Applied Intelligence (2023) 53:7408–7422
https://doi.org/10.1007/s10489-022-03838-0

Multi-layer noise reshaping and perceptual optimization for effective adversarial attack of images

Zhiquan He[1,2,3] · **Xujia Lan**[4] · **Jianhe Yuan**[5] · **Wenming Cao**[4]

Accepted: 21 May 2022 / Published online: 29 July 2022

Abstract

Adversarial attack aims to fail the deep neural network by adding a small amount of perturbation to the input image, in which the attack success rate and resulting image quality are maximized under the l_p norm perturbation constraint. However, the l_p norm is not accurately correlated to human perception of image quality. Attack methods based on l_0 norm constraint usually suffer from the high computational cost due to the iterative search for candidate pixels to modify. In this work, we explore how perceptual quality optimization can be incorporated into the adversarial attack design and propose a two-stage attack method to reshape the adversarial noise by an initial attack and optimize the visual quality of the attacked images without sacrificing the attack success rate. Specifically, we construct a visual attention network to generate a perceptual attention map to modulate the adversarial noise generated by a base attack method. The network is trained to maximize the visual quality in Structural Similarity Index Metric (SSIM) while achieving the same attack success rate. To improve the image perceptual quality further, we propose a fast search algorithm to perform an iterative block-wise pruning of the adversarial noise. We evaluate our method on the mini-ImageNet dataset against three different defense schemes. The results have demonstrated that our method can achieve better attack performance in image quality, attack success rate, and efficiency than the state-of-the-art attack methods.

Keywords Adversarial attack · Image visual quality · Deep convolution neural networks · Image classification

1 Introduction

Despite the remarkable success of deep neural networks (DNN) in various computer vision tasks such as image classification, object detection, and semantic segmentation, DNNs are found vulnerable to adversarial attacks [25]. An adversarial sample is a carefully crafted image to fool the target network by adding small perturbations onto the original clean image [2, 13, 23]. This has raised serious concerns for the security of deep learning networks as they have been extensively used in various applications that require high levels of robustness and security, such as face recognition for identification and authentication, autonomous driving, safety inspection, and surveillance. Therefore, adversarial attacks and defenses of deep neural networks have recently emerged as an important research task in deep learning.

✉ Zhiquan He
zhiquan@szu.edu.cn

Extended author information available on the last page of the article.

Recently, a number of dense adversarial attack methods have been developed, such as FGSM [9], BIM [15], and PGD [17]. The main objective of these methods is to maximize the attack success rate under a l_2 or l_∞ norm noise constraint [17]. However, the resultant attacked images ususally have low preceptual qualities.

Sparse attack methods based on l_0 norm try to modify as few pixels as possible to attack the image without limiting the noise magnitude. These methods usually suffer from low efficiency in achieving a high attack success rate as it is very computationally intensive to search the image space for the candidate pixels [4, 7, 12]. To alleviate the problem, JSMA [21] and GreedyFool [7] predict a saliency or distortion map using a trained network to guide the search process. However, these maps are not directly optimized towards image quality and attack success rate. On the other hand, attacking based on the l_0 norm constraint alone does not guarantee the high visual quality of the attacked images.

The problem is to design an adversarial attack method to simultaneously optimize the attack success rate, visual qualities of the attacked images, and time efficiency or attack complexity that can be measured by the number of model

inferences. To this end, we propose to take advantage of the high attack efficiency and success rate of those l_2 or l_∞ constraint based attack methods, and reshape the perturbation noise to optimize the image quality while keeping the attack success rate. However, it has been well recognized that the l_p norm of image noise is not accurately correlated to human perception of image quality [23, 32]. In the literature, the structural similarity index (SSIM) has been demonstrated to be an effective metric for perceptual image quality [20, 23, 28, 31]. Perceptual color distance is also a good metric for human perception [16, 32]. Figure 1 shows two adversarial examples which have almost the same Peak Signal to Noise Ratio (PSNR) between the original images (left) and the attacked images (right). However, we can clearly see that the adversarial noise in (a) is much more visible and annoying than that in (b), which corresponds to the significant difference between the SSIM values. We recognize that, in different regions of the image, the adversarial noise has different levels of visibility. Or specifically, with the same amount of noise, the SSIM values of different image regions are quite different. For example, the SSIM values of the structural regions are often higher than those in the smooth image regions. This motivates us to argue that the perceptual quality of the adversarial images can be improved by modulating the adversarial noise with the perceptual weights or noise sensitivity levels of different image regions.

Based on this, we propose to develop a perceptually optimized noise reshaping (PONS) scheme to reshape the adversarial noise generated by a base attacker and optimize the visual quality of the attacked images while achieving the same attack success rate. Specifically, in the first stage, we use a convolution SSIM model to calculate the SSIM value between a clean image and its attacked version. Guided by this SSIM visual quality module, the proposed method learns a perceptual attention network that predicts a perceptual sensitivity map to modulate the adversarial noise in the input image, aiming to maximize the visual quality while achieving the same attack success rate. The perceptual attention network and attack method are jointly trained.

Within the context of image classification, we recognize that the decision of the network is binary with the network inference score being compared to a decision threshold. This implies that some regions of the adversarial noise can be removed or pruned to further improve the perceptual quality while ensuring that the classification score remains above the threshold to achieve the same adversarial state. To this end, in the second stage, we propose a fast search algorithm to perform an iterative block-wise pruning of the adversarial noise without affecting the attack success rate.

We have tested our method on the mini-ImageNet dataset against different defense methods. Our method is able to significantly improve the image visual quality over the base attack method without significant loss of the attack success rate. Compared with the state-of-the-art attack methods, our method can achieve better attack performance in terms of image quality, attack success rate, and efficiency.

The major contribution of this work can be summarized as follows:

(1) We incorporate the perceptual quality optimization into the adversarial attack method design and propose a two-stage attack method to reshape the adversarial noise generated by an initial attack while achieving the same attack success rate.
(2) We develop a perceptual attention network that learns to predict a perceptual attention map to modulate the adversarial noise so that the SSIM visual quality of the image is optimized without significant loss of the attack success rate.
(3) We propose a fast binary search algorithm to perform iterative block-wise pruning of the adversarial noise to further improve the perceptual image quality, which does not affect the adversarial state of the image.
(4) Our experimental results on mini-ImageNet dataset with different defense schemes have shown that the proposed method is able to significantly improve the attack performance over the state-of-the-art.

2 Related work

2.1 White-box adversarial attack

This section reviews the related white-box adversarial attack methods as this work belongs to this category. In the white-box attack setting, the parameters of the target model are exposed to the attack process. For deep neural networks, Szegedy et al. pointed out an intriguing weakness of them within the context of image classification and developed a box-constrained L-BFGS method to generate adversarial examples [25]. To overcome the computation efficiency problem, Goodfellow et al. proposed the fast gradient sign method (FGSM) by performing a single gradient step [9]. Kurakin et al. extended this method to an iterative version and demonstrated the adversarial examples in physical world scenarios by feeding adversarial images obtained from cell-phone camera to an ImageNet Inception classifier [15]. Dong et al. proposed a broad class of momentum-based iterative algorithms to generate more transferable adversarial examples, and applied momentum iterative algorithms to an ensemble of models [8]. The PGD method developed in [17] also works in an iterative manner. To improve the transferability of adversarial examples, Xie et al. applied random transformations to the input images at each iteration of the attack process to create more diverse input noise patterns [30]. To circumvent the "obfuscated gradients" problem introduced by the defense methods,

(a) SSIM = 0.72 PSNR = 28.04

(b) SSIM = 0.95 PSNR = 28.14

Fig. 1 Comparison between SSIM and PSNR. The first column is the original images. In the second column, we can see that the upper image has much more distortion than the lower one when compared to the originals. However, the two PSNR scores are very close. SSIM score better reflects the visual distortions in images. A higher SSIM score means fewer distortions in the image

Athalye et al. proposed Backward Pass Differentiable Approximation (BPDA) to provide proximate gradient when the true gradient is unavailable [1]. Carlini and Wagner proposed a set of three adversarial attacks in [3] and declared that defensive distillation did not significantly increase the robustness of neural networks. Zhao et al. exploited human color perception and improved C&W [3] by minimizing the perturbation size with respect to perceptual color distance. Croce et al. extended the usual Projected Gradient Descent (PGD) attack to the L_0 norm to generate highly sparse adversarial examples [4].

