1. Introduction

Deep Neural Networks (DNNs) have demonstrated excellent performance in tasks such as image classification [14], object detection [31], and semantic segmentation [34]. However, their vulnerability to adversarial attacks has attracted a great deal of attention [8, 15]. Adversarial examples are generated by applying imperceptible perturbations to the input data that are sufficient to mislead DNNs to produce false predictions. This problem is particularly critical in security-critical domains (, autonomous driving [51], medical image analysis [25], and identity verification [33]) and can have disastrous consequences. The study of the generation mechanism of adversarial attacks has become an important topic to be addressed.

Adversarial attacks are an important research area for evaluating the robustness of deep neural networks (DNNs), where imperceptible perturbations are applied to input data to induce model misclassification. Attacks are categorized mainly as white-box or black-box. White-box attacks utilize the complete structure and parameters of the target model to generate perturbations through gradient optimization , the Fast Gradient Sign Method (FGSM) [8], the Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [5], and the Carlini and Wagner Attack (C&W) [1]. Black-box attacks are based only on model outputs and are more relevant to practical scenarios [30, 46]. Adversarial examples made in white-box settings often exhibit a degree of transferability between different models, , perturbations generated on one model may deceive other models [22, 26], which can be regarded as a black-box attack. However, existing white-box attack methods exhibit limited transferability when applied in a black-box setting [29]

The study of adversarial attacks can help develop more robust models. Existing research presents various methods to enhance the transferability of adversarial attacks[44], , Gradient-based [16], Advanced objective [21], Model-related [20], Ensemble-based [39], and Generation-based [2]. In particular, input transformation-based attacks are a significant method [9]. This field is divided into two subdomains: input transformation operators and operator combinations. Through a synthesis of the findings in [42] and [19], we conclude that, to generate adversarial perturbations with strong transferability, it is crucial to ensure that the transformed images exhibit rich attention heatmaps while maintaining global semantic consistency. However, existing input transformation operator methods fail to achieve both of these aspects simultaneously. To this end, this paper proposes a novel input transformation operator, Block Stretch and Shrink (BSS), which enriches attention heatmaps through block level stretch and shrink while preserving global semantics, thereby enhancing the transferability of perturbations across models.

We also examine the impact of the number scale on the performance of input transformation-based methods. Accordingly, we design controlled experiments with unified number scales to establish a fair and consistent framework for performance comparison, thereby enhancing the comparability of different methods. In summary, our contributions are as follows.

- We divide the field of input transformation-based attacks into two research directions: individual operator design and operator combination strategies.
- We reveal through analysis of input transformation-based methods that simultaneously promoting diverse attention heatmaps and preserving global semantics is crucial for enhancing transferability of adversarial attacks, and point out that existing operator methods struggle to achieve both aspects simultaneously.
- We design the Block Stretch and Shrink (BSS) method to enrich attention heatmaps while preserving global semantics and enhancing the transferability of adversarial attacks.
- We propose an experimental design with unified number scale, highlighting the impact of number scale on input transformation performance.

2. Related Work

2.1. Adversarial Attacks

Adversarial attacks generate adversarial examples by applying imperceptible perturbations to input data, causing deep neural networks to make incorrect predictions. Szegedy , [38] first revealed that DNNs are vulnerable to adversarial examples. Goodfellow , [8] proposed the Fast Gradient Sign Method (FGSM), which generates perturbations using a single-step gradient update. Kurakin , [15] extended FGSM to an iterative version, I-FGSM, which optimizes perturbations through multiple iterations, improving the effectiveness of white-box attacks. However, adversarial examples generated in white-box settings have limited transferability to black-box models[22]. To address this, momentum-based attacks, . MI-FGSM and NI-FGSM enhance transferability by stabilizing the optimization direction, making them more effective between different models[5, 18]

In addition to gradient-based methods, input transformation-based methods also represents a significant avenue to enhance the transferability of adversarial attacks. Research in this area mainly falls into two subdomains. The first involves developing efficient input transformation operators. DIM [48] proposes randomly resizing and padding images to generate multi-scale inputs. TIM [6] employs translation invariance to create multi-position inputs. SIM [18] using scale to generate transformed inputs. Admix [43] proposes to mix images from different categories to enhance input diversity, while DeCoWA [19] employs

deformation-constrained warping to augment input images. BSR [42] increasing transferability by diversifying attention heatmaps through random block shuffling and rotation.

