Neural Computing and Applications
https://doi.org/10.1007/s00521-023-09259-5

ORIGINAL ARTICLE

Improving the transferability of adversarial examples with separable positive and negative disturbances

Yuanjie Yan[1,2] · Yuxuan Bu[1,2] · Furao Shen[1,4] · Jian Zhao[3]

Received: 27 April 2022 / Accepted: 6 November 2023

Abstract
Adversarial examples demonstrate the vulnerability of white-box models but exhibit weak transferability to black-box models. In image processing, each adversarial example usually consists of original image and disturbance. The disturbances are essential for the adversarial examples, determining the attack success rate on black-box models. To improve the transferability, we propose a new white-box attack method called separable positive and negative disturbance (SPND). SPND optimizes the positive and negative perturbations instead of the adversarial examples. SPND also smooths the search space by replacing constrained disturbances with unconstrained variables, which improves the success rate of attacking the black-box model. Our method outperforms the other attack methods in the MNIST and CIFAR10 datasets. In the ImageNet dataset, the black-box attack success rate of SPND exceeds the optimal CW method by nearly ten percentage points under the perturbation of $L_\infty = 0.3$.

Keywords Adversarial examples · Transferability · Black-box attack

1 Introduction

Adversarial examples deceive the neural network to predict the incorrect outputs by adding little malicious perturbations to the original inputs [8]. Network models are surprisingly susceptible to adversarial examples, which seriously hinders the development of deep neural networks. The vulnerability of neural networks spurs the research of adversarial samples. Attack methods aim at generating adversarial examples, which have been extensively studied in the literature. There are roughly three different ways of generating adversarial examples: (a) using the gradient of the network [8, 13]; (b) using the optimization method [3]; and (c) using the generative adversarial network [15, 20]. Moreover, in the context of the disturbance, these methods can be categorized according to the L_0 distance [16], the L_2 distance [14], and the L_∞ distance [29].

A white-box attack on the neural network means that the model is exposed in public, including the framework and the parameters. White-box attack methods can exploit the details of model to generate adversarial examples. However, the model is unavailable to the attacker in realistic scenarios, which is regarded as the black-box model. Adversarial examples can be directly applied to attack the black-box model. Transferability aims to evaluate the ability of adversarial examples to deceive black-box models. Most white-box attack methods neglect to evaluate the transferability of adversarial examples. Wu et al. [24] demonstrate that white-box attack methods have limited transferability due to overfitting the employed model.

✉ Furao Shen
frshen@nju.edu.cn

✉ Jian Zhao
jianzhao@nju.edu.cn

Yuanjie Yan
yanyj@smail.nju.edu.cn

Yuxuan Bu
mg1833002@smail.nju.edu

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

[2] Department of Computer Science and Technology, Nanjing University, Nanjing, China

[3] Department of Electronic Science and Engineering, Nanjing University, Nanjing, China

[4] School of Artificial Intelligence, Nanjing University, Nanjing, China

Published online: 07 December 2023

Studying the performance of attack methods in black-box models is a practical problem.

An intriguing property of deep neural networks is that adversarial examples can be transferred between different models whether they are homogeneous or heterogeneous [17]. It reveals that the transferability of adversarial examples is ubiquitous across different models on the same task. Recently, some researchers are concerned about improving the transferability of adversarial examples. Xie et al. [26] apply random transformations to the input images to avoid gradient overfitting at each iteration. Dong et al. [6] propose a translation-invariant attack method to generate transferable adversarial examples. Li et al. [12] exploit the transferability of regional homogeneity to attack the black-box model. However, these works ignore quantitative studies on the transferability of black-box models under the same constrained perturbations.

In this paper, we focus on the transferability of attack methods based on the L_∞ distance. First, we verify the discrepancies among different models on the same task. Figure 1 shows that the pre-trained classifiers have different distinguishing areas. This phenomenon limits the transferability of adversarial examples for attack methods. It also indicates that the disturbance should cover the whole image for better transferability. Second, we analyse the disturbances about attack methods. The perturbations are determined by magnitude and direction. We evaluate the perturbations with different constrained L_∞ distances. For the same image, different attack methods can generate different adversarial examples with distinct disturbances. For adversarial examples, the larger the perturbation, the higher the transferability. However, not all attack methods can reach the upper and lower bounds of the L_∞ distance. Moreover, the directions of the perturbation also affect the transferability, even if the magnitudes are similar for adversarial examples. We observe that a disturbance with good transferability should have two characteristics under the L_∞ distance. One is that when the constraints are satisfied, the perturbation should be as large as possible in the whole image region. The other is that the direction of the disturbance is crucial for the misclassification of the classifier. Based on the distribution space, we propose the separable positive and negative disturbance method (SPND) which overcomes the "box constraint" by optimizing perturbation with unconstrained variables. SPND is in line with the above observations when the constraints are satisfied on the L_∞ distance. More experiments about SPND are verified in Sect. 4. Our contributions are summarized as follows:

- Considering of the L_∞ distance, we propose two characteristics of the transferable disturbances. First, when the constraints are satisfied, the perturbation should be as large as possible. Second, the direction of the disturbance is crucial for transferability.
- Based on those hypothesis, we propose a SPND method that overcomes the "box constraint" on the L_∞ distance.
- Compared with state-of-the-art methods, SPND demonstrates superior transferability of adversarial examples for black-box attacks. The disturbances generated by SPND are consistent with our hypothesis.

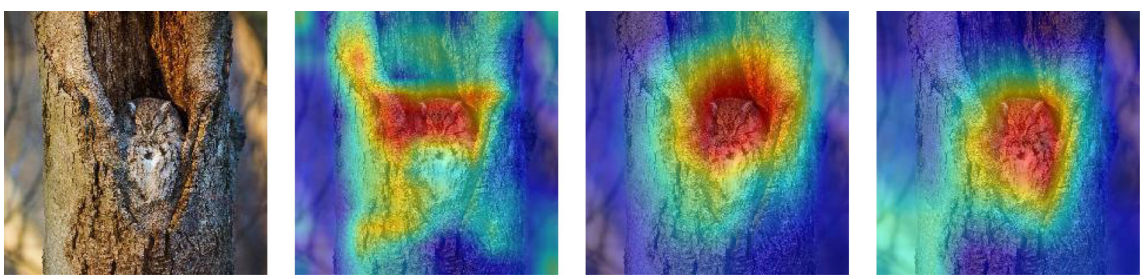
2 Problem and attack methods

First of all, we give a unified formula for searching the appropriate disturbance under the L_∞ distance. Then, we discuss some advanced attack methods as benchmarks.

2.1 Definition of the problem

Considering the image classification task, we denote the image as $\boldsymbol{x} \in \mathbb{R}^{N\times M}$ and the label of the image as $y \in \{0, 1, ..., K-1\}$, where N and M represent the size of the image, and K is the number of categories. Some neural network models in which f_w represents a white-box model and f_b represents a black-box model are pre-trained on the same dataset.

