Patch-wise Attack for Fooling Deep Neural Network

Lianli Gao[1[0000−0001−5883−599X]], Qilong Zhang[1[0000−0002−2134−2160]], Jignkuan Song[1[0000−0002−2549−8322]], Xianglong Liu[2[0000−0002−7618−3275]], and Shen Heng Tao[1[0000−0002−2999−2088]]

[1] University of Electronic Science and Technology of China, China
[2] Beihang University, China
qilong.zhang@std.uestc.edu.cn

Abstract. By adding human-imperceptible noise to clean images, the resultant adversarial examples can fool other unknown models. Features of a pixel extracted by deep neural networks (DNNs) are influenced by its surrounding regions, and different DNNs generally focus on different discriminative regions in recognition. Motivated by this, we propose a patch-wise iterative algorithm – a black-box attack towards mainstream normally trained and defense models, which differs from the existing attack methods manipulating pixel-wise noise. In this way, without sacrificing the performance of white-box attack, our adversarial examples can have strong transferability. Specifically, we introduce an amplification factor to the step size in each iteration, and one pixel's overall gradient overflowing the ϵ-constraint is properly assigned to its surrounding regions by a project kernel. Our method can be generally integrated to any gradient-based attack methods. Compared with the current state-of-the-art attacks, we significantly improve the success rate by 9.2% for defense models and 3.7% for normally trained models on average. Our code is available at `https://github.com/qilong-zhang/Patch-wise-iterative-attack`

Keywords: Adversarial examples, Patch-wise, Black-box attacks, Transferability

1 Introduction

In recent years, Deep neural networks (DNNs) [9,10,27,28] have made great achievements. However, the adversarial examples [29] which are added with human-imperceptible noise can easily fool the state-of-the-art DNNs to give unreasonable predictions. This raises security concerns about those machine learning algorithms. In order to understand DNNs better and improve its robustness to avoid future risks [6], it is necessary to investigate the defense models, and meanwhile the generation of adversarial examples.

Various attack methods have been proposed in these years. One of the most popular branches is gradient-based algorithms. For this branch, existing methods

can be generally classified as single-step attacks and iterative attacks. In general, iterative attacks perform better than single-step attacks in the white-box setting. But in the real world, attackers usually can not get any information about the target model, which is called the black-box setting. In this setting, single-step attacks always have a higher transferability than iterative attacks at the cost of poor performance of substitute models. For example, adversarial examples that are generated by the translation-invariant attack method [3] are less sensitive to the discriminative regions [36] of the white-box models, but it demonstrates a high success rate of attack towards black-box defense models. To sum up, the essential difference between the two sub-branches is the number of iterations. Iterative attacks require multiple iterations to obtain the final perturbation noise, and hence there is a risk of getting stuck in the local optimum during the iterations, reducing the transferability. While single-step attack methods only update once, which is easy to underfit but really improve the generalizability.

Moreover, with the development of attack methods, several adversarial examples have been applied to the physical world [17,6,12,26,31]. This has raised public concerns about AI security. Consequently, a lot of defense methods are proposed to tackle this problem. Guo *et al.* [8] use bit-depth reduction, JPEG compression [4], total variance minimization [23], and image quilting [5] to preprocess inputs before they are feed to DNNs. Tramr *et al.* [32] use ensemble adversarial training to improve the robustness of models. Furthermore, Xie *et al.* [34] add feature denoising module into adversarial training, and the resultant defense models demonstrate greater robustness in both white-box and black-box attack settings. To generate more effective adversarial examples, it is necessary to study the properties of intrinsic classification logic of the DNNs. Here we use class activation mapping [36] to analyze it. Zhou *et al.* [36] observed that the discriminative regions always vary across predicted labels. Recent research [3] also showed that different models focus on different discriminative regions in recognition, and the defense models generally focus on larger discriminative regions than the normally trained models. The discriminative regions are often clustered together, as shown in Fig.1. Therefore, only adding pixel-wise noise may hinder the transferability of adversarial examples across different DNNs. Based on these observations, we argue that in addition to reducing the responsiveness of the ground-truth regions while activating the regions of any other categories, crafting perturbation with the characteristic of aggregation in the above-discussed regions is also important. To that end, we study the advantages and disadvantages of single-step and iterative attacks [3,7,14], and we argue that linear nature of DNNs [7] does exist to some extent. Therefore we amplify step size with a fixed factor to analyze the effect of step size on transferability. Besides, we rethink the weakness of direct clipping operation which discards partial gradient information. To alleviate this problem, we reuse the cut noise and apply a heuristic project strategy to reduce the side effects of direct clipping as well as generating patch-wise noise. Finally, combined with the fast sign gradient method, we propose the **Patch-wise Iterative Fast Gradient Sign Method (PI-FGSM)** to generate strongly transferable adversarial examples.

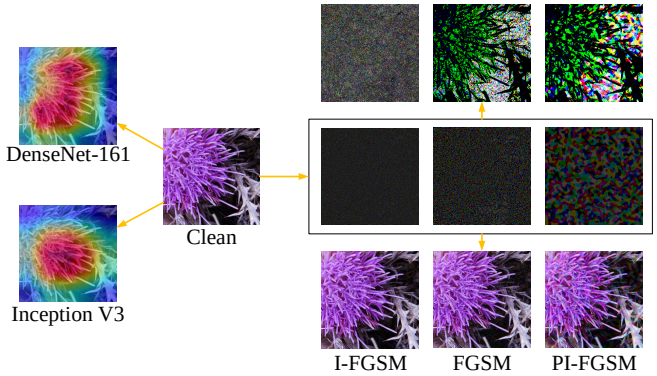

Fig. 1. We show the adversarial examples generated by FGSM [7], I-FGSM [14] and our method PI-FGSM for Inception V3 model [28] respectively. The maximum perturbation ϵ is limited to 16. On the left side of the figure, we use Inception V3 [28] and DenseNet-161 [10] to show the Gradient-weighted Class Activation Mapping (Grad-CAM) [25] of the ground-truth label, and on the right side of the figure, we divide it into three parts. **Top row**: the adversarial noise patch map (we define it in Sec. 3.2.1). **Middle row**: the adversarial noise. **Bottom row**: the adversarial examples. Our PI-FGSM can generate adversarial noise which has the same clustering property as the activation map and also well covers the different discriminative regions.