Sparse attack methods attempt to make the perturbations imperceptible by constraining the magnitude of the adversarial noise using l_p norm, such as the l_0, l_2, and l_∞ norms [22]. To search for a minimal adversarial perturbation for a given image, Moosavi-Dezfooli et al. proposed DeepFool [19] which can generate adversarial examples with less amount of perturbations than the FGSM method [9] while achieving similar attack success rates. An extreme case of minimizing image perturbation is the one-pixel attack method in which only one pixel in the image is changed to fool the classifier. Su et al. achieved an attack success rate of 70.97% on the tested images by changing just one pixel of the input image [24]. SparseFool [18] converts the l_0 constraint problem into an l_1 constraint problem and exploits the boundaries' low mean curvature to compute sparse adversarial perturbations. GreedyFool [7] selects a few of the most effective candidate pixels to modify using a predicted distortion map as guidance. TSAA [12] translates a benign image into an adversarial image by a generator network which is trained to learn the mapping between natural images and sparse adversarial images.

2.2 Defense against adversarial attack

In this work, we evaluate the attack methods against different defense schemes. Here we briefly review the related defense works. Recently, many methods have been developed for defending deep neural networks against adversarial attacks. Adversarial training is a common method to increase the network robustness by adding adversarial examples into the training data [14, 26, 27]. Tramer et al. proposed an ensemble adversarial training method to augment training data with perturbations transferred from other models [27]. Transformation to the input is another way to defend against adversarial attacks, such as bit-depth reduction, JPEG compression, and total variance minimization [10]. The network structure can be modified to improve its robustness to adversarial attacks. Dhillon et al. pruned a random subset of activations according to their magnitude to enhance

network robustness [6]. Cihang et al. proposed a feature denoising method using non-local means or other filters. Along with adversarial training, the adversarial robustness can be substantially improved [29].

3 Method

In this section, we present the proposed perceptually optimized noise reshaping (PONS) method for adversarial attacks.

3.1 Problem formulation

Let X be a natural image, and y^T be the corresponding ground-truth label, also referred to as the target label. A classifier $\Phi_\theta(X) = y$ takes an image X as input and predicts its label $y \in \mathcal{Y}$, where $\mathcal{Y}$ is the output label space. The goal of the adversarial attack is to generate an adversarial noise Z constrained by l_p norm so that the attacked image $X^a = X + Z$ is misclassified by the target model. In this work, the image quality $SSIM(X, X^a)$ is maximized during the attack process. Mathematically, it is written as

$$\begin{aligned} &\arg\max_{Z} SSIM(X, X+Z) \\ &s.t.\ \Phi_\theta(X+Z) \neq \Phi_\theta(X) \\ &s.t.\ ||Z||_p \leq \epsilon, \end{aligned} \tag{1}$$

where ϵ is the perturbation budget, p is the order of matrix norm and it can be 0, 1, 2 or ∞.

3.2 Method overview

In this work, we propose a novel two-stage adversarial attack framework that takes advantage of the high attack efficiency and success rate of a base attack method and reshapes the perturbation noise to optimize the visual quality of the attacked images while maintaining the same attack success rate. As illustrated in Fig. 2, the proposed PONS method uses the SSIM perceptual quality and sensitivity analysis to drive the adversarial noise generation process. In the first stage, we use a convolution model to calculate the SSIM index between the attacked image and the input, and a perceptual attention network to predict the perceptual sensitivity map, which is used to modulate the adversarial noise generated by the base attack method. The attacker network and the perceptual attention map prediction network are learned end-to-end to optimize the SSIM image qualities while achieving successful adversarial attacks. In the second stage, we propose a fast binary search method to perform a block-wise pruning of the adversarial noise based on the perceptual sensitivity map to improve the image qualities further.

3.3 SSIM prediction network and perceptual sensitivity map

SSIM has been extensively used as a metric to measure the perceptual quality of images. Here we give a brief definition. For more details, please go to the reference [28]. Let A and B be the two images being compared. A window

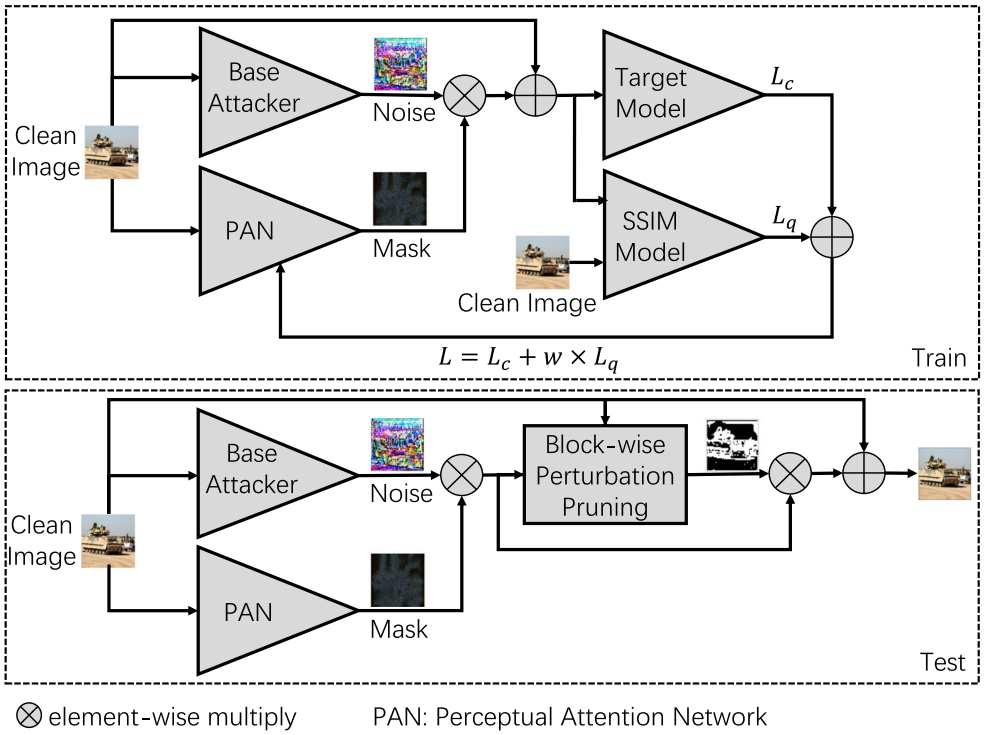


Fig. 2 Diagram of the proposed PONS method. The lower figure is a two-stage attack process. In the first stage, the perceptual attention network (PAN) is used to predict a perceptual attention mask using the extracted features from the images to modulate the adversarial noise generated by the base attack method. In the second stage, to further reduce the perturbation, a block-wise perturbation pruning is applied to find out image blocks to turn off the perturbation. In the upper figure, the perceptual attention network is trained to optimize the image quality (L_q) and attack success rate (L_c), where the SSIM model is used to calculate the SSIM score between the clean image and the adversarial

moves pixel-by-pixel from the top left corner to the bottom right corner of the image. In each step, the local statistics $\Theta(A_j, B_j)$ is calculated within local window j as follows [28]:

$$\theta(A_j, B_j) = \frac{(2 \cdot m_{A_j} m_{B_j} + C_1) \cdot (2 \cdot \sigma_{A_j B_j} + C_2)}{(m_{A_j}^2 + m_{B_j}^2 + C_1)(\sigma_{A_j}^2 + \sigma_{B_j}^2 + C_2)}, \quad (2)$$

where m_{A_j}, m_{B_j}, σ_{A_j}, σ_{B_j}, and $\sigma_{A_j B_j}$ represent the average intensity of image patches A_j and B_j, the standard deviation of A_j and B_j, and covariance between A_j and B_j, respectively. C_1 and C_2 are two small constants introduced to avoid numerical instability. The SSIM index between A and B is defined by [28]

$$\Theta(A, B) = \frac{\sum_{j=1}^{N_s} W(A_j, B_j)\theta(A_j, B_j)}{\sum_{j=1}^{N_s} W(A_j, B_j)}, \quad (3)$$

where N_s is the number of local windows in the image and $W(A_j, B_j)$ is the weights applied to window j [28]. It should be noted that the above computation is highly nonlinear. In this work, we use a PyTorch model (https://github.com/aserdega/ssim-pytorch) to approximate the SSIM function $\Theta(,)$.

We introduce the perceptual sensitivity map to measure the impact of adversarial noise on the image quality at different locations. For image A, B with size of $[W, H, C]$, we propose to use the feature map $F(m, n, c)$, $1 \le m \le W$, $1 \le n \le H$, $1 \le c \le C$, from the last layer of the SSIM calculation network. This feature map represents the impact of the difference between the attacked image and the original clean image on the overall image SSIM quality. The perceptual sensitivity map $\sigma(m, n)$ is defined as

$$\sigma(m, n) = 1 - \frac{1}{C}\sum_{c=1}^{C} |F(m, n, c)|. \quad (4)$$

Figure 3 shows an example of the perceptual sensitivity map. The figure shows that the smooth areas are relatively whiter than those texture-rich regions, which means smooth areas are more sensitive to adversarial perturbation.

