Hindawi
Security and Communication Networks
Volume 2022, Article ID 4447307, 8 pages
https://doi.org/10.1155/2022/4447307

Research Article

FAPA: Transferable Adversarial Attacks Based on Foreground Attention

Zhifei Yang, Wenmin Li, Fei Gao, and Qiaoyan Wen

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Wenmin Li; liwenmin@bupt.edu.cn

Received 19 June 2022; Revised 22 September 2022; Accepted 27 September 2022; Published 29 October 2022

Academic Editor: Hamad Naeem

Deep learning models are vulnerable to attacks by adversarial examples. However, current studies are mainly limited to generating adversarial examples for specific models, and the migration of adversarial examples between different models is rarely studied. At the same time, in only studies, it is not considered that adding disturbance to the position of the image can improve the migration of adversarial examples better. As the main part of the picture, the model should give more weight to the foreground information in the recognition. Will adding more perturbations to the foreground information of the image result in a higher transfer attack rate? This paper focuses on the above problems, and proposes the FAPA algorithm, which first selects the foreground information of the image through the DINO framework, then uses the foreground information to generate M, and then uses PNA to generate the perturbation required for the whole picture. In order to show that our method attaches importance to the foreground information, we give a greater weight to the perturbation corresponding to the foreground information, and a smaller weight to the rest of the image. Finally, we optimize the generated perturbation through the gradient generated by the dual attack framework. In order to demonstrate the effectiveness of our method, we have conducted relevant comparative experiments. During the experiment, we used the three white-box ViTs models to attack the six black-box ViTs models and the three black-box CNNs models. In the transferable attack of ViTs models, the average attack success rate of our algorithm reaches 64.19%, which is much higher than 21.12% of the FGSM algorithm. In the transferable attack of CNN models, the average attack success rate of our algorithm reaches 48.07%, which is also higher than 18.65% of the FGSM algorithm. By integrating ViTs and CNNs models, the attack success rate of transfer of our algorithm reaches 56.13%, which is higher than 1.18% of the dual attack framework we refer to.

1. Introduction

With the development of the transformer [1], it has more and more applications in natural language processing, such as BERT [2] and GPT [3], and more and more scholars are interested in whether it can replace CNNs as the backbone network in deep learning. Recently, the vision transformer (ViT) [4] only uses the transformer as the skeleton of the model to encode the remote dependence of image blocks through the multihead self-attention mechanism and has achieved a better classification effect than CNNs on the premise of requiring less training time. Based on this pioneering work, many variations were proposed to improve the performance of the ViTs. It includes improving the efficiency of training data [5] and introducing convolution [6–8] or a pooling layer [9, 10]. However, the ViTs models themselves are very vulnerable to attack from adversarial examples, which are indistinguishable from the original images under human observation but contain slight perturbations that can lead to misidentification by the model.

According to the master of the model structure, the attack can be divided into a white-box attack and a black-box attack. A white-box attack is an attack carried out by an attacker when the model, network structure, and weight parameters of the attacked are fully known. Black-box attacks, in contrast to white-box attacks, have no knowledge of the structure and parameters of the model. It is well known that the surrogate model can be used to generate adversarial examples and then use them to attack the black-box model without knowing the structure, which is also known as cross-model transferability [11], which is possible to study the transferability of adversarial examples.

High-performance transferable attacks usually apply data enhancement or advanced gradient calculations to prevent the overfitting of the perturbations and improve the transferability of the adversarial examples, but these are mostly used to attack CNNs. In contrast, less is known about the transferability of adversarial examples between ViT models. Due to the significant difference in the structure of the CNNs and ViTs, few scholars have transferred the methods applied to the CNNs to the ViTs. One related work [12] proposed a dual attack architecture: PNA and PatchOut, attacking both the attention mechanism and the patch simultaneously. The PNA attack mainly improves the success rate of adversarial example transfer attack by setting the attention weight calculated in the forward propagation process as a constant. Furthermore, the PatchOut attack is to randomly select a certain number of patches to generate perturbation. But in the article [12], there is no specific study on how to increase perturbation to improve the success rate of transfer attack and which part the model pays more attention to when identifying. BiCM was proposed in [13] to improve the coverage of the network. A new multilevel network structure was proposed by Wang et al. [14], which uses the pyramid features of different levels to extract the general and transferable features of the perturbation. Let us take inspiration from this, not all pixels are equally important for visual tasks. Does the model pay more attention to certain features in recognition? As the main part of the image, will the model pay more attention to the foreground information in the classification or recognition? In the process of adversarial example generation, whether adding more perturbations to the foreground part of the image will improve the attack success rate (ASR) of the model and the transferability of adversarial examples?