The correct predicted $f_w(\boldsymbol{x})$ and $f_b(\boldsymbol{x})$ should be the same as the image label y. However, if a small malicious disturbance $\boldsymbol{\delta}$ is added to $\boldsymbol{x}$, the output of the model $f_w(\boldsymbol{x})$ will



(a) The scope image (b) Vgg19 (c) Resnet50 (d) Densenet121

Fig. 1 Demonstration of the different discriminative regions of the classifier models

be different from the expected one, i.e. $f_w(\boldsymbol{x}+\boldsymbol{\delta}) \neq y$. The attack method aims to find malicious perturbations on the white-box model, which can be formalized as:

$$\boldsymbol{\delta}_{adv} = \arg\max_{\boldsymbol{\delta}}(\ell(f_w(\boldsymbol{x}+\boldsymbol{\delta}), y)), \text{s.t.} \parallel \boldsymbol{\delta} \parallel_\infty \leq \epsilon \tag{1}$$

where ℓ denotes the loss of model f_w and ϵ represents the upper bound of the disturbance. We denote an adversarial example as $\boldsymbol{x}_{adv} = \boldsymbol{x} + \boldsymbol{\delta}_{adv}$ for the model f_w.

We evaluate the transferability of adversarial samples on the black-box model. For the image dataset X, each adversarial sample $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is generated by (1). The accuracy of the attack method in the black-box model is defined as follows,

$$Acc(f_b, f_w, X) = \frac{1}{m}\Sigma_{\boldsymbol{x}_i \in X}\mathbb{I}(f_b(\boldsymbol{x}_{i,adv} = y_i)), \tag{2}$$

where m is the size of X and $\boldsymbol{x}_{i,adv}$ is the adversarial example of $\boldsymbol{x}_i$ by attacking the f_w. Lower accuracy means that the adversarial examples are more transferable. In this paper, we solve the constraint problem in (1) and evaluate attack methods about the transferability of adversarial examples in (2).

2.2 Attack methods

A series of attack methods have been proposed from different aspects [2]. However, each method has its own applicable scopes and conditions. It is difficult to quantify the advantages and disadvantages of attack methods under the same metrics. In the following, we focus on the relevant algorithms to solve the problem in (1).

The fast gradient sign method (FGSM) [8] generates adversarial examples by exploiting the gradients of the original input on the white-box model. This method obtains adversarial examples with only one operation step as follows:

$$\boldsymbol{x}_{adv} = \text{clip}_a(\boldsymbol{x} + \epsilon \cdot \text{Sign}(\nabla_{\boldsymbol{x}}(\ell(f_w(\boldsymbol{x}), y)))) \tag{3}$$

where $\text{clip}_a(\boldsymbol{x})$ is to limit the range of $\boldsymbol{x}$, i.e. the range of inputs is [0, 1] after normalization. The perturbation $\boldsymbol{\delta} = \epsilon \cdot \text{Sign}(\nabla_{\boldsymbol{x}}(\ell(f_w(\boldsymbol{x}), y)))$ satisfies the restriction of the L_∞ distance. FGSM is often used as a benchmark for comparing attack methods.

As a variant of FGSM, projected gradient descent (PGD) indicates a more aggressive attack method with iterative optimization on white-box models. The iterative process can be expressed as

$$\boldsymbol{x}' = \text{clip}_a(\boldsymbol{x}^t + \alpha \cdot \text{Sign}(\nabla_{\boldsymbol{x}'}(\ell(f_w(\boldsymbol{x}^t), y)))) \tag{4}$$

and $\boldsymbol{x}^{t+1} = \text{clip}_b(\boldsymbol{x}')$, where $\boldsymbol{x}^0 = \boldsymbol{x}$, α is the step size and $\text{clip}_b(\boldsymbol{x})$ limits the interference to the L_∞ distance. After iterating k rounds, the final adversarial example and disturbance are obtained as $\boldsymbol{x}^k$ and $\boldsymbol{\delta} = \boldsymbol{x}^k - \boldsymbol{x}$, respectively. Compared to FGSM, PGD can generate more aggressive adversarial examples for the white-box models. However, when exploiting the transferability of the adversarial samples, there is a non-negligible gap, even lower than FGSM due to overfitting.

Momentum-based iterative method (MIM) [5] won first place in both non-target adversarial attack and target adversarial attack competitions. In this competition, the adversarial examples are generated by attacking a white-box model and evaluated on a black-box model. MIM is committed to improving the transferability of adversarial examples for the white-box attack. FGSM and PGD only use the current gradient of the input, but MIM takes advantage of the accumulated gradients. The update process can be expressed as

$$\boldsymbol{g}^t = \mu \cdot \boldsymbol{g}^{t-1} + \nabla_{\boldsymbol{x}'}(\ell(f_w(\boldsymbol{x}^t), y)) \tag{5}$$

where $\boldsymbol{g}^t$ is the accumulated gradient and μ is a decay rate. The adversarial example $\boldsymbol{x}_{adv}$ is iteratively constructed as

$$\boldsymbol{x}^{t+1} = \text{clip}_b(\text{clip}_a(\boldsymbol{x}^t + \alpha \cdot \text{Sign}(\boldsymbol{g}^t))). \tag{6}$$

MIM can be regarded as a gradient descent method with momentum on the input.

To achieve one-to-one mapping about adversarial examples on the RGB space, the maximum perturbation of each pixel needs to be limited individually on the normalization space. For example, the values of pixels must be restricted to the range [0,1] after normalization on the MNIST dataset. The previous methods use the clip_a and clip_b functions to satisfy such constraints. The Carlini and Wagner attack method (CW) [3] introduces a new variable z instead of optimizing disturbance $\boldsymbol{\delta}$ by setting

$$\boldsymbol{\delta} = \frac{1}{2}(\tanh(\boldsymbol{z}) + 1) - \boldsymbol{x}. \tag{7}$$

Here, the disturbance $\boldsymbol{z}$ is unconstrained. The CW method also trains $\boldsymbol{z}$ with a gradient ascending on the white-box model. The original CW method is only applicable when the input range is from 0 to 1. The CW method can solve (1) by scaling and shifting the input.

Besides the above methods, Feature Importance-aware Attack (FIA) [21] disrupts important object-aware features

of intermediate layers. Zhang [28] introduce Just Noticeable Difference (JND) as prior information into adversarial attacks, making the insensitive modification to image. According to the research on adversarial examples about image, Wei [22] study adversarial examples in video recognition. However, most of the research only focuses on the adversarial example under specific conditions. Therefore, those methods fail to generate adversarial samples to solve the problem (1).

3 Separable positive and negative disturbances method

The gradient-based optimization approaches need to deal with the "box constraint" which is well known in the optimization literature. According to (1), FGSM and PGD use the clip_a and clip_b functions to satisfy those constraints. Those clipping method causes the disappearance of the gradient at the truncation [3]. To solve this problem, CW introduces a new variable $\boldsymbol{z}$ instead of the image $\boldsymbol{x}$. However, the above methods optimize the generated image rather than perturbation.

In this section, we extend the CW method with the positive and negative disturbances on normalized spaces. As shown in Fig. 2, we optimize the positive and negative disturbances instead of the generated image. The reason for separating positive and negative disturbances is the inconsistency of the upper and lower bounds of the disturbances. The optimization is smoother in the perturbation space than in the image space. We also use local random sampling to accumulate gradients around the image.