The visualization results in Fig. 1 also demonstrate our approach. Compared with other methods, the noise generated by our PI-FGSM has obvious aggregation characteristics and can better cover varied discriminative regions of different DNNs. Our major contributions can be summarized as: 1) We propose a novel patch-wise attack idea named PI-FGSM. Compared with existing methods manipulating pixel-wise noise, our approach can have the advantage of both single-step and iterative attacks, i.e., improving the transferability without sacrificing the performance of the substitute model. 2) Technically, based on the mature gradient-based attack pipeline, we adopt an amplification factor and project kernel to generate more transferable adversarial examples by patch-wise noise. Our method can be generally integrated to any iteration-based attack methods; and 3) Extensive experiments on ImageNet show that our method significantly outperforms the state-of-the-art methods, and improves the success rate by 9.2% for defense models and 3.7% for normally trained models on average in the black-box setting.

2 Related work

In this section, we briefly analyze the exiting adversarial attack methods, from the perspectives of classification of adversarial examples, attack setting, and ensemble strategy.

2.1 Adversarial Examples

Adversarial examples are first discovered by Szegedy *et al.*[29], which only added subtle perturbation to the original image but can mislead the DNNs to make an unreasonable prediction with unbelievably high confidence. To make matters worse, adversarial examples also exist in physical world [6,12,13], which raises security concerns about DNNs. Due to the vulnerability of DNNs, a large number of attack methods have been proposed and applied to various fields of deep learning in recent years, e.g., object detection and semantic segmentation [33], embodied agents [16], and speech recognition [1]. To make our paper more focused, we only analyze adversarial examples in the image classification task.

2.2 Attack Settings

In this section, we describe three common attack settings for the adversarial examples. One is the white-box setting. In this case, the adversary can get the full knowledge of the targeted models, thus obtaining accurate gradient information to update adversarial examples. The second is the semi-black-box setting where the output of the targeted model is available but model parameters are still unknown. E.g., Papernot *et al.* [21] train a local model with many queries to substitute for the target model. Ilyas *et al.* [11] propose the variant of NES [24] to generate adversarial examples with limited queries. The rest is the black-box setting, where the adversary generally can not access the target model and adversarial examples are usually generated for substitute models without exception. That is why transferability plays a key role in this setting. Recently, the black-box attack is a hot topic and many excellent works have been proposed. Xie *et al.* [35] apply random transformations to the input images at each iteration to improve transferability. Dong *et al.* [2] propose a momentum-based iterative algorithm to boost adversarial attacks. Besides, the adversarial examples which are crafted by their translation-invariant attack method [3] can evade the defenses with effect. However, the above works can not generate powerful patch-wise noise because they generally take valid gradient information into account. In this paper, our goal is crafting efficient patch-wise noise to improve the transferability of adversarial examples in the black-box setting.

2.3 Ensemble Strategy

There are two well-known but totally different strategies for this topic. One strategy uses an ensemble of legitimate examples to update only one universal adversarial perturbation. Moosavi-Dezfooli *et al.* [20] propose an iterative attack method to generate such perturbations which cause almost all images sampled from the data distribution to be misclassified. Another strategy uses an ensemble of models to get a better estimate of the gradient information. Liu *et al.* [18] propose novel ensemble-based approaches that attacking multiple models to generate adversarial examples. In this way, the adversarial examples are less likely to get stuck in the local optimum of any specific model, thus improving transferability.

3 Methodology

In this section, we describe our algorithm in detail. Let x^{clean} denote a clean example without any perturbation and y denote the corresponding ground-truth label. We use $f(x)$ to denote the prediction label of DNNs, and x^{noise} to denote the human-imperceptible perturbation. The adversarial example $x^{adv} = x^{clean} + x^{noise}$ is visually indistinguishable from x^{clean} but misleads the classifier to give a high confidence of a wrong label. In this paper, we focus on untargeted black-box attack, i.e., $f(x^{adv}) \neq y$. And the targeted version can be simply derived. To measure the perceptibility of adversarial perturbations, we follow previous works [2,3,35] and use l_∞-norm here. Namely, we set the max adversarial perturbation ϵ, i.e., we should keep $||x^{clean} - x^{adv}||_\infty \leq \epsilon$. To generate our adversarial examples, we should maximize the cross-entropy loss $J(x^{adv}, y)$ of the substitute models. Our goal is to solve the following constrained optimization problem:

$$\arg\max_{x^{adv}} J(x^{adv}, y), \qquad s.t.\ ||x^{clean} - x^{adv}||_\infty \leq \epsilon. \tag{1}$$

Due to the black-box setting, the adversary does not allow to analytically compute the target models' gradient $\nabla J(x, y)$. In the majority of cases, they use the information of substitute models (i.e., official pre-trained models) to generate adversarial examples. Therefore, it is very important to improve the transferability of adversarial examples so that they still fool the black-box models successfully.

3.1 Development of Gradient-based Attack Methods

In this section, we give a brief introduction of some excellent works which are based on the gradient sign method.