Beyond input transformation operators, another line of research explores the combination of operators. AITL [52] pioneered the optimization of combinations of multiple transformation operators to improve transfer performance. SIA [45] demonstrated the effectiveness of applying operators individually to segmented image blocks. L2T [54] dynamically learns transformation strategies to adaptively select and apply operators during the attack process. OPS [9] increasing transferability of adversarial perturbation by integrating diverse transformation operators with random perturbations. These approaches collectively demonstrate that diversifying image input can prompt models to enhance the transferability of adversarial examples.

2.2. Adversarial Defense

To mitigate the threat posed by adversarial attacks, researchers have developed a variety of defense strategies. In 2018, Madry , proposed PGD-based Adversarial Training (AT) to enhance robustness, although it incurs high computational costs [26]. In the same year, Tramér , introduced Ensemble Adversarial Training [41], resulting in robust models such as ens3_adv_Inception-V3, ens4_adv_Inception-V3, and ens2_adv_Inception-ResNet, which leverage the transferability of adversarial examples across multiple pre-trained models to improve defense efficiency. Furthermore, in 2017, Xu , introduced feature squeeze techniques, including bit-depth reduction (Bit-RD), to compress input features and detect adversarial examples [50]. In 2018, Liao , proposed high-level representation-guided denoisers (HGD) to purify adversarial perturbations [17]. In 2019, Cohen , employed random smoothing (RS) to train certifiably robust classifiers [4]. In 2020, Wong , developed a single-step adversarial training approach that reduces computational costs while maintaining robustness [47]. Naseer , (2020) introduced Neural Representation Purifiers (NRP), designed to defend against transferable adversarial attacks [28].

2.3. Attention Heatmap

Deep neural networks exhibit complex decision-making behavior, which required interpretation tools to reveal their inner mechanisms [24]. Attention heatmaps serve as a key tool in this regard, visualizing the image regions that the model focuses on during classification, and thereby providing insights into the basis of its decisions. . Class Activation Maps (CAMs) highlight discriminative regions through global average pooling [53], while Grad-CAMs use gradient information to produce more precise and class-specific heatmaps [32].

Research has shown that generating perturbations with high transferability requires simultaneously promoting a diverse attention heatmap and preserving the global semantic integrity of the transformed input images [19, 42]. This technique plays an important role in analyzing the model's attention on the transformed input images.

3. Methods

3.1. Preliminaries

An adversarial example $x_{\text{adv}} = x + \delta$ is obtained by adding a small perturbation δ to the input x so that $f(x_{\text{adv}}) \neq y$, with the perturbation constrained by $\|\delta\|_{\ell_p} \leq \epsilon$, where we adopt $\ell_p = \ell_\infty$ unless otherwise stated. The attacker maximizes the loss $J(f(x+\delta), y)$ by optimizing δ. Dong , [5] proposed Momentum Iterative FGSM (MI-FGSM), which optimizes the gradient update using a momentum term to enhance transferability in black-box scenarios, with the update formula:

$$g_t = \mu \cdot g_{t-1} + \frac{\nabla_{\delta_t} J(f(x+\delta_t), y)}{\|\nabla_{\delta_t} J(f(x+\delta_t), y)\|_1} \tag{1}$$

$$\delta_{t+1} = \delta_t + \alpha \cdot \operatorname{sign}\left(g_t\right) \tag{2}$$

where μ is the momentum decay factor and $\alpha = \epsilon / T$.

3.2. Motivation

In the field of input transformation-based adversarial attack, generating perturbations with high transferability requires simultaneously considering both the generation of diverse attention heatmaps and the preservation of the global semantic integrity of the transformed images. Diverse attention heatmaps enhance the transferability of perturbations by encouraging the model to focus on broader image regions during optimization, thereby generating more robust adversarial features that effectively improve the success of cross-model attack[42].Similarly, input transformations that preserve global semantic integrity enable surrogate models to leverage shared invariant features, significantly boosting cross-model transferability [19].