3.4 SSIM-optimized adversarial attack

The core idea of our method is to reshape the perturbation noise in adversarial examples that are successfully attacked by a base attack method. The base attacker usually has a high attack success rate. This work considers PGD and its variant BPDA as the base attack method. The PGD is an iterative version of the original FGSM method which adds perturbation along the gradient direction of the loss $\nabla_x L(X, y^T; \theta)$ onto the input X to generate adversarial example as follows [9]:

$$X^a = X + \epsilon \cdot sign(\nabla_X L(X, y^T; \theta)), \quad (5)$$

where $L(\cdot)$ is usually defined as the cross-entropy loss, and ϵ controls the l_∞-norm of the difference between X and X^a. The PGD method iteratively updates the image as [17]

$$X_{t+1}^a = \Gamma_\epsilon[X;\ X_t^a + \alpha \cdot sign(\nabla_X L(X_t^a, y; \theta))], \quad (6)$$

where t is the iteration index, α is the step size, and $\Gamma_\epsilon[X; X^a]$ is a clipping function which makes sure that the largest difference between X and X^a is less than ϵ.

As illustrated in Fig. 3, different image regions have different levels of sensitivity to the adversarial noise. Motivated by this observation, we propose to modulate the adversarial noise by a learned perceptual weight mask $\mathcal{M} = [\mathcal{M}(m, n)]$, as illustrated in Fig. 2. The perceptually modulated PGD attack is given by

$$X_{t+1}^a = \Gamma_\epsilon[X + Z_{t+1} \cdot \mathcal{M}]. \quad (7)$$

where Z_{t+1} is the final perturbation generated by the base attack method. The perceptual mask $\mathcal{M}$ reshapes the adversarial noise such that, in those image regions whose visual quality levels are sensitive to noise, the noise magnitude is reduced.

In our proposed method, this perceptual mask $\mathcal{M}$ is predicted by the perceptual attention network which is learned with the target network, the attack network, and the SSIM network. The output mask has the same size as the input image. The proposed perceptual mask prediction network is based on an auto-encoder-decoder structure with a Resnet-18 backbone [11]. The training process is to maximize the SSIM qualities while achieving successful

Fig. 3 Example of perceptual sensitivity map. Images from left to right are the original, the adversarial, and the corresponding perceptual sensitivity map, respectively. Smooth areas are relatively whiter than texture-rich regions, which means higher sensitivity to noise

attacks of the original images. Specifically, let y^A be the attack target, the incorrect label that the attack forces the classifier Φ_θ to produce, and y^a be the current classifier softmax output. We use the following cross-entropy loss $\mathcal{L}_C$ between the attack target y^A and the actual classifier output y^a

$$\mathcal{L}_C(y^A, y^a) = -\sum_{i=1}^{C} y_i^A \cdot \log_2 y_i^a. \tag{8}$$

The following loss is then used to train the perceptual attention network

$$\mathcal{L}_M = \mathcal{L}_C(y^A, y^a) - \lambda \cdot \Theta(X_{t+1}^a, X), \tag{9}$$

where λ is a weight factor, $\Theta(X_{t+1}^a, X)$ measures the SSIM quality between the current attacked image X_{t+1}^a and the original image X.

3.5 Block-wise binary pruning of adversarial noise

In the above section, we have learned a network to predict a perceptual attention mask $\mathcal{M}(m, n)$ to modulate the adversarial noise. It should be noted that this learned mask $\mathcal{M}(m, n)$ is a continuous value that aims to maximize the overall SSIM image quality of the attacked image. This prediction and quality optimization is a global process, and all entries of the mask $\mathcal{M}$ are jointly predicted from one network inference. From our experiments, we observe that the value of each individual entry $\mathcal{M}(m, n)$ can be fine-tuned to further improve the perceptual quality of the attacked image. To this end, we propose a fast and efficient method, called *block-wise binary pruning* to perform local adjustment of the perceptual mask. Specifically, we equally partition the mask $\mathcal{M}$ into non-overlapping blocks and, for each block, the proposed algorithm will make the following binary decision: the adversarial noise within this block being kept (indicated by 1) or totally removed / pruned (indicated by 0). Conceptually, we need to remove or turn off the adversarial noise in those image blocks with high levels of sensitivity to noise to maximize the perceptual quality while achieving a successful attack of the image. In (4), we have derived the perceptual sensitivity map from the SSIM index calculation network. We propose to use this perceptual sensitivity map $[\sigma(m, n)]_{W \times H}$ to guide the block-wise binary pruning of adversarial noise. Each entry $\sigma(m, n)$ of this map represents the perceptual sensitivity of an image pixel. The central task here is: we need to select a subset of these image blocks to remove or prune their adversarial noise, i.e., setting their attack noise to zeros, so that the SSIM perceptual quality of the attacked image is maximized without changing the adversarial state. Clearly, it is very computationally intensive to search for this subset of image blocks. Most importantly, at each search step, we need to run the classification network to check whether the resulting image is still successfully attacked or not. Therefore, the number of search steps has to be very limited.

To address this issue, we propose a fast binary search algorithm to obtain a sub-optimal solution. Specifically, we first sort the perceptual sensitivity value $\sigma[k]$ of all blocks in an ascending order, denoted by $\sigma[k]$, $1 \le k \le \frac{W}{w} \times \frac{H}{h}$, where $\sigma[k]$ is the mean value of $\sigma(m, n)$ in the k-th image block with size $[w, h]$. We aim to find a decision threshold σ_P such that all adversarial noise in image blocks with $\sigma[k] > \sigma_P$ will be removed while the noise in the remaining blocks will be kept. With the image blocks sorted according to the perceptual sensitivity, this search can be efficiently implemented with the following binary search method. Specifically, let $\{I_t\}$ be the sequence of search positions (or k values). Here, t is search step index. At search step t, we set $\sigma_P = \sigma[I_t]$ or remove adversarial noise in all image blocks with indices $k > I_t$. Let the corresponding pruned adversarial noise be $Z[I_t]$. We then evaluate the classification network on the attacked image $X + Z_t$ to produce the classification output $\Phi_\theta(X + Z[I_t])$. If the image is successfully attacked, we denote it by $\Gamma(X + Z[I_t]) = 1$. Otherwise, $\Gamma(X + Z[I_t]) = 0$.

Initially, we set $I_0 = 0$ and $I_1 = \frac{W}{w} \times \frac{H}{h}$. Certainly, we have $\Gamma(X + Z[0]) = 0$ since the adversarial noise in all blocks are removed and $X + Z[0]$ becomes the original image. If $\Gamma(X + Z[1]) = 0$, which means that the original attack is not successful, the search stops, and the adversarial attack fails. Otherwise, the proposed binary search process continues as follows: at search step t, let

$$I_t^+ = \min\{I_j |\ \Gamma[X + Z[I_j]) = 1, j < t\}, \tag{10}$$

which is the smallest index with successful attack, and

$$I_t^- = \max\{I_j |\ \Gamma[X + Z[I_j]) = 0, j < t\}, \tag{11}$$

which is the largest index with failed attack. Then, in our proposed binary search, we set

$$I_{t+1} = \frac{I_t^+ + I_t^-}{2}. \tag{12}$$

Once the I_{t+1} is determined, we will evaluate $\Gamma(X + Z[I_{t+1}])$ and repeat the above steps until $I_{t+1} - I_t < 1$. The overall binary search-based method is outlined as Algorithm 1. Figure 4 shows four examples of the above binary search process. The horizontal axis is the total number of blocks where the adversarial noises are removed. As the number of pruned blocks increases, the SSIM quality of the attacked image improves significantly. The ending value represents the point where the attack fails, which is the target value that the binary algorithm aims to find.