- *Fast Gradient Sign Method (FGSM)*: Goodfellow *et al.* [7] argue that the vulnerability of DNN is their linear nature. Consequently they update the adversarial example by:

$$x^{adv} = x^{clean} + \epsilon \cdot sign(\nabla_x J(x^{clean}, y)). \tag{2}$$

  where $sign(\cdot)$ indicates the sign operation.
- *Iterative Fast Gradient Sign Method (I-FGSM)*: Kurakin *et al.* [14] use a small step size α to iteratively apply the gradient sign method multiple times. This method can be written as:

$$x^{adv}_{t+1} = Clip_{x^{clean},\epsilon}\{x^{adv}_t + \alpha \cdot sign(\nabla_x J(x^{adv}_t, y))\}. \tag{3}$$

  where $Clip_{x^{clean},\epsilon}$ denotes element-wise clipping, aiming to restrict x^{adv} within the l_∞-bound of x^{clean}.
- *Momentum Iterative Fast Gradient Sign Method (MI-FGSM)*: Dong *et al.* [2] use momentum term to stabilize update directions. It can be expressed as:

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(x^{adv}_t, y)}{||\nabla_x J(x^{adv}_t, y)||_1}, \quad x^{adv}_{t+1} = x^{adv}_t + \alpha \cdot sign(g_{t+1}). \tag{4}$$

  where g_t is cumulative gradient, and μ is the decay factor.

– *Diverse Input Iterative Fast Gradient Sign Method (DI^2-FGSM)*: Xie *et al.* [35] apply diverse input patterns to improve the transferability of adversarial examples. With the replacement of Eq. (3) by:

$$x_{t+1}^{adv} = Clip_{x^{clean},\epsilon}\{x_t^{adv} + \alpha \cdot sign(\nabla_x J(D(x_t^{adv}), y))\}. \tag{5}$$

where $D(x)$ is random transformations to the input x. For simplicity, we use DI-FGSM later.

– *Translation-Invariant Attack Method*: Dong *et al.* [3] convolve the gradient with the pre-defined kernel W to generate adversarial examples which are less sensitive to the discriminative regions of the substitute model. For TI-FGSM, it is only updated in one step:

$$x^{adv} = x^{clean} + \epsilon \cdot sign(W * \nabla_x J(x_t^{adv}, y)). \tag{6}$$

and TI-BIM is its iterative version.

3.2 Patch-wise Iterative Fast Gradient Sign Method

In this section, we elaborate our method in details. We first introduce our motivations in Sec. 3.2.1 and Sec. 3.2.2. In Sec. 3.2.3, we will describe our solution.

3.2.1 Patch Map. Natural images are generally made up of smooth patches [19] and the discriminative regions are usually focused on several patches of them. However, as demonstrated in Fig. 1, different DNNs generally focus on different discriminative regions, but these regions usually contain clustered pixels instead of scattered ones. Besides, Li et al. [15] have demonstrated that regionally homogeneous perturbations are strong in attacking defense models, which is especially helpful to learn transferable adversarial examples in the black-box setting. For this reason, we believe that noises with the characteristic of aggregation in these regions are more likely to attack successfully because they perturb more significant information. To better view the adversarial noise x^{noise}, we take the absolute value of x^{noise} to define its patch map x^{map}[3], which is done by:

$$x^{map} = |x^{noise}| \times \frac{256}{\epsilon}. \tag{7}$$

As shown in Fig. 1, compared with the patch map of I-FGSM, and FGSM, our PI-FGSM can generate the noise with more obvious aggregation characteristics.

3.2.2 Box Constraint. To the best of our knowledge, almost all iterative gradient-based methods apply *projected gradient descent* to ensure the perturbation within the box. Although this method can improve the generalization of

[3] Pixel values of a valid image are in [0, 255]. If the values are more than 255, they will be modified into 0 for "uint8" type, to give better contrast.

adversarial examples to some extent [20], it also has certain limitations. Let us take the dot product $D(\cdot)$ as an example:

$$D(x_t^{adv}) = wx_t^{adv} + b, \qquad D^{'}(x_t^{adv}) = w. \tag{8}$$

where w denotes a weight vector and b denotes the bias. Then we add a noise αw to update x_t^{adv}:

$$D(Clip_{x^{clean},\epsilon}\{x_t^{adv} + \alpha w\}) \approx D(x_t^{adv}) + \alpha_2 w^2. \tag{9}$$

If $x_t^{adv} + \alpha w$ excess the ϵ-ball of original image x^{clean}, the result is Eq.(9). Obviously, $\alpha_2 < \alpha$ due to element-wise clipping operation. If we adopt this strategy directly, we will waste some of the gradient information and change the input unexpectedly.

3.2.3 Our Method. From the above analysis, we observe that adding noise in a patch-wise style will have better transferability than the pixel-wise style. Also, the element-wise clipping operation of existing gradient-based attack methods will lose part of the gradient information and lead to unexpected changes. Therefore, we propose our method, which follows the mature gradient-based attack pipeline and tackles the above issues simultaneously.

To the best of our knowledge, many recent iterative attack methods [2,3,35] set step size $\alpha = \epsilon/T$, where T is the total number of iterations. In such a setting, we do not need the element-wise clipping operation, and the adversarial examples can finally reach the ϵ bound of x^{clean}. This seems like a good way to get around the above problem of direct clipping, but we notice that single-step attacks often outperform iterative attacks in the black-box setting. To study the transferability with respect to the step size setting, we make a tradeoff between single large step and iterative small step by setting it to $\epsilon/T \times \beta$, where β is an amplification factor.

The results in Fig. 2 show that iterative approaches with a large amplification factor will help to avoid getting stuck in poor local optimum, thus demonstrating a stronger attack towards black-box models. One possible reason is that attacks with an amplification factor increase each element's value of the resultant perturbation, thus providing a higher probability of misclassification due to the linear assumption of Goodfellow *et al.* [7]. However, simply increasing the step size does not get around the disadvantages of direct clipping operation, because the excess noise would be eliminated.