However, existing input transformation-based methods face limitations. SIM [18], DIM [48], and TIM [6] maintain complete semantics through minor geometric adjustments (, resize scale and translation), but the diversity of attention

heatmaps is limited. BSR [42] and SIA [45] generate various attention heatmaps but have limited protection for global semantics. DeCoWA [19] preserves global semantics via non-rigid input-space warping, but the diversity of its induced attention heatmaps remains limited. The OPS [9] method simultaneously preserves both aspects at high number scales, thereby generating perturbations with high transferability; however, at low number scales, its few combination plans fail to provide sufficiently rich and diverse features for generating varied attention heatmaps, resulting in performance comparable to other methods.

To this end, we propose a constrained random segmentation mechanism with threshold-based point selection and a block stretch and shrink operator to enrich attention heatmaps while preserving global semantic integrity. For segmentation, we enforce minimum distances to image boundaries (d_b) and between adjacent points (d_p) to ensure that each block contains sufficient semantic content, thereby maintaining the global structure during stretch and shrink. The block-level stretch and shrink operator introduces controlled local scale variations across blocks while better matching the square, locally linear receptive fields of CNNs, thereby enriching attention heatmaps and preserving global semantics.

As shown in Firgure 2, BSS generates diverse distributions while preserving global semantics; in comparison, BSR disrupts semantics, DeCoWA lacks attention richness, and OPS shows different attention distributions only in scale-based combinations.

3.3. Evaluating Input Transformation-based Methods under a Unified Number Scale

Data augmentation improves the generalization of neural networks by generating diverse training samples without altering the model parameters [12]. OPS [9] generalizes this principle to the hypothesis space, viewing input transformation-based attacks as a form of data augmentation applied to the surrogate model, thereby enhancing adversarial transferability.

Input transformation-based attacks often rely on ensembles of transformed inputs to improve optimization [48]. However, the resulting number scale is rarely quantified or standardized. This lack of standardization complicates fair efficiency comparisons. In contrast, data augmentation studies typically evaluate methods under a bounded number of augmented samples per input to ensure reproducible and comparable efficiency [35]. Following this principle, we require experiments to be conducted under a limited *number scale*, defined as the number of transformed images generated from each original input.

For a number scale of N, each input is expanded into N transformed variants, requiring N times as many model evaluations per optimization step. Thus, this metric directly characterizes both the query count and the computational cost. Although sequential processing with gradient accumulation can reduce memory usage, the number of model evaluations remains the dominant factor in time overhead, particularly for large models. The number scale therefore serves as a standardized benchmark for assessing the efficiency of input transformation-based attacks, promoting fair and comparable evaluation across methods.

3.4. Block Stretch and Shrink Method

3.4.1. Constrained Random Segmentation Point Image Split Module

In input transformation-based attacks, image splitting requires careful selection of segmentation points to preserve semantic integrity. Previous methods like SIA and BSR (random points without constraints) often yield suboptimal segmentation. We propose a constrained random point selection mechanism that enforces a minimum distance between points (d_p) and from points to the edges of the image (d_b). This ensures well-distributed points, producing blocks with sufficient semantic content and effective local variations for adversarial perturbations.

Our method samples M pairs of segmentation points via constrained random sampling: candidate points are uniformly drawn within the image and accepted only if they satisfy the minimum distance between the pointsd_p (enforced separately among the width coordinates $\{p_{i,1}\}_{i=1}^{M}$ and height coordinates $\{p_{i,2}\}_{i=1}^{M}$) and the border margin d_b. These points define block boundaries along one dimension dim (, width or height), producing $M+1$ blocks B_i with lengths l_i computed from consecutive sorted points. The same set of points P is reused along the orthogonal dimension to avoid redundant sampling.