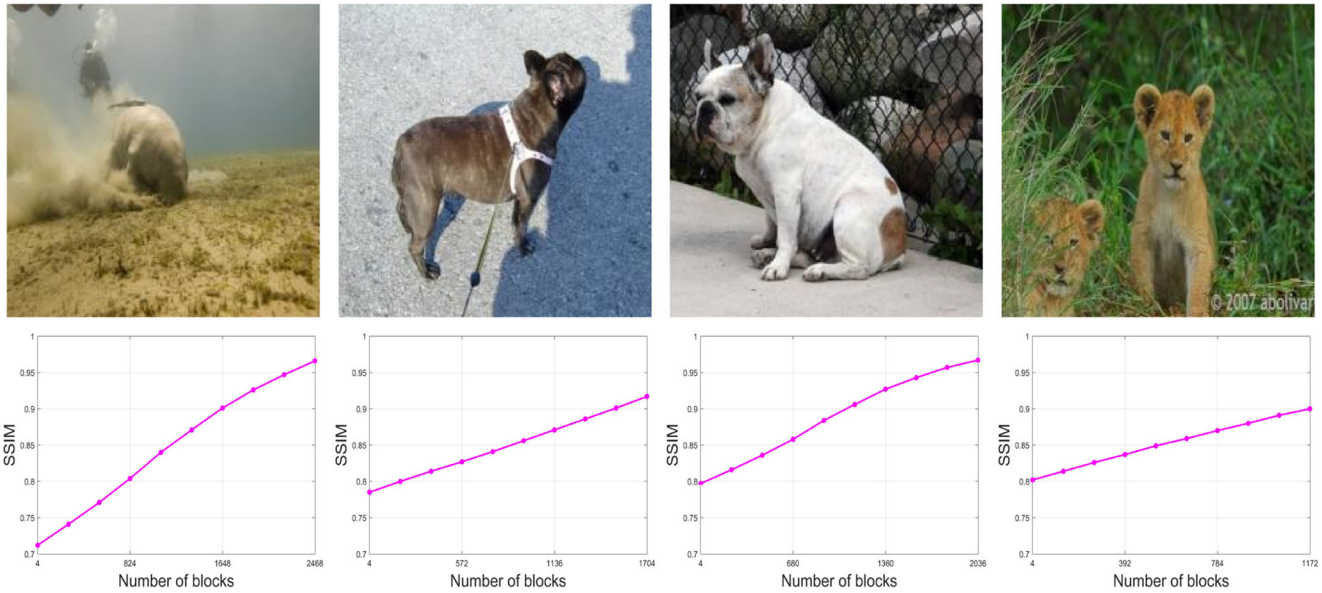


Fig. 4 SSIM scores keep increasing when turning off the noise on more and more image blocks until the classification result by the model changes. The X-axis is the number of image blocks where the perturbations are turned off. The rightmost number in each figure is the switching point that the classification result flips

Algorithm 1 Block-wise Perturbation Pruning

Input: x: input test image. $f_T()$: forward process of the target model, including the defense module. z: initial input perturbation

Output: x^a

Divide image x into non-overlapping blocks $\{b_1, .., b_N\}$
Calculate the block SSIM scores according to (4), and use the scores to sort the blocks in ascending order.
Let $left = 1, right = N$
repeat
    $m = (left + right)/2$, and remove the noise on blocks of $\{b_m, .., b_N\}$, giving x_m^a
    if $f_T(x_m^a) == f_T(x + z)$ **then**
        $right = m$
    else
        $left = m$
    end if
until $left == right$

3.6 Attack complexity analysis

Besides the computation consumed by the base attack method, the extra complexity of our method comes from the inference of the perceptual attention prediction network and the binary search for adversarial noise pruning. The first part is a one-time cost, which is relatively small, depending on the model complexity. The complexity of the second part is the number of search steps multiplied by the complexity of the target classification network. Theoretically, our binary search in Algorithm 1 has the complexity of $log_2(N_b)$, where $N_b = \frac{W}{w} \times \frac{H}{h}$ is the number of image blocks. In our experiments, the image size is 224×224, the block size is 4×4, so the total number of blocks is $N = 3136$. Therefore, the maximum number of model inferences is 12.

4 Experiments

In this section, we provide extensive experimental results to evaluate the performance of our proposed perceptual attention-guided visual quality optimization algorithm for adversarial attacks.

4.1 Experimental setup and datasets

In this section, we compare the performance of our method to the state-of-the-art attack methods with different defensive schemes. The methods include two dense attack methods PGD and BPDA, and four sparse attack methods Perc_CW [32], $\text{PGD}_{L0+\sigma}$ [4], GreedyFool [7] and TSAA [12]. The performance is studied in metrics of the SSIM score between the clean image and the adversarial, attack success rate, and attack efficiency in terms of the number of target model inferences.

As Perc_CW is optimized on the color distance score, we also use perceptual color distance to measure the quality of the adversarial images [32]. To compare to Perc_CW and $\text{PGD}_{L0+\sigma}$, we only compare the adversarial image qualities at five different attack success rates as it is time-consuming to get the desired attack success rate due to the

high dimensional search space of their hyperparameters. For example, Perc_CW has five parameters that can affect the attack success rate. We keep running these two tools with different hyperparameter settings and use the attack result if its attack success rate is within [0.85 0.95]. GreedyFool [7] and TSAA [12] are l_0 norm based attack methods. GreedyFool requires a large number of iterations to obtain good attack results. So we only evaluate it using one model. TSAA is slightly different from the others as it is designed mainly for the black-box attack to improve the transferability of the adversarial examples. We still show its results for comparison.

The dataset is the mini-ImageNet [5] which consists of 60,000 images for 100 classes. We select 20 classes, and for each class, 100 images are randomly selected as test images, and the rest is used for training.

4.2 Experimental results

4.2.1 Resnet-34 with adversarial training for defense

In this experiment, the target model is Resnet-34 which is adversarially trained on the training dataset, and the default classification accuracy is 76%. During adversarial training, the adversarial examples are generated by PGD with default parameters $\epsilon = 0.05$, $alpha = 0.01$, $num\_iter = 10$, where $alpha, num\_iter$ are the step size and the number of iterations, respectively. The parameter of λ in (9) is set to 0.001 during training of the mask prediction network. Table 1 shows the average SSIM score of PGD and our method at different values of ϵ. The scores are calculated over all tested images that are correctly classified by the target model. We can see that the SSIM gain obtained by the proposed PONS algorithm is quite significant. When $\epsilon = 0.03$, which is quite small, the SSIM gain is about 7%. For the loss of attack success rate, it is quite small.

Table 1 Comparison of SSIM and attack success rate between PGD and our method

ϵ	0.03	0.05	0.06	0.07	0.08	0.09	0.1
SSIM quality of the attacked image							
PGD	0.88	0.78	0.74	0.71	0.69	0.67	0.66
PONS	0.95	0.94	0.93	0.93	0.93	0.93	0.93
Gain	+0.07	+0.16	+0.19	+0.24	+0.25	+0.26	+0.27
Attack Success Rate							
PGD	0.75	0.91	0.94	0.96	0.97	0.97	0.97
PONS	0.74	0.89	0.93	0.95	0.96	0.97	0.97

The model Resnet-34 is adversarially trained for defense. Our method has consistently achieved substantial improvement in SSIM over PGD, especially when $\epsilon \geq 0.03$, without significant loss of attack success rate

Table 2 Comparison of SSIM score against Perc_CW and $\text{PGD}_{L0+\sigma}$ under the same attack success rate. The target model is Resnet-34 with adversarial training

Attack success rate	0.85	0.87	0.89	0.94	0.95
Perc_CW [32]	0.90	0.85	0.69	0.60	0.59
PONS	0.94	0.94	0.94	0.94	0.94
Gain	+0.04	+0.09	+0.25	+0.34	+0.35
Attack success rate	0.86	0.88	0.90	0.92	0.94
$\text{PGD}_{L0+\sigma}$ [4]	0.89	0.89	0.87	0.87	0.84
PONS	0.94	0.94	0.94	0.94	0.94
Gain	+0.05	+0.05	+0.07	+0.07	+0.10

Table 2 shows the average SSIM score of Perc_CW [32] and $\text{PGD}_{L0+\sigma}$ [4] and ours. Our method improves the SSIM score and outperforms these two state-of-the-art methods by large margins, especially at high attack success rates. For example, for Perc_CW, the SSIM gain of our method is 0.35 at the attack success rate 0.95, and for $\text{PGD}_{L0+\sigma}$, when the attack success rate is 0.94, the SSIM gain is 0.1.