Therefore, we propose a novel heuristic project strategy to solve this problem. Our inspiration comes from Rosen Project Gradient Method [22]: by projecting the gradient direction when the iteration point is on the edge of the feasible region, the method ensures the iteration point remains within the feasible region after updating. However, performing this method is a little complex and needs additional computational cost. Hence we take a heuristic strategy to apply this idea: just projecting the excess noise into the surrounding field. We argue that the part of the noise vector which is more easy to break ϵ-ball limitation has

Algorithm 1: PI-FGSM

Input : The cross-entropy loss function J of our substitute models; iterations T; L_∞ constraint ϵ; project kernel W_p; amplification factor $\beta(\geq 1)$; project factor γ; a clean image x^{clean} (Normalized to [-1,1]) and the corresponding groud-truth label y;

Output: The adversarial example x^{adv};

1 Initialize cumulative amplification noise a_0 and cut noise C to 0;
2 $x_0^{adv} = x^{clean}$;
3 **for** $t \leftarrow 0$ **to** T **do**
4 Calculate the gradient $\nabla_x J(x_t^{adv}, y)$;
5 $a_{t+1} = a_t + \beta \cdot \frac{\epsilon}{T} \cdot sign(\nabla_x J(x_t^{adv}, y))$; `// Update` a_{t+1}
6 **if** $||a_{t+1}||_\infty \geq \epsilon$ **then**
7 $C = clip(|a_{t+1}| - \epsilon, 0, \infty) \cdot sign(a_{t+1})$;
8 $a_{t+1} = a_{t+1} + \gamma \cdot sign(W_p * C)$;
9 **else**
10 $C = 0$;
11 **end**
12 $x_{t+1}^{adv} = Clip_{x^{clean},\epsilon}\{x_t^{adv} + \beta \cdot \frac{\epsilon}{T} \cdot sign(\nabla_x J(x_t^{adv}, y)) + \gamma \cdot sign(W_p * C)\}$;
13 $x_{t+1}^{adv} = clip(x_{t+1}^{adv}, -1, 1)$; `// Finally clip` x_{t+1}^{adv} `into [-1,1]`
14 **end**
15 Return $x^{adv} = x_T^{adv}$;

a higher probability of being in the highlighted area of discriminative regions. Our strategy can simply reuse the noise to increase the degree of aggregation in these regions without additional huge computational cost.

The integration of patch-wise iterative algorithm and fast gradient sign method (PI-FGSM) is summarized in Algorithm 1. Firstly, in line 5, we need to get the cumulative amplification noise a_t. After amplification operation, if L_∞-norm of a_t exceeds the threshold ϵ, we update the cut noise C by:

$$C = clip(|a_{t+1}| - \epsilon, 0, \infty) \cdot sign(a_{t+1}). \tag{10}$$

where $|\cdot|$ denotes the absolute operation. Finally, unlike other methods, we add an additional project term before restricting the L_∞-norm of the perturbations. Note that we do not abandon the clipping operation. Instead, we just reuse the cut noise to alleviate the disadvantages of direct clipping, thus increasing the aggregation degree of noise patches. More specifically, we update the adversarial examples by:

$$x_{t+1}^{adv} = Clip_{x^{clean},\epsilon}\{x_t^{adv} + \beta \cdot \frac{\epsilon}{T} \cdot sign(\nabla_x J(x_t^{adv}, y)) + \gamma \cdot sign(W_p * C)\}. \tag{11}$$

where W_p is a special uniform project kernel of size $k_w \times k_w$, and $sign(W_p * C)$ is cut noise's "feasible direction". In this paper, we simply define W_p as:

$$W_p[i,j] = \begin{cases} 0, & i = \lfloor k_w/2 \rfloor + 1, j = \lfloor k_w/2 \rfloor + 1. \\ 1/(k_w^2 - 1), & else. \end{cases} \tag{12}$$

We also test other types of kernels actually, e.g., Gaussian kernel. However, experiment results show that there are no significant difference (only $\sim$1%). Besides, the uniform kernel does not need extra parameters. Therefore we choose it finally.

4 Experiment

4.1 Setup

Following the previous works [2,3], we also do our experiment on ImageNet-compatible dataset[4], which contains 1000 images and is used for NIPS 2017 adversarial competition. We choose eleven models to do these experiments. For normally trained models, we consider Inception V3 (Inc-v3) [28], Inception V4 (Inc-v4), Inception-ResNet V2 (IncRes-v2) [27] ResNet152 V2 (Res-152) [9] and DenseNet 161 (Dense-161) [10]. For defense models, we include three ensemble adversarial trained models: Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$ [32], and three more robust models from [35]: ResNet152 Baseline (Res152$_B$), ResNet152 Denoise (Res152$_D$), ResNeXt101 DenoiseAll (ResNeXt$_{DA}$) [34]. For the sake of simplicity, we use NT to denote normally trained models, EAT to denote ensemble adversarial trained models and FD to denote feature denoising defense models (include Res152$_B$). It should be noted that the success rate of FD needs to minus the ratio of clean images that are predicted incorrectly by FD. In addition, our proposed approach PI-FGSM can be easily combined with other attack methods, e.g., MI-FGSM [2], DI-FGSM [35] and TI-BIM [3]. To make the abbreviation unambiguous, we use the first character to denote the corresponding method. For instance, DPI-FGSM means the integration of DI-FGSM (Input Diversity) with PI-FGSM.

In our experiment, we set equal weight when attacking an ensemble of models. The maximum perturbation ϵ is set to 16. The iteration T is set to 10 for all iterative methods. For iterative methods without our amplification factor, the step size is $\epsilon/T = 1.6$. For MI-FGSM, we set the decay factor $\mu = 1.0$; for TI-FGSM and TI-BIM, we set the kernel size $k = 15$; and for DI-FGSM, we set the transformation probability $p = 0.7$.