In this paper, based on [12], we mainly study whether adding more perturbations to the foreground information of the image will affect the success rate of the transfer attack. We propose the FAPA algorithm, which makes the model pay more attention to the foreground information of the image and adds more perturbations to the foreground of the image. We carried out relevant experiments in the ImageNet dataset with ASR as the index.

We briefly summarize the main contributions as follows:

(1) We find that adding more perturbations to the foreground information of the image can improve the success rate of the transfer attack when generating adversarial examples. Because not every pixel is equally important to the model, the model may pay more attention to the foreground part of the image when recognizing it.

(2) We propose the FAPA algorithm, compared with other methods, which increases the perturbations in foreground information with a greater probability on the basis of the overall perturbations being basically unchanged and the adversarial examples not producing local color blocks. Our algorithm takes the dual attack framework as its infrastructure and uses the DINO algorithm to select the patch corresponding to the foreground in the picture. In the perturbation increasing stage, we select the foreground information with a 40% probability and assign more weight to it. Similarly, we also chose background information with a 40% probability but assigned less weight to it. In addition, in order to make the generated adversarial examples not produce local color patches, we randomly select patches with a probability of 20%.

(3) We find that foreground information has different effects on CNNs and ViT models. Compared with CNNs, foreground information has less influence on ViT model, which may be due to the attention mechanism used in ViT model.

(4) We studied three different white-box ViTs to attack six black-box ViTs and three CNNs. The results show that the average success rate of the transfer attack of our algorithm in these nine models reaches 56.13%, which is better than [12]. Meanwhile, the average ASR of our method in the ViT models is as high as 64.19%, which is higher than 2.27% of the method [12].

2. Related Works

2.1. Vision Transformer. The transformer [1] uses a self-attention mechanism to achieve excellent performance in machine translation. Therefore, it has been further applied in many fields of natural language processing. Transformer mainly consists of two parts: the encoder and the decoder, each of which contains several blocks. Each block contains multihead self-attention layers and MLP layers. ViT uses the same structure as transformers, and their main difference is the image preprocessing layer. This layer in ViT splits the image into a series of nonoverlapping patches and then learns a linear projection. The ViT model requires a large amount of data for pretraining to achieve high accuracy. To address this issue, DeiT [5] added the teacher-student strategy in the transformer, which mainly uses a distillation token to make them acquire relevant knowledge from CNNs. TNT [15] uses an outer transformer block and an inner transformer block to learn the correlation between patches. Swin transformer [16] mainly applies a hierarchical structure similar to CNN to process pictures so that the model can flexibly process pictures of different scales. At the same time, the sliding window operation is introduced to reduce the computational complexity. Other ViT models include CaiT [17], LeViT [18], PiT [10], ConViT [19], and so on.

2.2. Transfer-Based Attacks on ViTs. Compared with CNN-based transferability attacks, the transferability of adversarial examples between different ViT models is relatively less studied. A study on the robustness of the ViT models [20] shows that ViTs not only exhibit good performance in multitask but also exhibit strong adversarial robustness. Spoofing ViTs in white-box scenarios require a larger noise magnitude [21], and existing shift-based black-box attacks also have difficulty in transferring adversarial examples from CNNs to ViTs. A study [22] found that the good performance of transformers on occlusion is not due to the dependence on local texture information; compared with CNNs, ViTs have much less dependence on texture. When properly trained to encode shape-based features, ViTs can show shape recognition capabilities comparable to those of the human visual system. However, existing ViTs focus on standard accuracy and computational cost, and there is a lack of research on the intrinsic impact of model robustness and generalization. In [22], the authors conducted a systematic study of the robustness of ViT's components to adversarial examples, common corruptions, and distribution changes, and they found that some components may be detrimental to the robustness of the model. By using and combining powerful components as building blocks of ViTs, [22] the robust vision transformer (RVT) is proposed. Regarding the transferability of ViTs, a self-ensemble (SE) method was proposed in [23], which optimizes perturbations on the ensemble model to improve the mobility of adversarial examples, but they ignored the effect of increased location of perturbations on the success rate. Aiming at the patch input and multi-headed self-attention (MSA) module in the ViT structure, the authors in [12] proposed a dual attack framework, which includes a pay no attention (PNA) attack and a PatchOut attack, to improve the transferability of adversarial examples between different ViTs and even between ViTs and CNNs. The PNA attack has a wide range of applications and can be used for any gradient-based attack method. The PatchOut attack uses a different patch as input to generate adversarial examples in each iteration. However, for [12], there is no relevant research on whether to pay more attention to which specific part in model recognition. Our proposed FAPA algorithm mainly studies the influence of the position with increased pertubation on the success rate of transfer attack and explores whether the model recognition will pay more attention to the foreground information of the picture.