3.1 Positive and negative disturbances

According to the definition of the problem in Sect. 2.1, it is challenging to directly optimize the disturbance under the joint constraints of the L_∞ distance and the RGB space. Inspired by the CW method, we propose the SPND method which improves the transferability of adversarial examples and narrows the gap between the white-box attack and the black-box attack. First, we define a positive perturbation $\boldsymbol{p_p}$ and a negative perturbation $\boldsymbol{p_n}$. Considering the input and the L_∞ distance limitations, $\boldsymbol{p_p}$ and $\boldsymbol{p_n}$ have the following restrictions:

$$\begin{aligned} &\| \boldsymbol{x} + \boldsymbol{p_p} \|_\infty \leq r, \ \| \boldsymbol{p_p} \|_\infty \leq \epsilon \\ &\| \boldsymbol{x} - \boldsymbol{p_n} \|_\infty \geq l, \ \| \boldsymbol{p_n} \|_\infty \leq \epsilon \end{aligned} \tag{8}$$

where the input x_i is limited to $[l, r]$ for the i-th pixel of $\boldsymbol{x}$. We denote $\boldsymbol{l}$, $\boldsymbol{r}$ and $\boldsymbol{\epsilon}$ as the extensions of l, r and ϵ with the same size of $\boldsymbol{x}$, respectively. We convert (8) to acquire the upper bounds for pixels as:

$$\begin{aligned} &0 \leq \boldsymbol{p_p} \leq \min(\boldsymbol{r} - \boldsymbol{x}, \boldsymbol{\epsilon}) \\ &0 \leq \boldsymbol{p_n} \leq \min(\boldsymbol{x} - \boldsymbol{l}, \boldsymbol{\epsilon}). \end{aligned} \tag{9}$$

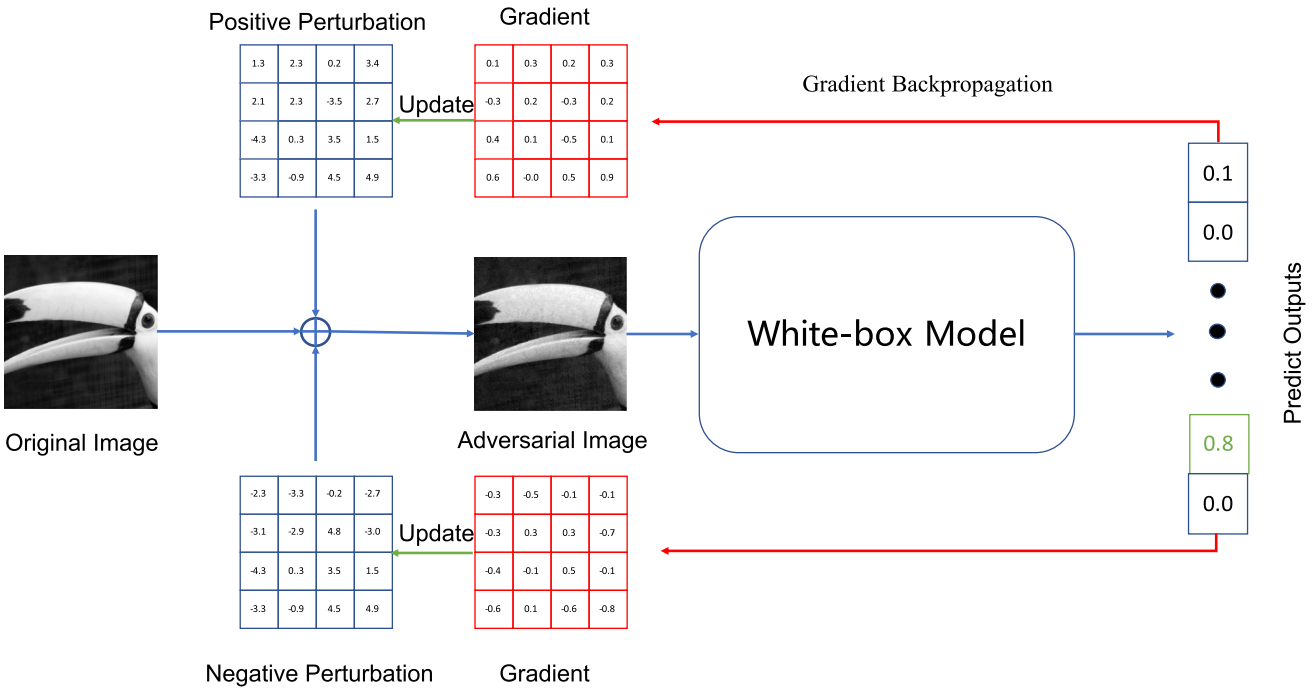


Fig. 2 The overview of SPND

For network models, directly optimizing $\boldsymbol{p_n}$ and $\boldsymbol{p_p}$ cannot satisfy Eq. (9).

We denote $\boldsymbol{b_p}$ as $\min(\boldsymbol{r}-\boldsymbol{x},\boldsymbol{\epsilon})$ and $\boldsymbol{b_n}$ as $\min(\boldsymbol{x}-\boldsymbol{l},\boldsymbol{\epsilon})$. Then, we introduce additional variables $\boldsymbol{z_p}$ and $\boldsymbol{z_n}$ as follows:

$$\begin{aligned} 0 \le \boldsymbol{b_p} \cdot \text{Sigmoid}(\boldsymbol{z_p}) \le \boldsymbol{b_p} \\ 0 \le \boldsymbol{b_n} \cdot \text{Sigmoid}(\boldsymbol{z_n}) \le \boldsymbol{b_n}. \end{aligned} \tag{10}$$

The unrestricted $\boldsymbol{z_p}$ and $\boldsymbol{z_n}$ are optimized instead of the restricted $\boldsymbol{p_p}$ and $\boldsymbol{p_n}$. This approach eliminates the problem of "box constraint" and makes the gradient descent smooth. The adversarial sample can be calculated as

$$\boldsymbol{x}^t = \boldsymbol{x} + \boldsymbol{b_p} \cdot \text{Sigmoid}(\boldsymbol{z_p}^t) - \boldsymbol{b_n} \cdot \text{Sigmoid}(\boldsymbol{z_n}^t). \tag{11}$$

We also iteratively optimize the variables $\boldsymbol{z_p}^t$ and $\boldsymbol{z_n}^t$ as

$$\begin{aligned} \boldsymbol{z_p}^{t+1} &= \boldsymbol{z_p}^t + \alpha \cdot \nabla_{\boldsymbol{z_p}^t}(\ell(f_w(\boldsymbol{x}^t), y)) \\ \boldsymbol{z_n}^{t+1} &= \boldsymbol{z_n}^t + \alpha \cdot \nabla_{\boldsymbol{z_n}^t}(\ell(f_w(\boldsymbol{x}^t), y)) \end{aligned} \tag{12}$$

where $\boldsymbol{x}^0 = \boldsymbol{x}$, $\boldsymbol{z_p}^0$ and $\boldsymbol{z_n}^0$ are initialized randomly. In the end, $\boldsymbol{x}^T$ is assigned to $\boldsymbol{x}_{adv}$ after iterating (12) for T rounds.