4.2 Amplification factor

In this section, we calculate the success rate of different amplification factors, which are varied from 1 to 10. The results are shown in Fig. 2. In general, a larger amplification factor does improve performance. Moreover, we observe that when the amplification factor is large enough, further increasing it will not bring significant improvement and may even reduce the transferability of some attacks. For example, it is better not to use large amplification factor for DI-FGSM, because this method applies random transformations to the input

[4] https://github.com/tensorflow/cleverhans/tree/master/examples/nips17_adversarial_competition/dataset

images. If we set the amplification factor too large, the adversarial examples may deviate from the global optimum. In addition, a larger amplification factor has no obvious effect for MI-FGSM, which also proves that the momentum term can stabilize the update directions and avoid getting stuck in a poor local optimum. For our method, as the amplification factor increases, the success rate will increase rapidly and outperform other approaches soon. If we set $\beta = 1$, PI-FGSM degrades to I-FGSM because the cut noise has no effect anymore in this setting. By comparing the growth curves of I-FGSM and PI-FGSM, we observe that our heuristic project strategy can improve the transferability by a large margin, which fully validates the effectiveness of our method. We also ex-

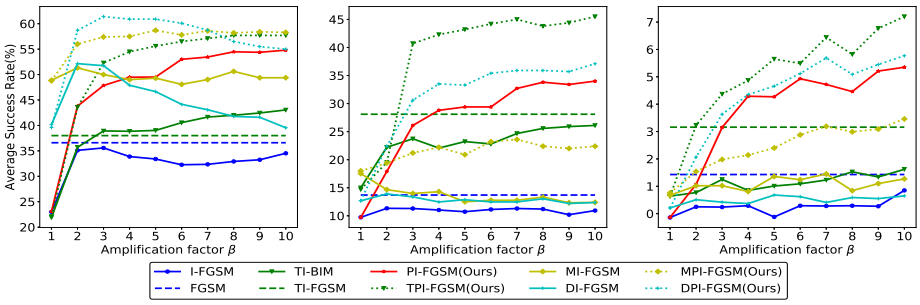

Fig. 2. The average success rate(%) of non-targeted attacks in different amplification factor β setting. The adversarial examples are crafted for Inc-v3 by FGSM, I-FGSM, MI-FGSM, TI-BIM, TI-FGSM, DI-FGSM, PI-FGSM and their combined versions respectively. **Left Column**: The result of *NT*, including Inc-v4, Res-152, IncRes-v2 and Dense-161 but except Inc-v3; **Middle Column**: The result of *EAT*, including Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$; **Right Column**: The result of *FD*, including ResNeXt$_{DA}$, Res152$_B$ and Res152$_D$.

amine the influence of amplification factor on the combined methods and show the results in Fig. 2. It can be observed that the optimal amplification factor of combined methods is usually between the best β of the two methods. If one of these methods' performance is negatively correlated with the amplification factor, then the combination with our method may not perform well (i.e., MPI-FGSM vs. EAT). In our experiments, the project factor γ is set to $\epsilon/T \cdot \beta$ in most cases, therefore we only focus on the amplification factor in this ablation study, specific parameter settings will be given in the later experiments.

4.3 Project Kernel Size

In fact, the function of the project kernel is to generate patch-wise noise. As shown in Fig. 1, the noise patch map of PI-FGSM is larger and regular than others. The results in Fig. 2 also demonstrate the effectiveness of our proposed

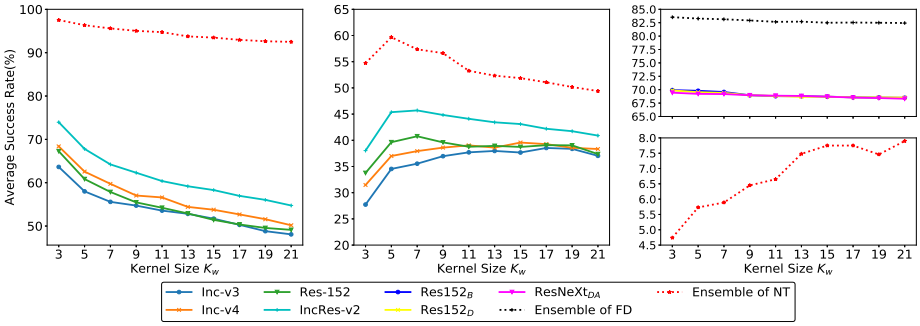

Fig. 3. The average success rate(%) of non-targeted attacks in different kernel size settings. The adversarial examples are crafted for different substitute models. **Left Column**: The results of *NT*, including Inc-v3, Inc-v4, Res-152, IncRes-v2 and Dense-161; **Middle Column**: The results of *EAT*, including Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$; **Right Column**: The results of *FD*, including ResNeXt$_{DA}$, Res152$_B$ and Res152$_D$.

method. However, The size of the kernel W_p play an important role for transferability. From Fig. 3, we find the optimal size of project kernel is different for NT, EAT and FD:

- When transferring to NT, 3×3 kernel is the best. Larger kernel size always decreases the transferability in the black-box setting.
- When transferring to EAT, 7×7 kernel can improve our transferability obviously. But if we keep increasing the size, the success rate grows slowly and even gets worse.
- When transferring to FD, if adversarial examples are generated for FD, 3×3 kernel is the best. And if we use NT to attack against FD, 21×21 is slightly better.

Considering the difference between the NT, EAT and FD, we use 3×3 kernel to attack against NT, 7×7 for EAT and 21×21 for FD if adversarial examples are crafted for NT. Also, we use 3×3 kernel to fool FD if adversarial examples are generated for FD.