2.3. DINO. The framework DINO [24] simplifies the self-supervised training process to predict the output of a teacher network through cross-entropy loss, which can be interpreted as a label-free knowledge distillation. In addition to the good performance of self-supervised learning in this framework, it is also observed that self-supervised ViT features contained clear image semantic segmentation information, especially the object boundary, which could be directly accessed in the self-attention module of the last block so as to further use it to locate foreground pixels. This is not seen in the supervised ViTs and CNN models. Therefore, this can be used to complete the image foreground extraction.

3. Method

In this section, we will detail the general framework of the model and our proposed FAPA algorithm. Meanwhile, our framework is based on the dual attack architecture proposed in [12]. Suppose we have a training set $X = \{x_i\}_{i=1}^N$, which contains N images with N corresponding labels $Y = \{y_i\}_{i=1}^N$. The general idea of the model is to input the sample x_i into the FAPA algorithm to generate the required perturbation δ, add it to x_i to generate an adversarial example x_{adv}, and input x_{adv} into the white-box model with PNA to calculate loss thereby updating δ. After F iterations, x_{adv} will be input as the final adversarial example into the black-box model to be attacked, and the model will produce the predicted result y. We want to make y as different from the real tag y_i as possible.

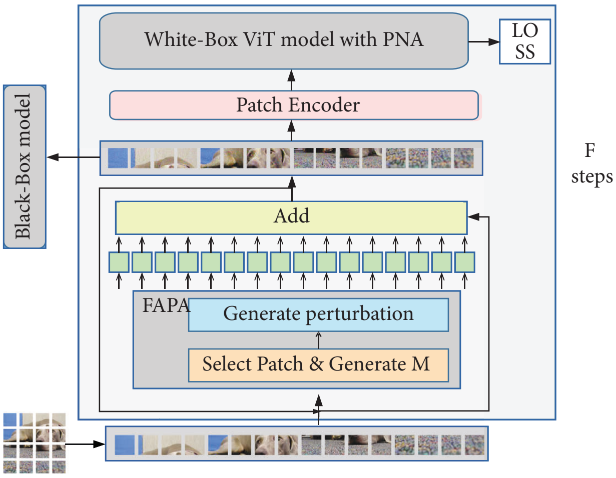

Figure 1: The framework of the proposed method. It mainly includes the FAPA algorithm, patch encoder, white-box model, and black-box model.

3.1. Overall Framework. Compared with [12], the FAPA algorithm proposed in this paper is applied before the image is input into the white-box model, which can add more perturbations to the foreground of the image. An overview of our architecture is illustrated in Figure 1. Suppose we input a labeled image $x \in X \subset R^{H\times W\times C}$, its ground-truth label $y \in Y = \{1, \ldots, K\}$, where H, W, and C, respectively, represent the width, height, and number of channels of the image, and K represents the number of classified categories. We divide x into N patches, $x = \{x_1, x_2, \ldots, x_N\}$, $x_p \in R^{P\times P\times C}$, where (P, P) represents the size of each patch, $N = H \cdot W/P^2$. We can see these operations clearly in the bottom part of Figure 1. Assuming that the ith iteration is now being performed. We input the segmented x into the FAPA algorithm module, which includes two parts: selecting patches to generate M and generating perturbations. In the first part, the DINO algorithm is applied to select the required foreground or background patches, and then the corresponding patches are set with corresponding weights to obtain M. The second part is to generate the perturbation δ_i based on M and the perturbation δ_{i-1} generated in the last iteration. We add the perturbation δ_i to the image x to generate the corresponding adversarial example x_{adv} at the ith iteration. Next, we input x_{adv} into the patch encoder module for encoding and then into a white-box model with PNA [12] to compute the loss. Next, the gradient of the perturbation is calculated according to the loss, and then the gradient descent algorithm is used to update the generated perturbation δ_i. The whole process requires F