3.2 Local random sampling

SPND uses the Sigmoid() function instead of the clip() function. SPND can iteratively search the feasible space more smoothly. However, in some researches [19, 25], they point out that attacks based on iterative optimization tend to overfit the white-box model. Iterative optimization methods weaken the transferability of adversarial samples to the black-box model. Therefore, we adopt local random sampling (LRS) to improve the transferability for the separated positive and negative disturbances method, which is abbreviated as RSPND. While SPND focuses on the gradient of a single sample $\boldsymbol{x}$, RSPND exploits the gradients of neighbouring samples around $\boldsymbol{x}$ to generate more transferable perturbations. The gradient of local random sampling is calculated as follows,

$$\begin{aligned} \boldsymbol{u_p}^t &= \frac{1}{g}\sum_{i=1}^{m} \nabla_{\boldsymbol{z_p}^t}(\ell(f_w(\boldsymbol{x}^t + \boldsymbol{r}_i), y)) \\ \boldsymbol{u_n}^t &= \frac{1}{g}\sum_{i=1}^{m} \nabla_{\boldsymbol{z_n}^t}(\ell(f_w(\boldsymbol{x}^t + \boldsymbol{r}_i), y)) \end{aligned} \tag{13}$$

where g is the group size of the sampling and $\boldsymbol{r_i}$ is a small random noise for the i-th sample. Then, we combine the neighbouring gradients with the original gradients to update the variables $\boldsymbol{z_p}$ and $\boldsymbol{z_n}$ as

$$\begin{aligned} \boldsymbol{z_p}^{t+1} &= \boldsymbol{z_p}^t + \beta \boldsymbol{u_p}^t + \alpha \cdot \nabla_{\boldsymbol{z_p}^t}(\ell(f_w(\boldsymbol{x}^t), y)) \\ \boldsymbol{z_n}^{t+1} &= \boldsymbol{z_n}^t + \beta \boldsymbol{u_n}^t + \alpha \cdot \nabla_{\boldsymbol{z_n}^t}(\ell(f_w(\boldsymbol{x}^t), y)) \end{aligned} \tag{14}$$

where β and α are the step size for updating the adjacent and original gradients, respectively.

Algorithm 1 Separable Positive and Negative Disturbances Method (SPND)

Input: image $\boldsymbol{x}$, the white-box model f_w, the L_∞ distance ϵ, iteration step T and the range of input is from l to r.
Output: adversarial example $\boldsymbol{x}_{adv}$, the positive perturbation $\boldsymbol{p_p}$ and the negative perturbation $\boldsymbol{p_n}$.
Process:

Initialize $\boldsymbol{z_p}$ and $\boldsymbol{z_n}$.
$\boldsymbol{b_p} = \min(\boldsymbol{r}-\boldsymbol{x},\boldsymbol{\epsilon})$, $\boldsymbol{b_n} = \min(\boldsymbol{x}-\boldsymbol{l},\boldsymbol{\epsilon})$.
for $t = 0$ to T **do**
  $\boldsymbol{x}^t = \boldsymbol{x} + \boldsymbol{b_p} \cdot \text{Sigmoid}(\boldsymbol{z_p}) - \boldsymbol{b_n} \cdot \text{Sigmoid}(\boldsymbol{z_n})$.
  $\boldsymbol{z_p} = \boldsymbol{z_p} + \alpha \cdot \nabla_{\boldsymbol{z_p}}(\ell(f_w(\boldsymbol{x}^t), y))$.
  $\boldsymbol{z_n} = \boldsymbol{z}_n + \alpha \cdot \nabla_{\boldsymbol{z_n}}(\ell(f_w(\boldsymbol{x}^t), y))$.
end for
$\boldsymbol{x}_{adv} = \boldsymbol{x} + \boldsymbol{b_p} \cdot \text{Sigmoid}(\boldsymbol{z_p}) - \boldsymbol{b_n} \cdot \text{Sigmoid}(\boldsymbol{z_n})$.
$\boldsymbol{p_p} = \boldsymbol{b_p} \cdot \text{Sigmoid}(\boldsymbol{z_p}), \boldsymbol{p_n} = \boldsymbol{b_n} \cdot \text{Sigmoid}(\boldsymbol{z_n})$.
return $\boldsymbol{x}_{adv}, \boldsymbol{p_p}, \boldsymbol{p_n}$.

3.3 Access transferability

In this work, we not only care about the performance of adversarial examples on the white-box model but also the transferability on the black-box model. The attack method employs the f_w model to generate adversarial examples which are directly migrated to attack the f_b model. According to the success rate of the black-box model, we evaluate the transferability of adversarial examples for different attack methods. For the image dataset X of size m, we generate a corresponding adversarial sample $\boldsymbol{x}_{i,adv}$ for each $\boldsymbol{x}_i$, where $(\boldsymbol{x}_i, y_i) \in X$. The transferability of the adversarial examples on the black-box model can be defined as follows:

$$Tr(f_b, f_w, X) = 1 - Acc(f_b, f_w, X). \tag{15}$$

Since the sum of $Acc(\cdot)$ in (2) and $Tr(\cdot)$ in (15) is equal to 1, we can compare $Acc(\cdot)$ about the attack methods to evaluate the transferability on the black-box model. In summary, Algorithm 1 and Algorithm 2 show a brief description of the SPND and RSPND methods, respectively.

Algorithm 2 Separable Positive and Negative Disturbances Method with Local Random Sampling (RSPND)

Input: input $\boldsymbol{x}$, the white-box model f_w, the L_∞ distance ϵ, iteration step T, the range of input is from l to r, the number of samples g and the maximum noise noi .
Output: adversarial example $\boldsymbol{x}_{adv}$, the positive perturbation $\boldsymbol{p_p}$ and the negative perturbation $\boldsymbol{p_n}$.
Process:

Initialize $\boldsymbol{z_p}$ and $\boldsymbol{z_n}$.
$\boldsymbol{b_p} = \min(\boldsymbol{r} - \boldsymbol{x}, \boldsymbol{\epsilon})$, $\boldsymbol{b_n} = \min(\boldsymbol{x} - \boldsymbol{l}, \boldsymbol{\epsilon})$.
for $t = 0$ to T **do**
  $\boldsymbol{x}^t = \boldsymbol{x} + \boldsymbol{b_p} \cdot \text{Sigmoid}(\boldsymbol{z_p}) - \boldsymbol{b_n} \cdot \text{Sigmoid}(\boldsymbol{z_n})$.
  $\boldsymbol{u_p} = 0, \boldsymbol{u_n} = 0$
  for $i = 0$ to g **do**
    Generate $\boldsymbol{r}_i = \text{Random}(noi)$
    $\boldsymbol{u_p} = \frac{1}{m}\nabla_{\boldsymbol{z_p}}(\ell(f_w(\boldsymbol{x}^t + \boldsymbol{r}_i, y)))$
    $\boldsymbol{u_n} = \frac{1}{m}\nabla_{\boldsymbol{z_n}}(\ell(f_w(\boldsymbol{x}^t + \boldsymbol{r}_i, y)))$
  end for
  $\boldsymbol{z_p} = \boldsymbol{z_p} + \beta\boldsymbol{u_p} + \alpha \cdot \nabla_{\boldsymbol{z_p}}(\ell(f_w(\boldsymbol{x}^t), y))$.
  $\boldsymbol{z_n} = \boldsymbol{z_n} + \beta\boldsymbol{u_n} + \alpha \cdot \nabla_{\boldsymbol{z_n}}(\ell(f_w(\boldsymbol{x}^t), y))$.
end for
$\boldsymbol{x}_{adv} = \boldsymbol{x} + \boldsymbol{b_p} \cdot \text{Sigmoid}(\boldsymbol{z_p}) - \boldsymbol{b_n} \cdot \text{Sigmoid}(\boldsymbol{z_n})$.
$\boldsymbol{p_p} = \boldsymbol{b_p} \cdot \text{Sigmoid}(\boldsymbol{z_p}), \boldsymbol{p_n} = \boldsymbol{b_n} \cdot \text{Sigmoid}(\boldsymbol{z_n})$.
return $\boldsymbol{x}_{adv}, \boldsymbol{p_p}, \boldsymbol{p_n}$.