4.4 Attacks vs. Normally Trained Models

In this section, we focus on the vulnerability of NT. we set $\beta = 10$ ($\gamma = 16$) for PI-FGSM and MPI-FGSM, and $\beta = 2.5, \gamma = 2$ for DMPI-FGSM and DPI-FGSM. We compare PI-FGSM with FGSM, I-FGSM, MI-FGSM, DI-FGSM to verify the effectiveness of our method. Moreover, we also test the performance of the combination of different methods, e.g., MPI-FGSM. It should be noted that we do not consider TI-BIM or TI-FGSM [3] here, because they focus more on attacking the defense models.

Table 1. The success rate(%) of non-targeted attacks against NT. The leftmost column models are substitute models ("*" indicates white-box attacks), the adversarial examples are crafted for them by FGSM, I-FGSM, MI-FGSM, DI-FGSM, PI-FGSM, and their combined versions respectively.

	Attacks	Inc-v3	Inc-v4	Res-152	IncRes-v2	Dense-161
Inc-v3	FGSM	80.9*	38.0	33.1	33.9	41.4
	I-FGSM	**100.0***	29.6	19.4	20.3	20.7
	MI-FGSM	**100.0***	54.1	43.5	50.9	45.8
	DI-FGSM	99.8*	54.2	32.1	43.6	30.4
	PI-FGSM(Ours)	**100.0***	**58.6**	**45.0**	**51.3**	**61.7**
	MPI-FGSM(Ours)	**100.0***	63.0	50.6	60.0	59.0
	DPI-FGSM(Ours)	**100.0***	73.1	51.2	67.4	55.9
	DMI-FGSM	99.9*	78.9	**63.9**	75.6	60.7
	DMPI-FGSM(Ours)	**100.0***	**81.8**	63.4	**77.1**	**63.7**
Inc-v4	FGSM	45.4	75.1*	35.1	35.8	45.4
	I-FGSM	43.3	**100.0***	25.5	25.3	24.7
	MI-FGSM	**71.2**	**100.0***	**52.4**	**59.0**	51.2
	DI-FGSM	66.6	**100.0***	39.8	50.4	33.2
	PI-FGSM(Ours)	70.7	**100.0***	50.9	54.9	**65.8**
	MPI-FGSM(Ours)	77.3	**100.0***	56.5	63.7	64.3
	DPI-FGSM(Ours)	84.3	**100.0***	57.6	70.6	61.8
	DMI-FGSM	89.0	**100.0***	**70.8**	80.2	67.7
	DMPI-FGSM(Ours)	**90.4**	**100.0***	70.6	**82.2**	**70.0**
Res-152	FGSM	41.4	36.4	82.3*	32.0	45.1
	I-FGSM	30.7	24.7	99.5*	16.9	23.7
	MI-FGSM	56.1	51.0	99.5*	47.9	50.2
	DI-FGSM	60.0	**56.5**	99.2*	49.3	43.1
	PI-FGSM(Ours)	**63.3**	54.4	**99.7***	**50.6**	**67.4**
	MPI-FGSM(Ours)	68.8	62.5	**99.7***	59.9	69.0
	DPI-FGSM(Ours)	81.0	77.2	99.6*	75.0	71.6
	DMI-FGSM	82.2	79.4	99.3*	74.8	72.0
	DMPI-FGSM(Ours)	**86.1**	**83.4**	99.5*	**82.0**	**75.2**
IncRes-v2	FGSM	45.9	39.2	35.7	68.3*	45.6
	I-FGSM	48.2	38.3	25.5	**100.0***	27.0
	MI-FGSM	**77.6**	68.4	57.0	**100.0***	57.1
	DI-FGSM	70.2	66.1	47.9	99.2*	42.0
	PI-FGSM(Ours)	76.1	**68.8**	**58.2**	99.9*	**69.8**
	MPI-FGSM(Ours)	80.2	75.5	63.8	**100.0***	70.3
	DPI-FGSM(Ours)	87.2	83.4	65.0	99.7*	69.2
	DMI-FGSM	86.8	85.6	76.5	99.1*	70.9
	DMPI-FGSM(Ours)	**92.2**	**89.0**	**77.1**	99.6*	**74.7**

The results are shown in Tab. 1. To sum up, compared with other attacks, our PI-FGSM can improve the success rate by **3.7%** on average, and when we attack against Dense-161[5], transferability can be increased by up to **17.2%**. This is because our perturbation patches have the property of aggregation, thus reducing the impact of resizing. In particular, if we integrate PI-FGSM into other attack methods such as DMI-FGSM, we can get a much better result. For instance, the adversarial examples generated for IncRes-v2 by DMPI-FGSM can fool Inc-v3 on **92.2%** images in the black-box setting which also demonstrates the vulnerability of NT.

[5] Input size need to be [224,224,3], therefore we need resize adversarial examples whose size is [299,299,3]

Table 2. The success rate(%) of non-targeted attacks against EAT. The top row models are substitute models, we use them to generate adversarial examples by FGSM, I-FGSM, DI-FGSM, MI-FGSM, TI-FGSM, PI-FGSM and their combined versions respectively.