4.2.2 Resnet-101 with feature denoising for defense

In the following experiment, the target model is the Resnet-101 network which is modified by adding *Feature denoising* layers [29] to reduce the effect of adversarial noise. The model is also adversarially trained using PGD attacked images. The model has a classification accuracy of 80% on clean images. Table 3 shows the average SSIM score of the baseline PGD method and our PONS method at different values of ϵ. Similar to the previous experiment, our method has significantly improved the SSIM quality. For the attack success rate, the loss is very small, mostly less than 1%. For tests with $\epsilon = 0.06, 0.07, 0.09, 0.1$, the drop of success rate is zero. Table 4 shows the average SSIM scores of Perc_CW, $\text{PGD}_{L0+\sigma}$ and our method, where we can see

Table 3 Comparison of SSIM and attack success rate between PGD and our method

ϵ	0.03	0.05	0.06	0.07	0.08	0.09	0.1
SSIM Quality of the Attacked Image							
PGD	0.91	0.84	0.80	0.78	0.75	0.74	0.72
PONS	0.96	0.95	0.94	0.94	0.93	0.93	0.93
Gain	+0.04	+0.07	+0.17	+0.20	+0.24	+0.26	+ 0.29
Attack Success Rate							
PGD	0.66	0.85	0.89	0.90	0.92	0.93	0.93
PONS	0.65	0.84	0.89	0.90	0.91	0.93	0.93

The target model is Resnet-101 modified by attaching feature denoising layers for defense. Our method has consistently achieved substantial improvement in SSIM over PGD, especially when $\epsilon \geq 0.03$, without significant loss of attack success rate

Table 4 SSIM comparison against Perc_CW and $PGD_{L0+\sigma}$ under the same attack success rate

Attack success rate	0.86	0.90	0.92	0.92	0.93
Perc_CW [32]	0.96	0.90	0.91	0.89	0.86
PONS	0.94	0.94	0.94	0.94	0.93
Gain	-0.20	+0.04	+0.03	+0.05	+0.07
Attack success rate	0.85	0.87	0.89	0.91	0.93
$PGD_{L0+\sigma}$ [4]	0.90	0.90	0.89	0.89	0.87
PONS	0.95	0.94	0.94	0.94	0.93
Gain	+0.05	+0.04	+0.05	+0.05	+0.06

The target model is Resnet-101 with feature denoising layers for defense

that our method has achieved much higher SSIM scores, especially when the attack success rate is greater than 0.9.

4.2.3 Resnet-50 with input transformation for defense

Input transform has been demonstrated as an effective method for adversarial defense [10]. In this experiment, the input transformation is bit reduction [10], which removes the least 5 significant bits of each pixel value. Or, equivalently, the pixel values are quantized into the set {0, 32, 64, 128, 160, 192, 224}. It should be noted that this bit reduction processing is not differentiable. In this case, the BPDA method uses an identity function during the gradient backward propagation process. During the BPDA attack, the number of steps is set to 200, and the learning rate is set to 0.1. The target model is Resnet-50, which is also adversarially trained using PGD-attacked images. The baseline classification accuracy on clean images is 77.3%. Table 5 shows the average SSIM score of BPDA and our method at different values of ϵ. Our PONS method can significantly improve the SSIM visual quality of the images attacked by BPDA. In the meantime, the drop of attack success rate remains very small. Again, Table 6 compares the average SSIM against Perc_CW and $PGD_{L0+\sigma}$. The improvement is consistent and significant.

Table 5 Comparison of SSIM score and attack success rate between BPDA and ours

ϵ	0.03	0.05	0.06	0.07	0.08	0.09	0.1
SSIM Quality of the Attacked Images							
BPDA	0.94	0.82	0.75	0.70	0.64	0.60	0.56
PONS	0.98	0.95	0.92	0.90	0.88	0.86	0.85
Gain	+0.04	+0.13	+0.17	+0.20	+0.24	+0.26	+0.29
Attack Success Rate							
BPDA	0.53	0.74	0.85	0.91	0.95	0.98	0.99
PONS	0.52	0.74	0.85	0.90	0.94	0.96	0.98

The target model is Resnet-50 with the defense method of bit-depth reduction. The SSIM improvement is quite significant, especially when $\epsilon \geq 0.03$. Moreover, the loss of attack success rate is ignorable

Table 6 SSIM score comparison against Perc_CW and $PGD_{L0+\sigma}$ under the same attack success rate

Attack success rate	0.88	0.89	0.91	0.92	0.93
Perc_CW [32]	0.90	0.88	0.56	0.55	0.55
PONS	0.91	0.91	0.90	0.89	0.88
Gain	+0.01	+0.03	+0.34	+0.34	+0.33
Attack success rate	0.85	0.88	0.90	0.93	0.95
$PGD_{L0+\sigma}$ [4]	0.90	0.88	0.86	0.83	0.80
PONS	0.92	0.91	0.90	0.89	0.87
Gain	+0.02	+0.03	+0.04	+0.04	+0.03

The target model is Resnet-50 with the defense method of bit-depth reduction

4.2.4 Color distance comparison

We notice that Perc_CW is optimized on the color distance or difference [16, 32] between the attacked image and the original one. Table 7 shows the color distance scores between Perc_CW and our method at different attack success rates in each of the previous tests. Smaller color distance with respect to the natural image means better image quality of the adversarial. We can see that even though our method is not optimized on the color distance, the reported average color distance is still comparable to

Table 7 Color distance comparison at different attack success rates between Perc_CW and our method in each of the previous tests

Resnet-34 with Adversarial Training Defense					
Attack success rate	0.85	0.87	0.89	0.94	0.95
Perc_CW	0.89	0.66	1.18	4.13	4.49
PONS	**0.82**	0.84	**0.86**	**0.97**	**1.00**
Gain	-0.07	+0.18	-0.32	-3.16	-3.49
Resnet-101 with Feature Denoising Defense					
Attack success rate	0.86	0.90	0.91	0.92	0.93
Perc_CW	0.55	0.93	0.93	1.00	1.18
PONS	0.83	**0.89**	**0.92**	**0.94**	**0.97**
Gain	+0.28	-0.04	-0.01	-0.06	-0.21
Resnet-50 with Bit-depth Reduction Defense					
Attack success rate	0.88	0.89	0.91	0.92	0.93
Perc_CW	0.91	1.00	4.73	4.90	5.38
PONS	1.00	1.03	**1.11**	**1.15**	**1.19**
Gain	+0.09	+0.03	-3.62	-3.75	-4.19

A smaller color distance score means better image quality. The color distance scores in the table are the original scores divided by 1000

that of Perc_CW. For most of the cases, our method is even better.

4.2.5 Comparison to GreedyFool and TSAA

TSAA [12] and GreedyFool [7] generate adversarial samples based on l_0 norm constraint which try to modify fewest number of image pixels. We study their attack performance when the perturbation magnitude, i.e., l_∞ norm constraint, is also applied. Table 8 shows the attack result using the model Resnet-34 in Section 4.2.1 when ϵ changes from 0.1 to 0.5 and 1.0, in which $\epsilon = 1.0$ means the adversarial attack is fully under l_0 norm constraint. The three numbers in the table are the average SSIM, attack success rate, and the number of model inferences. For TSAA, the performance improves when ϵ increases from 0.1 to 0.5 and 1.0, achieving the best average SSIM of 0.65 and attack success rate of 0.53. For GreedyFool, the number of iterations used to search for the candidate pixels affects its performance significantly. When $\epsilon = 1.0$, GreedyFool achieves the best average SSIM (0.97) and attack success rate (0.98) within 500 iterations, i.e., the GF(500) in the table. However, the actual average number of model inferences is as high as 669, which costs a significant amount of time. Compared to Table 1, our method can achieve the average SSIM of 0.93 and attack success rate of 0.97 with only 12 model inferences at $\epsilon = 0.1$. In this case, GreedyFool works fully under the l_0 norm constraint. We have further checked the Mean Squared Error (MSE) of the attacked images and found that GF(500) has an average MSE of 104.95, and ours is 80.02. This means that GreedyFool adds much more perturbations to images than our method. For GF(500), when $\epsilon = 0.5$, the attack success rate drops to 0.89 at the cost of 1009 model inferences. When only 12 iterations are

Table 8 Attack performance of TSAA [12] and GreedyFool [7] with different values of perturbation budget

ϵ	0.1	0.5	1.0
TSAA	0.85/0.03/1	0.68/0.34/1	0.65/0.53/1
GF(500)	1.00/0.16/1221	0.97/0.89/1009	0.97/0.98/669
GF(12)	1.00/0.01/30	1.00/0.04/43	1.00/0.08/48

$\epsilon = 1.0$ means the adversarial attack is fully under l_0 norm constraint. The target model is Resnet-34 as in Section 4.2.1. GF(12) is the GreedyFool method with the parameter of iteration set to 12, and GF(500) sets the parameter to 500. The three numbers in the table are the average SSIM, attack success rate, and the number of model inferences

