split grid mask transform in capturing those important features.

Conclusion

In this paper, we propose a novel Split Grid Mask attack (SGMA), which can generate adversarial examples with higher transferability. The SGM-transform can alleviate the overfitting problem by randomly removing some discontinuous regions so that the model can extract more features. Using the aggregate gradients of the SGM transformed image can reduce the intensity of model-specific features and effectively highlight important object-related features of the input images. Perturbing these important features guides the development of adversarial examples in a more transferable direction. As demonstrated by the experimental results, compared with the SOTA transfer-based attack approaches, SGMA achieves higher success rates when attacking both the normal training model and the defense model. Although our algorithm improves the transferability of adversarial examples, there still exists some directions to be investigated in the future. Under the constraints of the recognized modification range, adversarial examples can still be found through careful observation; the success rates of attacking models with defense mechanisms are not high enough. We hope that these problems can be resolved in future research.

Acknowledgements This work was supported by the National Key R&D Program of China (No. 2020AAA0107704), National Natural Science Foundation of China (Nos. 62073263, 62025602, 62102105), Guangdong Basic and Applied Basic Research Foundation (Grant Nos. 2020A1515110997, 2022A1515011501), Technological Innovation Team of Shaanxi Province (Grant No. 2020TD-013).

Data availability The data sets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1440–1448
2. Shi L, Wang L, Long C, Zhou S, Zhou M, Niu Z, Hua G (2021) SGCN: Sparse graph convolution network for pedestrian trajectory prediction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 8994–9003
3. Tang K, Ma Y, Miao D, Song P, Gu Z, Tian Z, Wang W (2022) Decision fusion networks for image classification. IEEE Trans Neural Netw Learn Syst 1:1
4. Li W, Guo T, Li P, Chen B, Wang B, Zuo W, Zhang L (2021) Enhancing face recognition via unlabeled shallow data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 14729–14738
5. Li J, Li B, Jiang Y, Cai W (2022) MSAt-GAN: a generative adversarial network based on multi-scale and deep attention mechanism for infrared and visible light image fusion. Complex Intell Syst 8(6):4753–4781
6. Haq EU, Jianjun H, Huarong X, Li K (2021) Block-based compressed sensing of MR images using multi-rate deep learning approach. Complex Intell Syst 7(5):2437–2451
7. Guo S, Li X, Zhu P, Mu Z (2023) Ads-detector: an attention-based dual stream adversarial example detection method. Knowl Based Syst 265:110388
8. Wang K, Li F, Chen C-M, Hassan MM, Long J, Kumar N (2021) Interpreting adversarial examples and robustness for deep learning-based auto-driving systems. IEEE Trans Intell Transp Syst 23(7):9755–9764
9. Zhang Y, Tian X, Li Y, Wang X, Tao D (2020) Principal component adversarial example. IEEE Trans Image Process 29:4804–4815
10. Gao H, Zhang H, Yang X, Li W, Gao F, Wen Q (2022) Generating natural adversarial examples with universal perturbations for text classification. Neurocomputing 471:175–182
11. Tang K, Shi Y, Wu J, Peng W, Khan A, Zhu P, Gu Z (2022) Normalattack: curvature-aware shape deformation along normals for imperceptible point cloud attack. Secur Commun Netw 6:1–11
12. Zhang R, Luo S, Pan L, Hao J, Zhang J (2022) Generating adversarial examples via enhancing latent spatial features of benign traffic and preserving malicious functions. Neurocomputing 490:413–430
13. Liu Y, Chen X, Liu C, Song D (2017) Delving into transferable adversarial examples and black-box attacks. In: International conference on learning representations (ICLR)
14. Goodfellow IJ, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples. In: Proceedings of international conference on learning representations (ICLR)
15. Lin J, Song C, He K, Wang L, Hopcroft JE (2020) Nesterov accelerated gradient and scale invariance for adversarial attacks. In: International conference on learning representations (ICLR)
16. Dong Y, Liao F, Pang T, Su H, Zhu J, Hu X, Li J (2019) Boosting adversarial attacks with momentum. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 9185–9193
17. Xie C, Zhang Z, Zhou Y, Bai S, Wang J, Ren Z, Alan Y (2019) Improving transferability of adversarial examples with input diversity. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2730–2739