	Inc-v3			Inc-v4			Res-152			IncRes-v2		
Attack	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
FGSM	16.8	15.8	8.3	16.6	17.2	9.1	21.4	19.4	11.4	18.6	17.5	11.2
I-FGSM	11.7	12.1	5.5	11.8	13.0	6.6	13.0	13.3	6.7	13.7	13.3	8.2
MI-FGSM	21.9	21.1	10.5	24.7	23.7	13.2	27.0	24.9	15.9	31.9	29.1	20.7
DI-FGSM	15.0	16.2	7.1	14.7	17.7	8.4	21.6	21.1	12.9	19.3	20.2	12.7
TI-FGSM	30.8	30.6	22.7	30.2	31.3	23.2	36.6	36.1	29.5	36.3	36.0	30.4
PI-FGSM(Ours)	**39.3**	**39.5**	**28.8**	**40.3**	**41.8**	**31.4**	**43.0**	**45.0**	**34.9**	**46.4**	**48.4**	**42.4**
TPI-FGSM(Ours)	49.1	50.2	36.5	49.6	51.7	38.5	51.5	51.4	43.3	59.2	60.3	56.5
DPI-FGSM(Ours)	40.9	41.6	31.2	43.4	46.0	33.8	47.9	48.5	41.4	48.8	50.4	45.8
DTMI-FGSM	50.7	50.3	39.5	**54.0**	54.2	**42.9**	59.5	58.4	53.1	65.4	63.8	**62.9**
DTPI-FGSM(Ours)	**51.5**	**53.1**	**40.0**	52.2	**54.7**	41.7	**65.3**	**64.5**	**56.3**	**70.0**	**67.7**	62.3

4.5 Attacks vs. Defense Models

Our approach is also effective for defense models. In this experiment, we use EAT [32] and FD [34] to examine the transferability and we do not integrate the momentum term into our proposed PI-FGSM because it may hinder the performance.

Here we study the single-model attacks firstly. In this case, we set $\beta = 10$ ($\gamma = 16$), and the results are shown in Tab. 2. Compared with TI-FGSM, the average success rate of our method is improved by about **9.0%**. In particular, if we use DTPI-FGSM to attack IncRes-v2, **70.0%** adversarial examples can fool Inc-v3$_{ens}$. Noted that the results of Tab. 2 are not our best parameter setting, because the best kernel size for a single-model attack is not the same. Here we just set the $k_w = 7$ to keep the experimental parameters consistent.

The transferability can be greatly improved when the adversarial examples are crafted for an ensemble of models at the same time [18]. It is because this strategy can prevent adversarial examples from falling into a local optimum of any specific model. In this case, we set $\beta = 5$ ($\gamma = 8$) and the result are shown in Tab. 3. Compared with MI-FGSM, our proposed PI-FGSM improves the performance by about **9.6%** on average. Furthermore, compared with DTMI-FGSM [3] which takes momentum into account, our DTPI-FGSM still outperform it.

Furthermore, when using Inc-v3 to attack against FD, we are surprised to find that sometimes I-FGSM even perturbs misclassified images into correctly classified ones (See Fig. 2). This may be due to the significant difference in decision boundaries between the NT and the FD. Since FD are very robust, and the transferability largely depends on the substitute models. To make our proposed PI-FGSM more convincing, we generate adversarial examples for ResNeXt$_{DA}$, Res152$_B$, Res152$_D$ and an ensemble of them respectively. In this case, we set $\beta = 2.5, \gamma = 1$. In Tab. 4, we sort these methods in an ascending order. As we can observe, our approach is superior to other methods by a large margin for both the white-box and black-box settings. However, existing methods' performance is even worse than I-FGSM, which is a basic iterative method. It also reminds us that when attacking several robust defense models, simply combining with different methods may not be effective.

Table 3. The success rate(%) of non-targeted attacks. We use an ensemble of Inc-v3, Inc-v4, Res-152, and IncRes-v2 to generate our adversarial examples by FGSM, I-FGSM, MI-FGSM, DI-FGSM, TI-FGSM, PI-FGSM, and their combined versions.

	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes$_{ens}$	ResNeXt$_{DA}$	Res152$_B$	Res152$_D$
FGSM	27.1	24.0	13.5	3.1	1.4	2.2
I-FGSM	26.2	25.2	16.0	0.7	0.8	0.4
MI-FGSM	51.9	49.3	32.9	2.8	1.8	2.2
DI-FGSM	40.5	38.5	25.6	1.2	1.5	1.2
TI-FGSM	39.3	38.9	31.5	6.1	3.7	3.0
PI-FGSM(Ours)	**61.0**	**62.8**	**51.3**	**8.7**	**8.5**	**6.0**
TPI-FGSM(Ours)	79.6	81.4	74.0	11.5	10.5	9.4
DPI-FGSM(Ours)	66.7	68.5	58.7	9.0	7.4	6.4
DTMI-FGSM	81.2	81.1	76.6	6.1	5.5	4.8
DTPI-FGSM(Ours)	**89.3**	**89.2**	**83.4**	**11.7**	**10.6**	**10.4**

Table 4. The average success rate(%) of non-targeted attacks. The top row models are substitute models ("*" indicates white-box attacks). We use ResNeXt$_{DA}$, Res152$_B$, Res152$_D$ and an ensemble of them to generate adversarial examples by FGSM, I-FGSM, DI-FGSM, TI-FGSM, MI-FGSM, PI-FGSM, and their combined versions.