iterations, which are represented by the largest gray-box in Figure 1. The $x_{a\ dv}$ that is finally input into the black-box model is produced after F iterations. We use $f(x)\colon X \longrightarrow Y$ to represent the whole process of the white-box model. During the whole attack process, we use the method of untargeted attack and enforce an $L\infty$-norm constraint on perturbations to make it within $[-\varepsilon, \varepsilon]$. We use δ for the final added perturbation, and J is the cross-entropy loss. The formula for the entire model can be expressed as follows:

$$\arg_{\delta}\max J(f(x+\delta), y), s.t.\|\delta\|_{\infty} < \varepsilon. \tag{1}$$

3.2. FAPA Algorithm. In order to better select the patch related to the foreground of the image so that the model can add more perturbations to the foreground information of the image and improve the accuracy of the model transfer attack, we propose the FAPA algorithm, which uses the DINO as the foreground patch selection model because [24] found that the self-supervised ViT features contain clear image semantic segmentation information and can be used to extract foreground pixels.

We summarize the proposed FAPA in Algorithm 1. For an input image x, we divide it into N patches, then $x = \{x_1, x_2, \ldots, x_N\}$, x_i represents a patch in the image, and the total perturbation is divided into F times. The M is an attack mask, $M \in \{0, 1, u, v\}^{H\times W\times C}$, where $u < v$ and $u + v = 2$. We utilize the self-supervised ViT model DINO to efficiently segment foreground information in images. In the iterative process from 0 to $2/5 \cdot F$, we use the DINO algorithm to select the foreground information of T patches in the picture, use A to represent it, set the M value of the corresponding patch position to v, and the rest of the position is set to 0. Here, more perturbation can be added to the foreground information of the picture. Then, in the $2/5 \cdot F$ to $4/5 \cdot F$ iteration process, we also use the DINO algorithm to select the background information of the T patches in the figure, use C to represent it, and put the M of the corresponding patch position to u, and the rest of the positions are 0. The operation here is to make fewer perturbations added to the background information, and $u + v = 2$ is to make the perturbations added in the whole process tending to remain unchanged as a whole. In the last $1/5 \cdot F$ iterations, we randomly select T patches to attack and denote the selected patch set by B, and inspired by the authors in [25], diverse input modes can improve the transferability of adversarial examples and alleviate model overfitting. At the same time, this can avoid local color blocks in the generated adversarial examples, making it more difficult for humans to detect. For patches that are not selected during each iteration, we use the D set to represent them; the value in D set is set to 0. Therefore, the value of the entire M can be expressed by the following formula:

$$M^i = \begin{cases} v, & x_i \text{in} A, \\ 1, & x_i \text{in} B, \\ u, & x_i \text{in} C, \\ 0, & x_i \text{in} D. \end{cases} \tag{2}$$

So, the function in equation (1) can be expressed as follows, where $\odot$ denotes element-wise multiplication, and L_2 norm can better prevent δ overfitting.

$$\arg_{\delta}\max J(f(x + M \odot \delta), y) + \lambda\|\delta\|_2, s.t.\|\delta\|_{\infty} < \varepsilon. \tag{3}$$

Assuming that this is the kth iteration, after generating M, we use the PNA module as shown in equation (3), and the gradient descent algorithm is used to optimize the generated perturbation so as to obtain g. Finally, we multiply g by the increasing perturbation size α at each iteration to obtain δ_k. After F iterations, our final δ_F is our required perturbation, which we add to image x to generate the final adversarial example x_{adv}.

4. Experiment

4.1. Datasets. We selected 1000 images from the ImageNet validation set as our laboratory data, and any one of them could be correctly classified when fed into the model used in this experiment. Each image is a 224×224 three-channel color map with a size of around 100 kB. A sample image is shown in the "Clean" column in Figure 2.

4.2. Models. In order to evaluate the quality of our method, we take the migration ASR of adversarial examples generated on the white-box model as the index when they are migrated to the black-box model. The ViT-related models were selected as PiT-B [10], CaiT-S-24 [17], Visformer-S [26], TNT-S [15], LeViT-256 [18], and ConViT-B [19]. CNN-related models are Inception v3 [27], Inception v4, and Inception ResNet v2 [28]. We randomly select one model among PiT-B, CaiT-S-24, and Visformer-S as the white-box model to generate adversarial examples, and the other models are used as black-box models to accept the attack. We choose these models as the usage models for our experiments because they are exposed in the timm [29] module and can be easily invoked. When using the DINO model for foreground selection of pictures, both the teacher and student network in the model use the DeiT-S/16 [5].