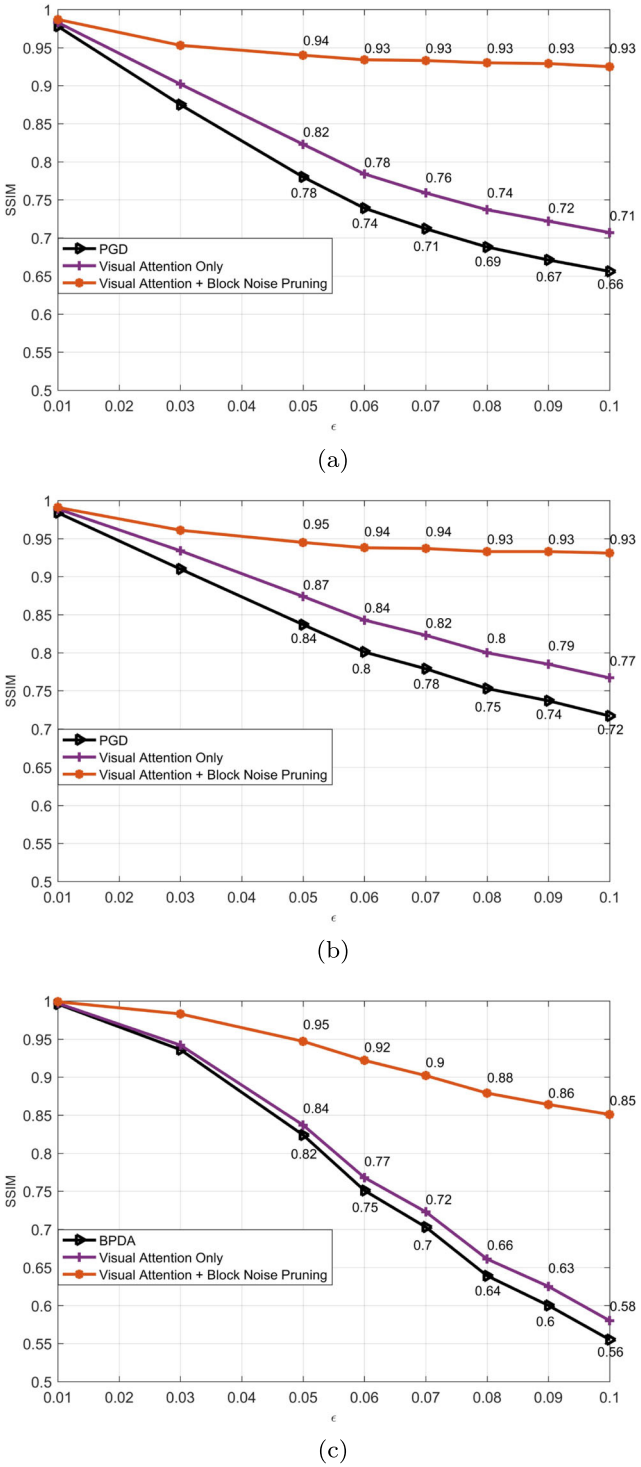


Fig. 5 Figure (a), (b), and (c) shows the SSIM improvement by the perceptual attention prediction and block-wise pruning of the adversarial noise. In all three figures, the black line is the base attack method, the brown line is the SSIM score with only perceptual attention prediction, and the orange line shows the SSIM score with both attention mask prediction and block-wise noise pruning

used for GreedyFool, i.e., the GF(12), it fails to attack the images. The success rate is almost zero.

4.3 Ablation studies

In the following experiments, we conduct ablation studies to further understand the performance of the proposed PONS method.

4.3.1 Contribution of algorithm components

Our method has two stages, perceptual attention mask prediction, and block-wise adversarial noise pruning. In Fig. 5, we show the average SSIM improvement brought by each stage in the previous tests. The horizontal axis is the perturbation budget ϵ. In each sub-figure, we show the average SSIM score for (a) the baseline attack method, (b) the baseline method plus the perceptual attention noise reshaping, and (c) the baseline method plus both algorithm modules. In all three tests, we can see that both algorithm modules contribute significantly to the overall performance. The performance gain achieved by the binary pruning is more significant than the perceptual attention noise reshaping, especially for the BPDA attack method. For example, in Fig. 5 (b), at $\epsilon = 0.05$, the SSIM score of the base attack is 0.84, the perceptual attention module increases it to 0.87, and the block-wise perturbation pruning further increases it to 0.95.

4.3.2 Comparison of different block sorting schemes

In our method, we use SSIM sensitivity score to sort the image blocks to find out candidate blocks to turn off the perturbation by the binary search algorithm. Table 9 compares the average SSIM scores when different block sorting schemes are used, namely 1) sorting by the SSIM sensitivity, 2) original order (no sorting), and 3) randomly sorting. From Table 9, we can see that sorting image blocks according to the SSIM sensitivity score achieves the largest improvement of the SSIM score. The random sorting is the least. For example, in the test of BPDA with input transform, when $\epsilon = 0.05$, the average SSIM scores are $0.88, 0.90, 0.95$ respectively, sorting based on the SSIM sensitivity improves the average SSIM by 5 percentage points when compared to the case without any sorting.

4.3.3 Effect of SSIM loss ratio

To train the perceptual attention model, Eq. (9) has two loss terms, the cross-entropy between the attack target and the predicted labels, and the SSIM image quality between the attacked image and the input image. The value of ratio λ affects the quality of the attention mask predicted by the perceptual attention network. More weight on the SSIM term will likely improve the image quality and decrease the attack success rate. We redo the test of Table 1 in Section 4.2.1, but without the block-wise noise pruning. Table 10 shows the performance of average SSIM and attack success rate with different values of λ. We can see that the performance is quite robust when λ increases from 0.001 to 0.5. When it reaches 1.0, the average SSIM increases, and the attack success rate drops significantly, compared with the case of 0.001.

Table 9 SSIM comparison of block-wise perturbation pruning with different block sorting methods in three experiments

$\epsilon =$	0.03	0.05	0.06	0.07	0.08	0.09	0.1
PGD + Adversarial Training Defense							
Random	0.93	0.89	0.87	0.86	0.85	0.85	0.84
Original	0.94	0.91	0.89	0.88	0.88	0.88	0.87
SSIM	0.95	0.94	0.93	0.93	0.93	0.93	0.93
PGD + Feature Denoising Defense							
Random	0.94	0.90	0.88	0.87	0.86	0.86	0.85
Original	0.96	0.93	0.92	0.92	0.91	0.91	0.90
SSIM	0.96	0.95	0.94	0.94	0.93	0.93	0.93
BPDA + Bit-depth Reduction Defense							
Random	0.96	0.88	0.84	0.80	0.76	0.74	0.71
Original	0.97	0.90	0.86	0.84	0.80	0.78	0.76
SSIM	0.98	0.95	0.92	0.90	0.88	0.86	0.85

The three image block sorting methods are 1) original order, i.e., no sorting, 2) randomly sorting, and (3) sorting by the SSIM sensitivity score. All three tests show that block-wise perturbation pruning with blocks sorted by the SSIM sensitivity score gives the best performance

4.3.4 Effect of block size in perturbation pruning

In block-wise perturbation pruning, increasing the block size will speed up the process and likely lower the image quality. Table 11 shows the average SSIM and the number of model inferences when the block size increases from 2 to 16 for the test of Table 1 in Section 4.2.1. The table shows

Table 10 Attack performance with different values of λ in (9)

ϵ	0.03	0.05	0.07	0.09
$\lambda = 0.001$	0.90/0.74	0.82/0.89	0.76/0.95	0.72/0.97
$\lambda = 0.01$	0.90/0.73	0.83/0.88	0.77/0.94	0.73/0.96
$\lambda = 0.05$	0.91/0.73	0.83/0.87	0.77/0.94	0.73/0.96
$\lambda = 0.1$	0.91/0.73	0.83/0.87	0.77/0.94	0.74/0.96
$\lambda = 0.5$	0.91/0.73	0.84/0.88	0.78/0.95	0.74/0.96
$\lambda = 1.0$	0.93/0.70	0.88/0.86	0.83/0.93	0.80/0.95

The block-wise noise pruning is not used here. The two numbers in the table are the average SSIM score and attack success rate

Table 11 Effect of block size in block-wise perturbation pruning for the test of Table 1 in Section 4.2.1

ϵ	0.03	0.05	0.07	0.09
Block size=2	0.95/10	0.94/12	0.94/13	0.94/13
Block size=4	0.95/9	0.94/11	0.93/11	0.93/11
Block size=8	0.95/7	0.94/9	0.93/9	0.93/9
Block size=16	0.95/6	0.93/7	0.92/7	0.92/7

The two numbers are the average SSIM and the number of model inferences

that the average SSIM is highest when the block size is 2, which is reasonable as we search for the smaller blocks to turn off the perturbation. However, it is the slowest among the four. For block sizes of 4 and 8, the average SSIM is very close, which means our method can be even faster without significant loss of the image qualities.

4.3.5 Perceptual quality comparison

Figure 6 compares the image quality of several adversarial examples generated by the base attack method and our method. For each pair, the left one is generated by the base attack method, and the right one is ours. The three rows correspond to $\epsilon = 0.05, 0.06, 0.07$, respectively. For each adversarial image, we also show the SSIM score with respect to the clean image. We can see that our method significantly improves the visual quality of the attacked images. In Fig. 7, we show several pairs to compare the quality of the adversarial images generated by Perc_CW, $\text{PGD}_{L0+\sigma}$ and our method. We also show the SSIM score of each adversarial image. These sample pairs are taken from attack tests with similar attack success rates. We can see that under the same attack success rate, the qualities of the attacked images by Perc_CW, $\text{PGD}_{L0+\sigma}$ are worse than ours.