	ResNeXt$_{DA}$			Res152$_B$			Res152$_D$			Ensemble		
Attack	ResNeXt$_{DA}$	Res152$_B$	Res152$_D$	ResNeXt$_{DA}$	Res152$_B$	Res152$_D$	ResNeXt$_{DA}$	Res152$_B$	Res152$_D$	ResNeXt$_{DA}$	Res152$_B$	Res152$_D$
DI-FGSM	58.3*	34.4	33.3	33.5	57.3*	32.5	34.7	34.8	56.8*	67.0*	66.8*	65.9*
TI-FGSM	66.7*	44.5	44.6	46.9	69.8*	44.3	48.6	46.6	65.3*	63.8*	62.1*	63.1*
FGSM	67.2*	45.7	45.0	47.5	71.3*	45.4	49.2	46.3	68.4*	67.2*	67.1*	65.7*
DTMI-FGSM	72.1*	51.0	52.3	52.4	74.0*	49.1	54.4	51.8	69.8*	69.6*	68.3*	65.4*
DMI-FGSM	72.5*	50.3	50.8	49.8	76.0*	48.3	51.7	48.9	72.2*	71.7*	71.9*	71.3*
MTI-FGSM	72.6*	50.1	52.1	51.8	77.7*	49.7	53.5	51.2	73.8*	70.6*	71.3*	68.4*
MI-FGSM	79.0*	54.0	54.4	56.6	81.5*	55.0	57.1	54.8	78.5*	75.3*	77.1*	75.7*
I-FGSM	81.6*	55.1	55.9	56.8	83.7*	55.9	57.7	55.8	82.5*	78.9*	79.6*	79.8*
PI-FGSM(Ours)	**86.9***	**62.0**	**60.8**	**62.0**	**88.1***	**61.9**	**62.7**	**62.4**	**86.6***	**84.8***	**82.5***	**83.2***

5 Conclusions

In this paper, we propose a novel patch-wise iterative algorithm – a black-box attack towards mainstream normally trained and defense models, which differs from the existing attack methods manipulating pixel-wise noise. With this approach, our adversarial perturbation patches in discriminative regions will be larger, thus generating more transferable adversarial examples against both normally trained and defense models. Compared with state-of-the-art attacks, extensive experiments have demonstrated the extraordinary effectiveness of our attack. Besides, our method can be generally integrated to any gradient-based attack methods. Our approach can serve as a baseline to help generating more transferable adversarial examples and evaluating the robustness of various deep neural networks.

References

1. Cissé, M., Adi, Y., Neverova, N., Keshet, J.: Houdini: Fooling deep structured prediction models. CoRR **abs/1707.05373** (2017) 4
2. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: CVPR (2018) 4, 5, 7, 9
3. Dong, Y., Pang, T., Su, H., Zhu, J.: Evading defenses to transferable adversarial examples by translation-invariant attacks. In: CVPR (2019) 2, 4, 5, 6, 7, 9, 11, 13
4. Dziugaite, G.K., Ghahramani, Z., Roy, D.M.: A study of the effect of JPG compression on adversarial images. CoRR **abs/1608.00853** (2016) 2
5. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: SIGGRAPH (2001) 2
6. Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., Song, D.: Robust physical-world attacks on deep learning visual classification. In: CVPR (2018) 1, 2, 4
7. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015) 2, 3, 5, 7
8. Guo, C., Rana, M., Cissé, M., van der Maaten, L.: Countering adversarial images using input transformations. In: ICLR (2018) 2
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 1, 9
10. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017) 1, 3, 9
11. Ilyas, A., Engstrom, L., Athalye, A., Lin, J.: Black-box adversarial attacks with limited queries and information. In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML (2018) 4
12. Komkov, S., Petiushko, A.: Advhat: Real-world adversarial attack on arcface face id system. CoRR **abs/1908.08705** (2019) 2, 4
13. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: ICLR (2017) 4
14. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial machine learning at scale. In: ICLR (2017) 2, 3, 5
15. Li, Y., Bai, S., Xie, C., Liao, Z., Shen, X., Yuille, A.L.: Regional homogeneity: Towards learning transferable universal adversarial perturbations against defenses. CoRR **abs/1904.00979** (2019) 6
16. Liu, A., Huang, T., Liu, X., Xu, Y., Ma, Y., Chen, X., Maybank, S., Tao, D.: Spatiotemporal attacks for embodied agents. In: European Conference on Computer Vision (2020) 4
17. Liu, A., Wang, J., Liu, X., Cao, b., Zhang, C., Yu, H.: Bias-based universal adversarial patch attack for automatic check-out. In: European Conference on Computer Vision (2020) 2
18. Liu, Y., Chen, X., Liu, C., Song, D.: Delving into transferable adversarial examples and black-box attacks. In: ICLR (2017) 4, 13
19. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: CVPR (2015) 6
20. Moosavi-Dezfooli, S., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: CVPR (2017) 4, 7
21. Papernot, N., McDaniel, P.D., Goodfellow, I.J., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Karri, R., Sinanoglu, O., Sadeghi, A., Yi, X. (eds.) AsiaCCS (2017) 4

22. Rosen, J.: The gradient projection method for nonlinear programming. part i. linear constraints. Journal of The Society for Industrial and Applied Mathematics **8**, 181–217 (1960) 7
23. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: nonlinear phenomena **60**(1-4), 259–268 (1992) 2
24. Salimans, T., Ho, J., Chen, X., Sutskever, I.: Evolution strategies as a scalable alternative to reinforcement learning. CoRR **abs/1703.03864** (2017) 4
25. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV (2017) 3
26. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: SIGSAC (2016) 2
27. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI (2017) 1, 9
28. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016) 1, 3, 9
29. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: ICLR (2014) 1, 4
30. Tabacof, P., Tavares, J., Valle, E.: Adversarial images for variational autoencoders. CoRR **abs/1612.00155** (2016)
31. Thys, S., Ranst, W.V., Goedemé, T.: Fooling automated surveillance cameras: adversarial patches to attack person detection. In: CVPR Workshops (2019) 2
32. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I.J., Boneh, D., McDaniel, P.D.: Ensemble adversarial training: attacks and defenses. In: ICLR (2018) 2, 9, 13
33. Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., Yuille, A.L.: Adversarial examples for semantic segmentation and object detection. In: ICCV (2017) 4
34. Xie, C., Wu, Y., van der Maaten, L., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: CVPR (2019) 2, 9, 13
35. Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., Yuille, A.L.: Improving transferability of adversarial examples with input diversity. In: CVPR (2019) 4, 5, 6, 7, 9
36. Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR (2016) 2