4.3. Parameter. We set $\varepsilon = 16$, and the number of iterative attacks $F = 10$. The input size of the image is 224×224, and then it is divided into $N = 196$ patches with $P = 16$ in each patch. We select $T = 130$ patches as the target patch for attack. When using the FAPA algorithm, M is processed as shown in equation (2), where $u = 0.7$ and $v = 1.3$. In the whole experiment, we use the cross-entropy loss function, and the evaluation indicator adopts the attack success rate (ASR), that is, the probability that the model will misclassify the image by inputting the data into the black-box model. Higher ASR means higher migration of adversarial samples.

4.4. Baselines. We will compare our method with typical algorithms, such as FGSM [30], MI [31], SIM [32], SGM [33], TAP [34], and so on. These traditional algorithms have been tested on ViTs and CNN models, respectively, and the experimental environment is the same as the FAPA algorithm. The specific settings are shown in 4.3. The experiment's specific data are shown in Tables 1 and

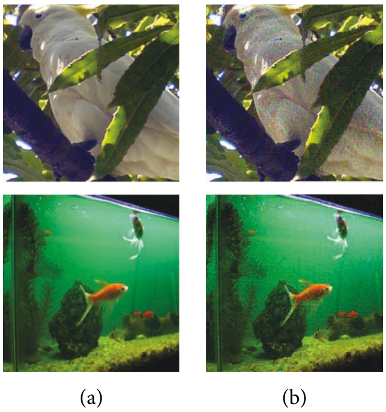

Figure 2: Visualization of randomly selected clean images and corresponding adversarial images produced by our FAPA on Visformer-S model. (a) Clean. (b) Adversarial.

```
Input: The loss function $J$ of equation (2), a white-box model $f$, a clean image $x$ with its ground-truth class $y$, DINO model, $u$, $v$, the perturbation budget $\varepsilon$, iteration number $F$, used patch number $T$.
Output: The adversarial example $x_{adv}$.
$\delta_0 \leftarrow 0$
$\alpha \leftarrow \varepsilon / F$
for $k = 0$: $F - 1$ do
    if $0 \leq k < 2/5 \cdot F$ then
        $x_i \leftarrow$ DINO $(x, T)$, $x_i$ is foreground patch
        $M \leftarrow$ equation (2), ($x_i$ use $v$, other use 0)
    if $2/5 \cdot F \leq k < 4/5 \cdot F$ then
        $x_i \leftarrow$ DINO $(x, T)$, $x_i$ is background patch
        $M \leftarrow$ equation (2), ($x_i$ use $u$, other use0)
    if $4/5 \cdot F \leq k < F$ then
        $x_i \leftarrow$ PatchOut $(x, T)$, $x_i$ is random patch
        $M \leftarrow$ equation (2), ($x_i$ use 1, other use 0)
    $g \leftarrow$ PNA ($\nabla_\delta J$ with the $L_2$ norm)
    $\delta_k \leftarrow \mathrm{clip}_\varepsilon (\delta_{k-1} + \alpha \cdot g)$
$x_{adv} = x + \delta_F$
return $x_{adv}$
```

Algorithm 1: FAPA based on the dual attack on ViTs.

Table 1: Attack success rate of the typical algorithm in the ViT model (%).

Method	PiT-B	CaiT-S-24	Visformer-S	TNT-S	LeViT-256	ConViT-B	Mean
FGSM	19.80	20.43	19.37	22.78	18.80	25.58	21.12
BIM	22.17	22.63	22.70	32.13	20.45	35.30	25.89
MI	45.23	47.13	45.97	55.23	43.75	58.25	49.26
DI	45.13	43.07	47.77	55.18	43.25	49.35	47.29
TI	17.67	16.50	19.00	28.18	13.70	27.53	20.43
SIM	32.73	35.17	31.13	46.73	36.43	45.68	37.98
SGM	41.60	52.30	48.80	64.33	51.13	60.68	53.14
IR	22.70	24.00	23.43	33.43	21.30	36.38	26.87
TAP	24.73	33.40	32.20	39.78	30.03	42.20	33.72
ATA	1.13	0.97	2.67	3.37	2.02	3.72	2.31
SE	21.25	31.40	24.90	37.87	21.73	46.03	30.53

2, and the data are obtained from [12]. The vertical axis in the table represents the white-box model, and the horizontal axis represents the black-box model. The numbers in it, respectively, represent the ASR when the adversarial examples are generated by the white-box model to attack the black-box model. Each number in the rightmost column of the table represents the average ASR for that row. The horizontal and vertical axes of each subsequent table and the meaning of the figures in the table have been described previously. Meanwhile, in

Table 2: Attack success rate of the typical algorithm in the CNN model (%).