5 Discussion and future work

In the previous section, we have compared our method to the state-of-the-art dense and sparse attack methods against different defense schemes. The experimental results have demonstrated that our method can achieve significantly better attack performance in image qualities between the clean image and the adversarial, the attack success rate, and the attack efficiency in terms of the number of model

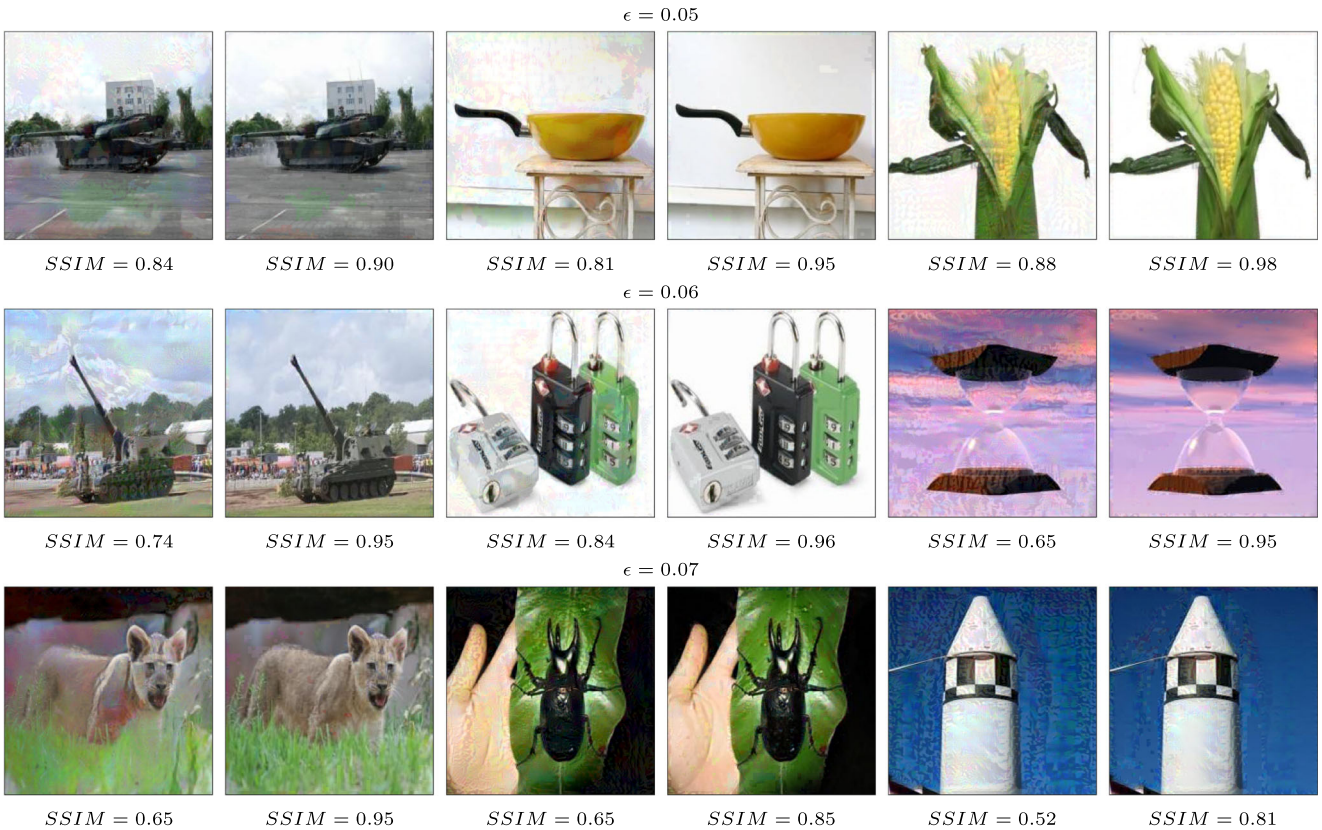


Fig. 6 Image quality comparison between the result of the base attack method and our method. For each pair, the left one is from the base attack method, and the right one is our result. The three image rows correspond to $\epsilon = 0.05, 0.06, 0.07$, respectively

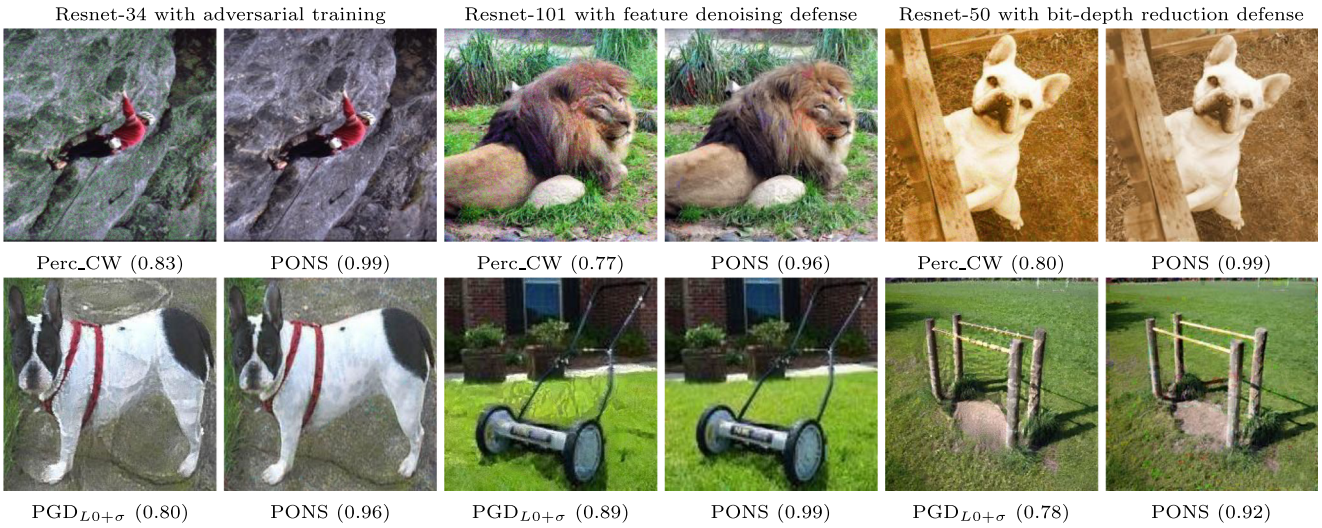


Fig. 7 Image quality comparison against Perc_CW and PGD$_{L0+\sigma}$. The first row is the comparison between ours and Perc_CW, and the second row is against PGD$_{L0+\sigma}$. The numbers in parentheses are the SSIM scores. Each pair is taken from tests with similar attack success rates

inferences. GreedyFool under l_0 norm constraint, i.e., the GF(500) in Table 8 at $\epsilon = 1.0$, has achieved a significant better average SSIM than ours at $\epsilon = 0.1$ in Table 1. However, the average MSE and attack complexity are substantially worse than ours.

The advantage of our method is that we use a base attack method that has strong attack capabilities so that we only need to optimize the image qualities without significant loss of attack success rate. Tables 1, 3, and 5 have demonstrated this. On the other side, the attack success rate of our method is largely determined by the base attack method as we only reshape the adversarial noise to improve the image quality. For this point, we can increase the perturbation budge to obtain a high attack success rate, which will inevitably introduce more noise to the images and require a more powerful perceptual attention network to produce high-quality perceptual attention masks. A better base attack method will certainly help improve the performance of our method.

In this work, the perceptual attention network is designed based on the Resnet-18 backbone to demonstrate its effectiveness. If we use a more advanced architecture for the perceptual attention network, we expect to see more improvement in the image quality and attack success rate.

6 Conclusion

In this work, we have observed that most existing adversarial attack methods are designed to maximize the attack success rate under the l_p norm constraint, which has not fully considered the perceptual sensitivity of the adversarial noise in different image regions. Motivated by this, we propose a novel two-stage attack method to maximize the image perceptual quality as well as the attack success rate. Specifically, we construct and learn a perceptual attention network to generate a perceptual attention mask to modulate the adversarial noise generated by a base attack method in the input image, aiming to maximize the visual quality while achieving the same attack success rate. To further improve the image perceptual quality, we propose a fast binary search algorithm to perform an iterative pruning of the adversarial noise based on the perceptual sensitivity map. We have conducted comprehensive evaluations and demonstrated that our method could significantly improve the image visual quality over the base attack method without sacrificing the attack success rate. When compared with the state-of-the-art adversarial attack methods, our method can achieve better attack performance in terms of image quality, attack success rate, and attack efficiency.