3.4 Overall disturbance

The overall disturbance $\boldsymbol{\delta}$ is the sum of the positive and negative perturbations, which can be expressed as

$$\boldsymbol{\delta} = \boldsymbol{b_p} \cdot \text{Sigmoid}(\boldsymbol{z_p}) - \boldsymbol{b_n} \cdot \text{Sigmoid}(\boldsymbol{z_n}). \tag{16}$$

The unconstrained $\boldsymbol{z_p}$ and $\boldsymbol{z_n}$ are optimized iteratively according to (12) and (14). Moreover, the direction of the gradients is opposite for z_p and z_n. For each pixel $x_{i,j}$ meaning the value of the i-th row and j-th column in the image, the positive and negative gradients satisfy

$$\frac{\partial \ell(f_w(\boldsymbol{x}_{adv}), y)}{\partial z_{p_{ij}}} \cdot \frac{\partial \ell(f_w(\boldsymbol{x}_{adv}), y)}{\partial z_{n_{ij}}} \leq 0. \tag{17}$$

When $z_{p_{ij}}$ increases, the $z_{n_{ij}}$ inevitably decreases and vice versa.

It is impossible for $z_{p_{ij}}$ and $z_{n_{ij}}$ to increase or decrease simultaneously while the overall disturbance $\delta_{i,j}$ remains constant. The proof is by the chain rule of derivation. We denote the $\ell(f_w(\boldsymbol{x}_{adv}), y)$ as $\mathcal{L}$, the derivatives of $z_{p_{ij}}$ and $z_{n_{ij}}$ with respect to $\mathcal{L}$ are as follows,

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial z_{p_{ij}}} &= b_{p_{ij}} \cdot \text{Sigmoid}(z_{p_{ij}})' \cdot \frac{\partial \mathcal{L}}{\partial \delta_{i,j}} \\ \frac{\partial \mathcal{L}}{\partial z_{n_{ij}}} &= -b_{n_{ij}} \cdot \text{Sigmoid}(z_{n_{ij}})' \cdot \frac{\partial \mathcal{L}}{\partial \delta_{i,j}} \end{aligned} \tag{18}$$

Since $b_{p_{ij}} \cdot \text{Sigmoid}(z_{p_{ij}}))'$ and $b_{n_{ij}} \cdot \text{Sigmoid}(z_{n_{ij}})'$ are positive, (17) is proved.

4 Experiment

In this section, we evaluate attack methods by attacking the classification networks on the MNIST, CIFAR10 [11] and ImageNet [4] datasets. First, different architecture networks have been pre-trained to achieve a fine classification performance on their own datasets. Those models are classified as the white-box models and black-box models. Then, under the L_∞ distance, attack methods exploit those white-box models to generate the adversarial examples. We evaluate the transferability of those adversarial examples when testing them on the black-box models. Our implementations are based on PyTorch [1].

Note that the perturbation settings in this paper are different from most previous works. As shown in Refs. [6, 12], they set the $L_\infty = 16/32$ with the range [0, 255] for each pixel in the RGB space. However, we define the disturbance constraints in the regularization space. We consider three reasons to evaluate the disturbances in the regularization space. Firstly, the adversarial examples are actually evaluated in the regularization space, even though the constraints of disturbance are defined in the RGB space. Most pre-trained classifiers normalize images before inputting them to the network. For example, the normalization with $mean = [0.485, 0.456, 0.406]$ and $std = [0.229, 0.224, 0.225]$ for channels is adopted to deal with the original images in ImageNet. Secondly, the relative error is negligible when converting an image from the regularization space to the RGB space. The regularization space and RGB space are easily converted. Thirdly, we analyse the directions and the distributions of disturbances on the regularization space. We verify two hypotheses about the transferability of the disturbance by analysing the disturbances of attack methods. Therefore, this paper focuses on the disturbance in the regularization space.

4.1 Experiment setup

Datasets According to (1) and (15), we apply attack methods to generate adversarial samples on the white-box

model and evaluate those samples on the black-box model. The lower the classification accuracy in the black-box model, the better the transferability of the adversarial samples generated by the attack method. When evaluating the classification accuracy with different L_∞ distances, we choose to generate 3200 adversarial samples on MNIST, 5120 adversarial samples on CIFAR10 and 3200 adversarial samples on ImageNet. These samples are all randomly selected from their test datasets, respectively.

Pretrained models We pre-train the different architecture models on MNIST and CIFAR10 datasets. Model $Mnist_s$ represents a classified network with two layers of convolution for MNIST. Model $Mnist_b$ has six convolution layers, which is deeper than the model $Mnist_s$. The $ResNet101_c$ and $DenseNet121_c$ models are selected by [10] on CIFAR10. Those models are also chosen in ImageNet. For ImageNet dataset, we also test the pretrained models like $EfficientNetB0_i$ [18] and $ShuffleNetV1_i$ [27], which are built into the PyTorch. The models and datasets are summarized in Table 1.

Attack methods The proposed SPND and RSPND are compared with other methods, including FGSM, PGD and MIM. We also test transferability by adding random noise to the original image, which is called RAND. We scale the optimization variables of CW to generate the adversarial examples in regularization space [0, 1].

Parameter settings For non-iterative FGSM, the adversarial samples are calculated according to (3). For other iterative methods, we set the same update step size $\alpha = 0.005$ and the number of iterations $T = 200$. The hyperparameters of the PGD, MIM, and SPND are set to the same and the default parameters are used in multiple datasets. For RSPND, we set the default sample size $g = 2$, $\alpha = 1$ and $\beta = 1$ in Algorithm 2. Considering the L_∞ distances, we take five points at equal intervals from 0 to 0.4. In the regularization space, 0.1 represents 25 perturbations in the RGB space for MNIST. For CIFAR10 and ImageNet, 0.1 roughly represents 5 perturbations in the RGB space.

Table 1 The accuracy of pre-trained models on datasets

Dataset	Size	Model	Accuracy(%)
MNIST	3200	$Mnist_s$	98.72
		$Mnist_b$	98.62
CIFAR10	5120	$ResNet101_c$	91.79
		$DenseNet121_c$	91.99
ImageNet	3200	$ResNet101_i$	75.50
		$DenseNet121_i$	75.00
		$EfficientNetB0_i$	77.25
		$ShuffleNetV1_i$	73.75

4.2 Performances

Under the constraints of different disturbances, the success rates of white-box attack and black-box attack are evaluated for attack methods. Figure 3 demonstrates the results of attack methods on the classification networks. The white-box models are $Mnist_s$, $ResNet101_c$ and $ResNet101_i$ and the black-box models are $Mnist_b$, $DenseNet121_c$ and $DenseNet121_i$. From Fig. 3, we can draw the following conclusions. First, for adversarial sample, the attack success rates are quite different in the white-box model and the black-box model. When the adversarial samples are transferred to the black-box model, the attack success rate drops sharply. Those adversarial examples generated by attack methods are still more deceptive than RAND due to the transferability. Second, for attack methods, the performance of SPND is comparable to other methods (such as PGD, MIM) on white-box attacks, but outperforms those methods on black-box attacks. Third, for pre-trained models, the models on MNIST are more robust than the models on CIFAR10 and ImageNet. As shown in Fig. 3a and d, the black-box model is able to withstand the adversarial examples with the 25/256 perturbations in the RGB space of MNIST dataset.