Method	Inc-v3	Inc-v4	IncRes-v2	Mean
FGSM	20.78	18.80	16.38	18.65
BIM	17.88	14.77	12.40	15.02
MI	39.65	37.43	32.17	36.42
DI	32.78	31.75	26.40	30.31
TI	23.27	23.60	15.28	20.72
SIM	30.55	27.63	24.17	27.45
SGM	38.42	34.00	27.25	33.22
IR	17.65	15.83	12.08	15.19
TAP	29.58	26.10	20.67	25.45
ATA	3.25	2.53	2.03	2.60
SE	18.40	16.47	12.33	15.73

Table 3: Comparison with state-of-the-art methods (%).

Method	ViTs	CNNs	Mean
[12]	61.92	47.98	54.95
Ours	64.19	48.07	56.13

Table 4: Attack success rate of our method in the ViT model (%).

Method	PiT-B	CaiT-S-24	Visformer-S	TNT-S	LeViT-256	ConViT-B	Mean
PiT-B	Null	59.30	72.60	70.00	64.00	57.30	64.64
CaiT-S-24	52.50	Null	57.90	71.60	58.30	76.50	63.36
Visformer-S	68.10	58.70	Null	73.40	71.20	51.50	64.58
Mean	60.30	59.00	65.25	71.67	64.50	61.77	64.19

order to illustrate the effectiveness of our algorithm, we also compare it with the most advanced algorithms, as shown in Table 3.

4.5. Performance on ViTs and CNNs. We evaluate the transferability of adversarial examples generated by our method in ViTs and CNN models. Tables 4 and 5 summarize the results of our three white-box ViT models attacking various black-box ViTs and CNN models. In Tables 4 and 5, the vertical axis represents the white-box model, and the horizontal axis represents the black-box model. The null value in the table indicates that the case where the white-box model and the black-box model are identical is not counted. As can be seen from the table, our method has high transferability in the ViT models and CNN models, and the overall average ASR reaches 64.19% and 48.07%.

Table 5: Attack success rate of our method in the CNN model (%).

Method	Inc-v3	Inc-v4	IncRes-v2	Mean
PiT-B	51.80	50.40	38.40	46.87
CaiT-S-24	50.90	48.80	41.30	47.00
Visformer-S	51.70	58.30	41.00	50.33
Mean	51.47	52.50	40.23	48.07

4.6. Comparison with Typical Algorithm in the ViT Models. It can be seen from Tables 1 and 4 that adversarial examples generated by our method have high transferability in the ViT models, with an overall average ASR of 64.19%, far higher than the highest average ASR of 53.14% generated by the SGM algorithm in Table 1. In addition, when the CaiT-S-24 model is used as a proxy to attack the ConViT-B model, our method achieves 76.50%, which is far higher than the 25.58% accuracy achieved by the FGSM algorithm on the ConViT-B model. Meanwhile, the average ASR of our three models on TNT-S reaches 71.67%, which is also higher than the 64.33% average ASR generated by the SGM algorithm.

4.7. Comparison with the Typical Algorithm in the CNN Models. As can be seen from Table 5, when adversarial examples generated on the ViT models are migrated to the CNNs models, the success rate of the attack decreases, which further indicates that there are essential differences in architecture between ViTs and CNN models. However, compared with the algorithms in Table 2, our method still achieves good results, with an average ASR of 48.07%, which exceeds the highest average ASR of 36.42% achieved by using the MI algorithm in Table 2. When the Visformer-S model is used to attack the Inc-v4 model, our method even achieves a 58.30% ASR, which is far higher than the 20% ASR of any model in Table 2. Meanwhile, when our method attacks the Inc-v4 model, the average

Table 6: Ablation studies are conducted on the ImageNet dataset (%).

u	v	ViTs	CNNs	Mean
1.0	1.0	61.92	47.98	54.95
0.9	1.1	63.19	45.47	54.33
0.8	1.2	63.56	46.90	55.23
0.7	1.3	64.19	48.07	56.13
0.6	1.4	63.78	48.47	56.13
0.5	1.5	63.75	49.80	56.78

success rate of the attack also reaches 52.50%, which is a good result. Compared with each algorithm in Table 2, our method has made great progress, and the ASR is higher than any algorithm in Table 2. This shows that compared with the traditional algorithm, our algorithm is effective and feasible.