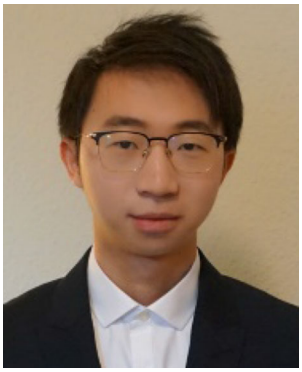
Acknowledgements This work is supported by the National Natural Science Foundation of China under grants of 61971290, 61771322, and 61871186, and partially by the Fundamental Research Foundation of Shenzhen under Grant of JCYJ20190808160815125.

References

1. Athalye A, Carlini N, Wagner D (2018) Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. CoRR arXiv:1802.00420
2. Athalye A, Engstrom L, Ilyas A, Kwok K (2017) Synthesizing robust adversarial examples. CoRR arXiv:1707.07397
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. Croce F, Hein M (2019) Sparse and imperceivable adversarial attacks. In: 2019 IEEE/CVF international conference on computer vision, ICCV 2019, Seoul, Korea (South), 27 October - 2 November, 2019. IEEE, pp 4723–4731 https://doi.org/10.1109/ICCV.2019.00482
5. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: Computer vision and pattern recognition, 2009. CVPR 2009. IEEE conference on. IEEE, pp 248–255
6. Dhillon GS, Azizzadenesheli K, Lipton ZC, Bernstein J, Kossaifi J, Khanna A, Anandkumar A (2018) Stochastic activation pruning for robust adversarial defense. CoRR arXiv:1803.01442
7. Dong X, Chen D, Bao J, Qin C, Yuan L, Zhang W, Yu N, Chen D (2020) Greedyfool: Distortion-aware sparse adversarial attack. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H (eds) Advances in neural information processing systems 33: annual conference on neural information processing systems 2020, NeurIPS 2020, 6-12 December, 2020, virtual
8. Dong Y, Liao F, Pang T, Su H, Zhu J, Hu X, Li J (2018) Boosting adversarial attacks with momentum. In: 2018 IEEE/CVF Conf Comput Vis Pattern Recognit:9185–9193
9. Goodfellow IJ, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples. In: Bengio Y, LeCun Y (eds) 3rd International conference on learning representations, ICLR 2015, San Diego, CA, USA, 7–9 May, 2015, conference track proceedings. arXiv:1412.6572
10. Guo C, Rana M, Cissé M, Maaten LVD (2018) Countering adversarial images using input transformations. arXiv:1711.00117
11. 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, pp 770–778
12. He Z, Wang W, Dong J, Tan T (2021) Transferable sparse adversarial attack. CoRR arXiv:2105.14727
13. Jordan M, Manoj N, Goel S, Dimakis AG (2019) Quantifying perceptual distortion of adversarial examples. CoRR arXiv:1902.08265
14. Kurakin A, Goodfellow IJ, Bengio S (2016) Adversarial machine learning at scale. CoRR arXiv:1611.01236
15. Kurakin A, Goodfellow IJ, Bengio S (2017) Adversarial examples in the physical world. In: 5th International conference on learning representations, ICLR 2017, Toulon, France, 24-26 April, 2017, workshop track proceedings. Openreview.net
16. Luo MR, Cui G, Rigg B (2001) The development of the cie 2000 color difference formula: Ciede2000. Color Res Appl 26(5):340–350
17. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A (2018) Towards deep learning models resistant to adversarial attacks. In: 6th International conference on learning representations, ICLR 2018, Vancouver, BC, Canada, 30 April - 3 May, 2018, Conference track proceedings. OpenReview.net
18. Modas A, Moosavi-Dezfooli S, Frossard P (2019) Sparsefool: a few pixels make a big difference. In: IEEE conference on computer vision and pattern recognition, CVPR 2019, long beach, CA, USA, 16-20 June, 2019, pp 9087–9096. Computer Vision Foundation/IEEE
19. Moosavi-Dezfooli SM, Fawzi A, Frossard P (2016) Deepfool: a simple and accurate method to fool deep neural networks. In: Computer vision & pattern recognition
20. Nagalla S, Inampudi MRB (2014) Perceptual weights based on local energy for image quality assessment. Int J Image Process, vol 8
21. Papernot N, McDaniel PD, Jha S, Fredrikson M, Celik ZB, Swami A (2016) The limitations of deep learning in adversarial settings. In: IEEE european symposium on security and privacy, EuroS&P 2016, Saarbrücken, Germany, 21-24 March, 2016. IEEE, pp 372–387. https://doi.org/10.1109/EuroSP.2016.36
22. Papernot N, McDaniel PD, Jha S, Fredrikson M, Celik ZB, Swami A (2016) The limitations of deep learning in adversarial settings. In: IEEE european symposium on security and privacy, EuroS&P 2016, Saarbrücken, Germany, 21-24 March, 2016. IEEE , pp 372–387. https://doi.org/10.1109/EuroSP.2016.36
23. Rozsa A, Rudd EM, Boult TE (2016) Adversarial diversity and hard positive generation. In: IEEE, pp 410–417
24. Su J, Vargas DV, Kouichi S (2017) One pixel attack for fooling deep neural networks. IEEE Trans Evol Comput
25. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow IJ, Fergus R (2014) Intriguing properties of neural networks. In: Bengio Y, LeCun Y (eds) 2nd International conference on learning representations, ICLR 2014, Banff, AB, Canada, 14-16 April, 2014, conference track proceedings. arXiv:1312.6199
26. Thomas S, Tabrizi N (2018) Adversarial machine learning: a literature review. In: International conference on machine learning & data mining in pattern recognition
27. Tramèr F, Kurakin A, Papernot N, Goodfellow I, Boneh D, Mcdaniel P (2017) Ensemble adversarial training: attacks and defenses. CoRR arXiv:1705.07204
28. Wang Z (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process
29. Xie C, Wu Y, van der Maaten L, Yuille AL, He K (2018) Feature denoising for improving adversarial robustness. CoRR arXiv:1812.03411
30. Xie C, Zhang Z, Zhou Y, Bai S, Wang J, Ren Z, Yuille A (2019) Improving transferability of adversarial examples with input diversity. In: Computer vision and pattern recognition. IEEE
31. Yeo C, Tan HL, Tan YH (2013) On rate distortion optimization using ssim. IEEE Trans Circuits Syst Video Technol 23(7):1170–1181. https://doi.org/10.1109/TCSVT.2013.2240918
32. Zhao Z, Liu Z, Larson M (2020) Towards large yet imperceptible adversarial image perturbations with perceptual color distance. In: 2020 IEEE/CVF Conference on computer vision and pattern recognition (CVPR)

Zhiquan He is currently working as assistant professor in College of Information Engineering, Shenzhen University, China. He received his M.S. degree from Institute of Electronics, Chinese Academy of Sciences in 2001, and the PhD degree from the department of Computer Science, University of Missouri-Columbia in 2014. His research area is in the areas of image processing, computer vision and machine learning.

Xujia Lan is currently pursuing the M.Eng. degree in Electronic Information in Shenzhen University, Shenzhen, China. His research interests are image processing and Artificial intelligence.



Jianhe Yuan was pursuing the PhD degree in the Department of Electrical Engineering and Computer Science, University of Missouri-Columbia, Missouri, USA. He has now joined NVIDIA to continue his research in deep learning and computer vision.

Wenming Cao received the M.S. degree from the System Science Institute, China Science Academy, Beijing, China, in 1991, and the Ph.D. degree from the School of Automation, Southeast University, Nanjing, China, in 2003. From 2005 to 2007, he was a postdoctoral Researcher with the Institute of Semiconductors, Chinese Academy of Sciences, Beijing, China. He is currently a Professor with Shenzhen University, Shenzhen, China. He has authored or coauthored more than 80 publications in top-tier conferences and journals. His research interests include pattern recognition, image processing, and visual tracking.

Affiliations

Zhiquan He[1,2,3] · Xujia Lan[4] · Jianhe Yuan[5] · Wenming Cao[4]

Xujia Lan
2070436049@email.szu.edu.cn

Jianhe Yuan
jianhe@mizzou.edu

Wenming Cao
wmcao@szu.edu.cn

1 College of Electronics and Information Engineering, Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen, People's Republic of China

2 Guangdong Multimedia Information Service Engineering Technology Research Center, Shenzhen, People's Republic of China

3 Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen, People's Republic of China

4 College of Electronics and Information Engineering, Shenzhen University, Shenzhen, People's Republic of China

5 Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO, 65211, USA