As shown in Table 2, we investigate the performances of attack methods on different architecture models. DenseNet, EfficientNet and ShuffleNet represent the $DenseNet121_i$, $EfficientNetB0_i$ and $ShuffleNetV1_i$, respectively. Adversarial examples generated by SPND and RSPND have superior transferability in the black-box models. As different architecture models have regional homogeneity in Li et al. [12], the adversarial examples are robust to black-box models.

4.3 Ablation experiments and Hyperparameters analysis

The influence of LRS In some studies [19, 23], local random samples have been proposed to improve the transferability of attack methods, e.g. PGD. However, we find that RSPND has two main drawbacks as a simple ensemble method for SPND. We evaluate the LRS in terms of the performance and speed. First, the above experiments demonstrate that LRS can improve the transferability of adversarial examples when comparing SPND and RSPND with the group size $g = 2$. Furthermore, we investigate the group size for RSPND in Table 3. A larger group size does not lead to more transferability for adversarial examples. Second, the running speed of RSPND is linearly related to the size of the local random sample. As shown in Table 4,

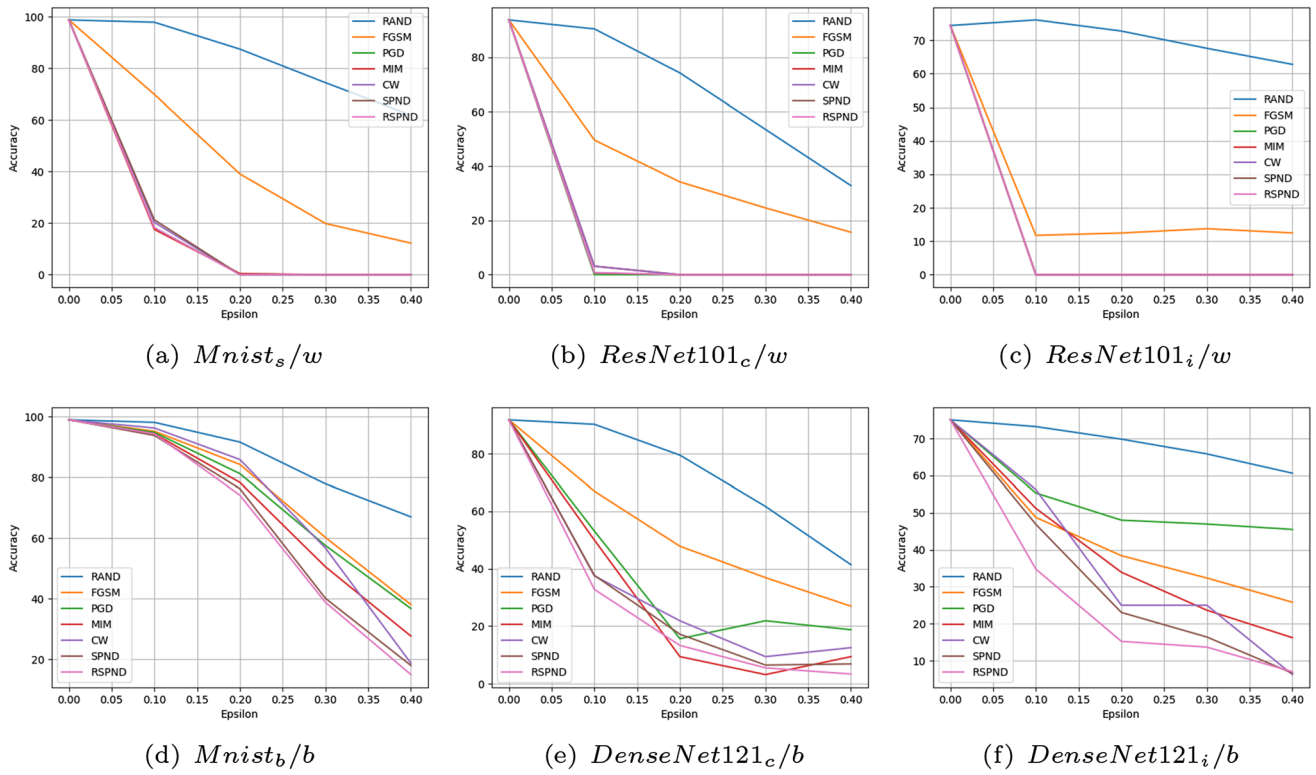


Fig. 3 A series of attack methods are evaluated at different L_∞ distances. The first row indicates that the adversarial samples are generated and evaluated on the white-box models. The second row represents that the adversarial samples are applied to the black-box models

Table 2 The accuracy (%) of the black-box models with the $L_\infty = 0.3$ distance on ImageNet

Models	DenseNet	EfficientNet	ShuffleNet
FGSM	37.03	39.47	33.75
PGD	49.00	53.22	45.25
MIM	26.97	27.00	24.94
CW	33.66	37.50	30.36
SPND	16.28	18.56	14.50
RSPND	**13.66**	**15.88**	**11.78**

Bold values indicate the optimal result on each indicator

Table 3 The accuracy (%) of the black-box $DenseNet121_c$ with various sampling sizes and L_∞ distances of RSPND attack

Size	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.4$
1	41.01	14.25	5.85	6.25
2	**32.81**	**13.28**	**5.46**	**3.32**
3	36.81	14.68	6.44	6.44
4	35.54	13.28	9.37	5.42

Bold values indicate the optimal result on each indicator

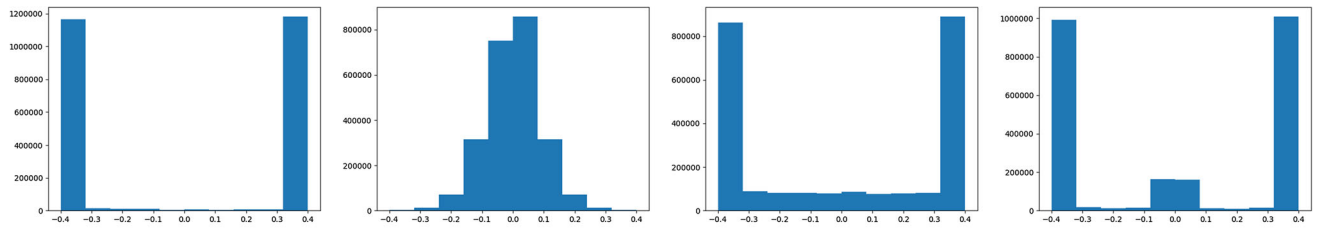
the speed of SPND is similar to other methods. The cost of RSPND with the group size $g = 2$ is almost twice that of SPND. Therefore, LRS only brings limited improvement with the group size $g = 2$ or $g = 3$.