18. Dong Y, Pang T, Su H, Zhu J (2019) Evading defenses to transferable adversarial examples by translation-invariant attacks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 4312–4321
19. Wang X, He X, Wang J, He K (2021) Admix: enhancing the transferability of adversarial attacks. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 16138–16147
20. Zou J, Pan Z, Qiu J, Liu X, Rui T, Li W (2020) Improving the transferability of adversarial examples with resized-diverse-inputs, diversity-ensemble and region fitting. In: European conference on computer vision (ECCV)
21. Li Y, Bai S, Zhou Y, Xie C, Zhang Z, Yuille A (2020) Learning transferable adversarial examples via ghost networks. In: the 34th AAAI conference on artificial intelligence, pp 11458–11465
22. Hao L, Hao K, Wei B, Tang X-S (2022) Boosting the transferability of adversarial examples via stochastic serial attack. Neural Netw 150:58–67
23. Xu Z, Li X, Stojanovic V (2021) Exponential stability of nonlinear state-dependent delayed impulsive systems with applications. Nonlinear Anal Hybrid Syst 42:101088
24. Wei T, Li X, Stojanovic V (2021) Input-to-state stability of impulsive reaction-diffusion neural networks with infinite distributed delays. Nonlinear Dyn 103:1733–1755
25. Song X, Sun P, Song S, Stojanovic V (2022) Event-driven NN adaptive fixed-time control for nonlinear systems with guaranteed performance. J Frankl Inst 359(9):4138–4159
26. Tramér F, Kurakin A, Papernot N, Goodfellow I, Boneh D, McDaniel P (2018) Ensemble adversarial training: attacks and defenses. In: International conference on learning representations (ICLR)
27. Zi B, Zhao S, Ma X, Jiang Y-G (2021) Revisiting adversarial robustness distillation: robust soft labels make student better. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR), pp 16443–16452
28. Liao F, Liang M, Dong Y, Pang T, Hu X, Zhu J (2018) Defense against adversarial attacks using high-level representation guided denoiser. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1778–1787
29. Cohen JM, Rosenfeld E, Kolter JZ (2019) Certified adversarial robustness via randomized smoothing. In: International conference on machine learning (ICML), pp 1310–1320
30. Guo F, Zhao Q, Li X, Kuang X, Zhang J, Han Y, Tan Y-A (2019) Detecting adversarial examples via prediction difference for deep neural networks. Inf Sci 501:182–192
31. Wang Z, Guo H, Zhang Z, Liu W, Qin Z, Ren K (2021) Feature importance-aware transferable adversarial attacks. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 7619–7628
32. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2014) Intriguing properties of neural networks. In: International conference on learning representations (ICLR)
33. Zhang X, Zhang X, Sun M, Zou X, Chen K, Yu N (2022) Imperceptible black-box waveform-level adversarial attack towards automatic speaker recognition. Complex Intell Syst 2022:1–15
34. Chen J, Zheng H, Xiong H, Shen S, Su M (2020) MAG-GAN: massive attack generator via gan. Inf Sci 536:67–90
35. Yuan X, He P, Zhu Q, Li X (2019) Adversarial examples: attacks and defenses for deep learning. IEEE Trans Neural Netw Learn Syst 30(9):2805–2824
36. Kurakin A, Goodfellow IJ, Bengio S (2017) Adversarial examples in the physical world. In: Proceedings of international conference on learning representations (ICLR)
37. Wang X, He K (2021) Enhancing the transferability of adversarial attacks through variance tuning. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1924–1933
38. Hong J, Tang K, Gao C, Wang S, Guo S, Zhu P (2022) GM-Attack: improving the transferability of adversarial attacks. In: 2022 international conference on knowledge science, engineering and management (KSEM), pp 489–500
39. Zhu P, Hou X, Tang K, Liu Y, Zhao Y, Wang Z (2023) Unsupervised feature selection through combining graph learning and $\ell_{2,0}$-norm constraint. Inf Sci 622:68–82
40. Tang K, Shi Y, Lou T, Peng W, He X, Zhu P, Gu Z, Tian Z (2022) Rethinking perturbation directions for imperceptible adversarial attacks on point clouds. IEEE Internet Things J 1:1
41. Zhou W, Hou X, Chen Y, Tang M, Huang X, Gan X, Yang Y (2018) Transferable adversarial perturbations. In: Proceedings of European conference on computer vision (ECCV), pp 471–486
42. Ganeshan A, Vivek BS, Radhakrishnan VB (2019) FDA: feature disruptive attack. In: Proceedings of IEEE international conference on computer vision (ICCV), pp 8068–8078
43. Inkawhich N, Wen W, Li H, Chen Y (2019) Feature space perturbations yield more transferable adversarial examples. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR), pp 7066–7074
44. Huang Q, Katsman I, He H, Gu Z, Belongie S, Lim S-N (2019) Enhancing adversarial example transferability with an intermediate level attack. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 4732–4741
45. Zhang Y, Tan Y-A, Chen T, Liu X, Zhang Q, Li Y (2022) Enhancing the transferability of adversarial examples with random patch. In: Proceedings of the 31th international joint conference on artificial intelligence (IJCAI), pp 1672–1678
46. Nips17 Adversarial Attacks and Defenses Competition. https://github.com/cleverhans-lab/cleverhans/tree/master/cleverhans_v3.1.0/examples/nips17_adversarial_competition/dataset
47. Szegedy C, Vanhoucke V, Sergey I, Jon S, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR), pp 2818–2826
48. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of AAAI conference on artificial intelligence, pp 4278–4284
49. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778
50. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations (ICLR)
51. Inkawhich N, Liang KJ, Carin L, Chen Y (2020) Transferable perturbations of deep feature distributions. In: International conference on learning representations (ICLR)