4.8. Comparison with State-of-the-Art Methods. In this section, we compare the results of our proposed FAPA and the state-of-the-art [12] methods in Table 3. Our average ASR exceeded it by 1.18% and was superior to it in both ViTs and CNNs, particularly evident in ViTs. Focusing on the ViT model, the average ASR of our method reaches 64.19%, which is higher than [12] 2.27%, which further demonstrates the effectiveness of our algorithm.

4.9. Ablation Study. To illustrate the effectiveness of our method, we performed ablation experiments, and the results are shown in Table 6. Table 6 shows the influence of different values of u and v in the FAPA algorithm on the success rate of the model attack. The horizontal axis shows two different models. The value in the table represents the average ASR of the algorithm on the model, and the larger the value, the stronger the migration attack ability. The larger the value of v is, the algorithm pays more attention to the foreground information in the picture, which leads to larger perturbations. When u and v are both 1, it is equivalent to not using our method, and the accuracy is low at this time. As can be seen from the table, as the perturbations are added to the foreground information of the picture, the overall ASR of the model is constantly rising, which shows that our method is effective and feasible. For the ViT models, the values of u and v reached the highest value of 64.19% when they were set to 0.7 and 1.3, respectively. But for the CNN models, their ASR keeps increasing with the increase of foreground attention. It can be further seen from here that there are some differences between the ViTs and CNN models in structure. Compared with CNNs, the ViT models pay attention to the foreground information while also taking into account the global information, which may be due to the use of the attention mechanism in the ViT models.

4.10. Visualization. In Figure 2, we randomly select two clean images and two images generated by our algorithm. We can easily find that the adversarial samples generated by our algorithm are imperceptible to the naked eye.

5. Conclusion

In order to study the influence of the position with increased perturbation on the success rate of the transfer attack and explore whether the model will pay more attention to the foreground information of the picture in the recognition process, we propose the FAPA algorithm. The FAPA algorithm can achieve, add more perturbations to the foreground information of the image. Experiments show that the transfer attack success rate of our method in ViTs and CNNs reaches 56.13%, which is 1.18% higher than that of the dual attack framework. Meanwhile, the average attack success rate of our algorithm on the ViT models reaches 64.19%, which is 2.27% higher than that of the dual attack framework. For foreground and background information, we found that the best transfer attack success rate was achieved with weights of 1.3 and 0.7, respectively. From these experiments, it can be concluded that the model will pay more attention to the foreground information of the picture in the recognition process, and adding more disturbance to the foreground information of the picture can improve the success rate of the transfer attack. In addition, compared with CNNs, the ViT models not only pay attention to the foreground information of an image but also consider the global information of the image to some extent, which may come from the attention mechanism used in the ViT models. Therefore, when attacking the ViT models, we should take into account the global information while adding more disturbance to the foreground information. How to better take into account global and local information will be our next main research direction.

Data Availability

The data used to support the findings of this study are available from the authors upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by NSFC (Grant nos. 62072051, 61976024, and 61972048), 111 Project B21049, and BUPT Innovation and Entrepreneurship Support Program (2022-YC-A201).

References

[1] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: pre-training of deep bidirectional transformers for language understanding,” 2018, https://arxiv.org/abs/1810.04805.

[3] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training,” 2018, https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov et al., “An image is worth 16x16 words: transformers for image recognition at scale,” 2020, https://arxiv.org/abs/2010.11929.

[5] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, “Training data-efficient image transformers & distillation through attention,” in *Proceedings of the International Conference on Machine Learning*, pp. 10 347–410 357, PMLR, Shenzhen, China, March 2021.

[6] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, “Localvit: bringing locality to vision transformers,” 2021, https://arxiv.org/abs/2104.05707.

[7] H. Wu, B. Xiao, N. Codella et al., “Cvt: introducing convolutions to vision transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22–31, Montreal, Canada, October 2021.