Table 4 The average time (s) of generating one adversarial example for attack methods

Model	PGD	MIM	CW	SPND	RSPND
$Mnist_b$	0.01	0.01	0.01	0.01	0.02
$DenseNet_c$	0.78	0.80	0.79	0.78	0.19
$DenseNet_i$	1.06	1.05	1.05	1.06	2.08

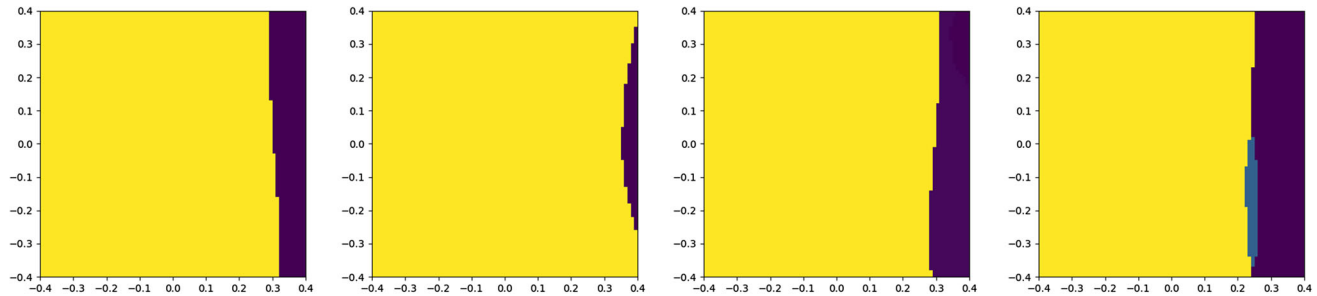
The analysis of perturbations The main differences among these attack methods lie in the directions of perturbations. SPND overcomes the "box constraint" which hinders optimizing the perturbation. In Fig. 4, we draw the distributions of those perturbations. SPND tends to push each pixel to the constraint boundary. The bigger perturbations on the entire image are vital to the transferability of adversarial examples as the heat-maps are different for models in Fig. 1.

Then, we investigate the directions of the generated perturbations for attack methods. In Fig. 5, with the distance $L_\infty = 0.2$, we choose a sample in which all methods fail to generate a valid adversarial sample. Following the settings in Fig. 3, the adversarial sample is generated by



(a) FGSM (b) PGD (c) MIM (d) SPND

Fig. 4 Under the $L_\infty = 0.4$ constraint, the distributions of the L_2 distances are shown with attack methods on $DenseNet121_i$ model. The L_2 distances are calculated for each pixel between original and adversarial images on 16 samples



(a) FGSM (b) PGD (c) MIM (d) SPND

Fig. 5 Church window plots [9] are generated on the black-box model $ResNet121_i$ with attack methods. The colours indicate the classes. The centre of each plot is the same original image x. At coordinate (h, v) within the plot, each pixel indicates the class output by $f_b(x + hu_1 + vu_2)$, where u_1 and u_2 are orthogonal unit vectors that span a 2-D subspace of R^d. The x-axis represents the distance following the direction of u_1 disturbance, and the y-axis represents the distance following the direction of u_2 disturbance. In each figure, u_1 is the normalized direction following the disturbance vector defined by the attack methods, while u_2 is a direction chosen uniformly at random among those orthogonal to u_1

attacking the white-box model $ResNet101_i$. We normalize the perturbation to acquire the direction. We evaluate those directions on the black-box model $ResNet121_i$. We construct the adversarial samples along the perturbation direction and its orthogonal direction. We colour the predicted labels of those adversarial examples in the orientation plane. These visualizations are called church window plots [9]. The centre of each figure is the correct classification result for the original image. The directions of disturbance about SPND are closer to the classification boundaries of models. Therefore, the adversarial examples generated by SPND have better transferability in terms of the magnitude and direction of perturbations.

Table 5 The accuracy (%) of the model $Mnist_b$ finetuning by adversarial training on the $L_\infty = 0.3$ distance

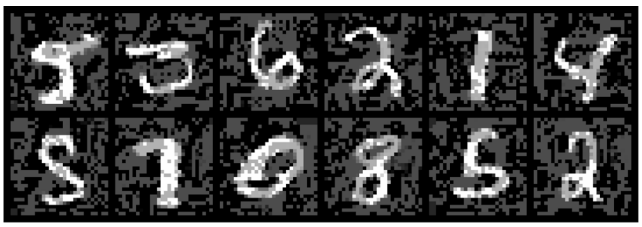
Methods	FGSM	PGD	MIM	CW	SPND
PGD	96.38	85.28	84.14	83.17	81.61
SPND	97.03	79.79	85.35	87.54	90.69

4.4 Discussion and visualization

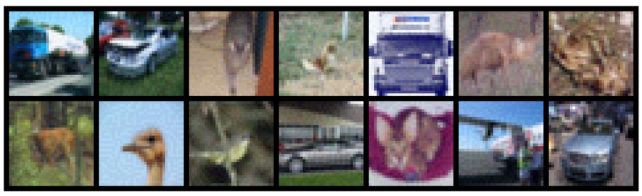
Adversarial training with SPND To defence the attack methods, adversarial training is an effective way to resist the adversarial samples [13]. We replace PGD with SPND to generate adversarial samples for adversarial training. On the $L_\infty = 0.3$ distance, we train the model with the original image and adversarial samples. We evaluate the model as the black-box model. First, Table 4 demonstrates the speed of generating adversarial examples for attack methods. The generating speed of SPND ensures the effectiveness of adversarial training. Then, in Table 5, we finetune the same model $Mnist_b$ with PGD and SPND adversarial training methods for 10 epochs. Other attack methods generate new adversarial examples by attacking the white-box model $Mnist_s$. Table 5 shows the accuracy of those adversarial examples for the defence model $Mnist_b$. SPND improves the model's resistance to other unknown attack methods by adversarial training.

Visualization of adversarial samples Fig. 6 visualizes some adversarial samples on related datasets. The perturbation is negligible with the $L_\infty = 0.2$ distance, while the perturbation is likely noticed under the $L_\infty = 0.3$ distance. To improve the transferability of adversarial examples,

(a) The samples on MNIST, $\epsilon = 0.2$



(b) The samples on MNIST, $\epsilon = 0.3$



(c) The samples on CIFAR10, $\epsilon = 0.2$

(d) The sample on CIFAR10, $\epsilon = 0.3$

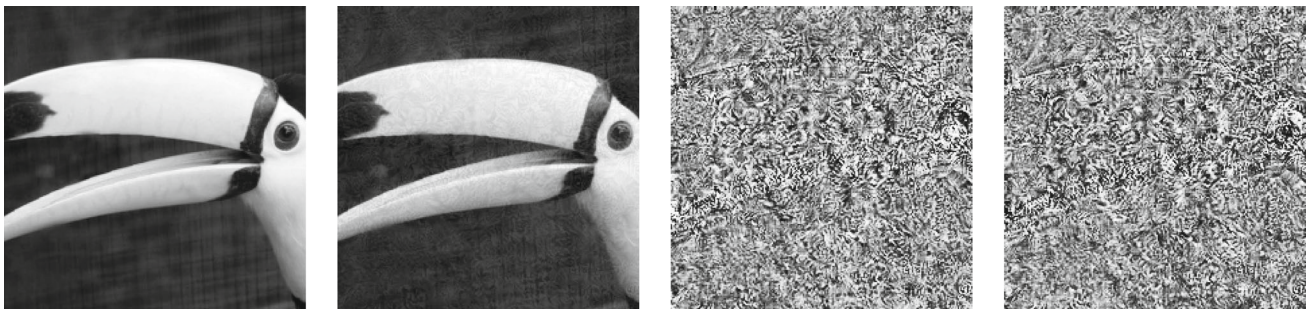
(e) The samples on ImageNet, $\epsilon = 0.2$

(f) The samples on ImageNet, $\epsilon = 0.3$

Fig. 6 Adversarial samples generated by SPND are visualized under the $L_\infty = 0.2$ and 0.3 distances on various datasets

appropriately increasing the intensity of perturbation is an effective method. Although adversarial samples with large perturbations are recognized by humans, the model cannot distinguish maliciously tampered samples. Adversarial training can improve the robustness of the model with small perturbations. For adversarial examples with large perturbations, detecting malicious inputs is an effective method to protect the model. Drenkow et al. [7] present a defence technique to distinguish the clean and adversarial examples across a set of subspaces by random projections.

Visualization of positive and negative disturbances It is necessary to study the disturbance of SPND in details. Figure 7 shows the positive and negative disturbance about an adversarial image. As shown in Fig. 7c and d, positive and negative disturbances are mutually exclusive about adversarial examples, which also are proved in Eq. (17). The positive and negative disturbances are not limited to the area of heat map, but are distributed within the whole image.



(a) Original image (b) Adversarial image (c) Positive perturbations (d) Negative perturbations

Fig. 7 The positive and negative disturbance for a sample image. We select the perturbations on "R" channel of the RGB image as the display to show the magnitude of the positive and negative perturbations. The initial state of z_n and z_p are zeros. In addition, to enhance the visualization, the disturbances are normalized to [0, 255] on the grey space. The white pixels in the plot correspond to a large perturbation

5 Conclusion

In this work, we quantitatively evaluate the attack methods on white-box models and compare the transferability of adversarial samples on black-box models. Within the L_∞ distance, the direction and magnitude of disturbances determine the transferability of adversarial examples. To find the robust disturbances, we propose the SPND and RSPND methods on the perturbation space. Our methods improve the transferability of adversarial samples on black-box models while maintaining a high attack success rate on white-box models. Compared with other attack methods, SPND verifies the hypothesis of robust disturbances. In the future, our attack methods can be used as a benchmark for evaluating the robustness of models. Our method can be extended to perturbation analysis under different constrained distances. Furthermore, how to effectively defend against the SPND and RSPND attacks is the next step of research.

Funding This work is supported in part by the National Natural Science Foundation of China under Grant Nos. (62276127).

Data availability The datasets generated during and analysed during the current study are available in the https://pytorch.org/vision/stable/datasets.html website.

Declarations

Conflict of interest No potential conflict of interest is reported by the authors.

References

1. Adam P, Sam G, Soumith C et al (2017) Automatic differentiation in pytorch. In: Proceedings of neural information processing systems
2. Akhtar N, Mian A (2018) Threat of adversarial attacks on deep learning in computer vision: a survey. IEEE Access 6:14410–14430
3. Carlini N, Wagner D (2017) Towards evaluating the robustness of neural networks. In: 2017 IEEE symposium on security and privacy (sp), pp 39–57
4. Deng J, Dong W, Socher R et al (2009) Imagenet: a large-scale hierarchical image database. In: Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE, pp 248–255
5. Dong Y, Liao F, Pang T et al (2018) Boosting adversarial attacks with momentum. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 9185–9193
6. Dong Y, Pang T, Su H et al (2019) Evading defenses to transferable adversarial examples by translation-invariant attacks. Proceedings of the IEEE conference on computer vision and pattern recognition
7. Drenkow N, Fendley N, Burlina P (2022) Attack agnostic detection of adversarial examples via random subspace analysis. In: Proceedings of the IEEE winter conference on applications of computer vision, pp 472–482
8. Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572
9. Hazan T, Papandreou G, Tarlow D (2016) Perturbations, optimization, and statistics. MIT Press, Cambridge
10. Huang G, Liu Z, Van Der Maaten L et al (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
11. Krizhevsky A, Hinton G et al (2009) Learning multiple layers of features from tiny images. Citeseer
12. Li Y, Bai S, Xie C et al (2020) Regional homogeneity: towards learning transferable universal adversarial perturbations against defenses. Lecture Notes in Computer Science p 795-813
13. Madry A, Makelov A, Schmidt L et al (2017) Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083
14. Moosavi-Dezfooli SM, Fawzi A, Fawzi O et al (2017) Universal adversarial perturbations. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1765–1773
15. Song Y, Shu R, Kushman N et al (2018) Constructing unrestricted adversarial examples with generative models. In: Advances in Neural Information Processing Systems, pp 8312–8323
16. Su J, Vargas DV, Sakurai K (2019) One pixel attack for fooling deep neural networks. IEEE Trans Evolut Comput 23(5):828–841

17. Szegedy C, Zaremba W, Sutskever I et al (2013) Intriguing properties of neural networks. arXiv preprint arXiv:1312f6199
18. Tan M, Le Q (2019) Efficientnet: rethinking model scaling for convolutional neural networks. In: Proceedings of the IEEE international conference on machine learning, PMLR, pp 6105–6114
19. Tramèr F, Kurakin A, Papernot N et al (2017) Ensemble adversarial training: attacks and defenses. arXiv preprint arXiv:1705.07204
20. Wang X, He K, Hopcroft JE (2019) At-gan: A generative attack model for adversarial transferring on generative adversarial nets. arXiv preprint arXiv:1904.07793 3(4)
21. Wang Z, Guo H, Zhang Z et al (2021) Feature importance-aware transferable adversarial attacks. In: Proceedings of the IEEE International Conference on Computer Vision, pp 7639–7648
22. Wei Z, Chen J, Wu Z et al (2022) Boosting the transferability of video adversarial examples via temporal translation. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 2659–2667
23. Wu L, Zhu Z, Tai C et al (2018) Understanding and enhancing the transferability of adversarial examples. arXiv preprint arXiv:1802.09707
24. Wu W, Su Y, Lyu MR et al (2021) Improving the transferability of adversarial samples with adversarial transformations. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 9024–9033
25. Xiao C, Li B, Zhu JY et al (2018) Generating adversarial examples with adversarial networks. arXiv preprint arXiv:1801.02610
26. Xie C, Zhang Z, Zhou Y et al (2019) Improving transferability of adversarial examples with input diversity. Proceedings of the IEEE conference on computer vision and pattern recognition
27. Zhang X, Zhou X, Lin M et al (2018) Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6848–6856
28. Zhang Y, Ya Tan, Sun H et al (2023) Improving the invisibility of adversarial examples with perceptually adaptive perturbation. Inf Sci 635:126–137
29. Zheng T, Chen C, Ren K (2019) Distributionally adversarial attack. In: Proceedings of the AAAI conference on artificial intelligence, pp 2253–2260