[8] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, and W. Wu, “Incorporating convolution designs into visual transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 579–588, Montreal, Canada, October 2021.

[9] W. Wang, E. Xie, X. Li et al., “Pyramid vision transformer: a versatile backbone for dense prediction without convolutions,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 568–578, Montreal, Canada, October 2021.

[10] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, “Rethinking spatial dimensions of vision transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11 936–1011 945, Montreal, Canada, October 2021.

[11] Y. Liu, X. Chen, C. Liu, and D. Song, “Delving into transferable adversarial examples and black-box attacks,” 2016, https://arxiv.org/abs/1611.02770.

[12] Z. Wei, J. Chen, M. Goldblum, Z. Wu, T. Goldstein, and Y.-G. Jiang, “Towards transferable adversarial attacks on vision transformers,” 2021, https://arxiv.org/abs/2109.04176.

[13] S. Ashraf, O. Alfandi, A. Ahmad et al., “Bodacious-instance coverage mechanism for wireless sensor network,” *Wireless Communications and Mobile Computing*, vol. 2020, 11 pages, Article ID 8833767, 2020.

[14] H. Wang, G. Wang, Y. Li, D. Zhang, and L. Lin, “Transferable, controllable, and inconspicuous adversarial attacks on person re-identification with deep mis-ranking,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 342–351, Seattle, WA, USA, June 2020.

[15] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, “Transformer in transformer,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[16] Z. Liu, Y. Lin, Y. Cao et al., “Swin transformer: hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Montreal, Canada, October 2021.

[17] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jegou, “Going deeper with image transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 32–42, Montreal, Canada, October 2021.

[18] B. Graham, A. El-Nouby, H. Touvron et al., “Levit: a vision transformer in convnet’s clothing for faster inference,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12 259–312 269, Montreal, Canada, October 2021.

[19] S. d’Ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun, “Convit: improving vision transformers with soft convolutional inductive biases,” in *Proceedings of the International Conference on Machine Learning*, pp. 2286–2296, PMLR, Montreal, Canada, October 2021.

[20] S. Paul and P.-Y. Chen, “Vision transformers are robust learners,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, pp. 2071–2081, Pennsylvania, USA, March 2022.

[21] R. Shao, Z. Shi, J. Yi, P.-Y. Chen, and C.-J. Hsieh, “On the adversarial robustness of vision transformers,” 2021, https://arxiv.org/abs/2103.15670.

[22] M. M. Naseer, K. Ranasinghe, S. H. Khan, M. Hayat, F. Shahbaz Khan, and M.-H. Yang, “Intriguing properties of vision transformers,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 296–323 308, 2021.

[23] M. Naseer, K. Ranasinghe, S. Khan, F. S. Khan, and F. Porikli, “On improving adversarial transferability of vision transformers,” 2021, https://arxiv.org/abs/2106.04169.

[24] M. Caron, H. Touvron, I. Misra et al., “Emerging properties in self-supervised vision transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, Montreal, Canada, October 2021.

[25] C. Xie, Z. Zhang, Y. Zhou et al., “Improving transferability of adversarial examples with input diversity,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2730–2739, Long Beach, CA, USA, June 2019.

[26] Z. Chen, L. Xie, J. Niu, X. Liu, L. Wei, and Q. Tian, “Visformer: the vision-friendly transformer,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 589–598, Montreal, Canada, October 2021.

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, Honolulu, Hawaii, July 2016.

[28] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Proceedings of the Thirty-first AAAI conference on artificial intelligence*, San Francisco, California, USA, February 2017.

[29] R. Wightman, “PyTorch image models,” 2019, https://github.com/rwightman/pytorch-image-models.

[30] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2014, https://arxiv.org/abs/1412.6572.

[31] Y. Dong, F. Liao, T. Pang et al., “Boosting adversarial attacks with momentum,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9185–9193, Salt Lake City, Utah, June 2018.

[32] J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft, “Nesterov accelerated gradient and scale invariance for adversarial attacks,” 2019, https://arxiv.org/abs/1908.06281.

[33] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, and X. Ma, “Skip connections matter: on the transferability of adversarial examples generated with resnets,” 2020, https://arxiv.org/abs/2002.05990.

[34] W. Zhou, X. Hou, Y. Chen et al., “Transferable adversarial perturbations,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 452–467, Munich, Germany, September 2018.