The transformation formula for this module, starting from the input image $x+\delta$ with dimensions $w \times h$, is as follows:

$$\begin{aligned} P = \Big\{ (p_{i,1}, p_{i,2}) \;\Big|\; & \\ & p_{i,1} \in [d_b, w-d_b],\ p_{i,2} \in [d_b, h-d_b], \\ & \min_{j\neq i} |p_{i,1} - p_{j,1}| \geq d_p,\ \min_{j\neq i} |p_{i,2} - p_{j,2}| \geq d_p, \\ & i = 1, \dots, M \Big\} \end{aligned} \tag{3}$$

$$\begin{cases} p_{0,\mathrm{w}}^{\mathrm{sort}} = 0,\ p_{1:M,\mathrm{w}}^{\mathrm{sort}} = \mathrm{sort}(\{p_{i,1}\}_{i=1}^{M}),\ p_{M+1,\mathrm{w}}^{\mathrm{sort}} = w \\ p_{0,\mathrm{h}}^{\mathrm{sort}} = 0,\ p_{1:M,\mathrm{h}}^{\mathrm{sort}} = \mathrm{sort}(\{p_{i,2}\}_{i=1}^{M}),\ p_{M+1,\mathrm{h}}^{\mathrm{sort}} = h \end{cases} \tag{4}$$

$$l_i = \left| p_{i,dim}^{\mathrm{sort}} - p_{i-1,dim}^{\mathrm{sort}} \right|,\quad i = 1, \dots, M+1 \tag{5}$$

$$B_i = \text{slice}\big(x+\delta,\ p^{\text{sort}}_{i-1,\text{dim}} : p^{\text{sort}}_{i,\text{dim}}, \\ \text{dim}\big), \quad i = 1, \dots, M+1 \tag{6}$$

where P is the set of M randomly sampled point pairs satisfying distance constraints, $p_{i,j}$ is the coordinate of the point j in the pair i along the dimension, l_i is the length of the block B_i, and B_i is the image block obtained by slicing the input image $x+\delta$ between the coordinates $p_{i,1}$ and $p_{i,2}$ along the specified dimension (height or width). The blocks B_i are then processed by the block lever stretch and shrink module to generate transformed versions of the image.

3.4.2. Block Lever Stretch and Shrink Module

The block lever stretch and shrink module create transformed images by introducing local stretch and shrink variations between blocks. The algorithm segments the input image into blocks B_i along a specified dimension (height or width) based on points selected by the constrained random point selection model. Each block B_i, with an initial length of l_i, is scaled to a target length l''_i using normalized random weights drawn from a uniform distribution. The stretch and shrink factors are sampled from a uniform distribution centered at 0.5, parameterized by the range-to-center ratio r, expressed as $s_i \sim \mathcal{U}\left(\frac{1-r/2}{2}, \frac{1+r/2}{2}\right)$, where the center 0.5 corresponds to the normalized proportion per-block of the total dimension.

The sampled values are then normalized to preserve the original overall length. Due to floating-point discrepancies, the final conversion may deviate from the original dimension length L. In such cases, the following adjustment strategy is applied: the lengths are rounded and clamped to a minimum of 1; if the sum deviates from the target length L, excess length is iteratively added to the largest blocks, or surplus is removed from the smallest blocks greater than 1, to preserve semantic content in larger regions while ensuring $\sum l''_i = L$. Stretch and shrink are implemented through bilinear interpolation, enabling smooth deformation across each block B_i.

The variables are defined as follows: range-to-center ratio r, number of blocks $M+1$, and image dimension along the specified axis L.

The transformation formula for this module, based on the input image $x+\delta$ with dimensions $w \times h$, is the following:

$$s_i \sim \mathcal{U}\left(\frac{1-r/2}{2}, \frac{1+r/2}{2}\right), \quad w_i = \frac{s_i}{\sum_{j=1}^{M+1} s_j} \tag{7}$$

$$l'_i = \text{round}(l_i \cdot w_i), \quad l''_i = \text{adj}(l'_i), \quad B'_i = \text{interp}(B_i, l''_i) \tag{8}$$

$$\tilde{x} = \text{concat}(B'_1, B'_2, \dots, B'_{M+1}) \tag{9}$$

where s_i is the stretch or shrink factor for block B_i, w_i is the normalized weight ensuring $\sum w_i = 1$, l_i is the original length of block B_i as defined in the point selection module, l'_i is the target length rounded to the nearest integer, $l''_i = \text{adj}(l'_i)$ is the adjusted integer length ensuring $\sum l''_i = L$ by incrementally adding to or subtracting from the largest or smallest lengths, B_i is the original block with length l_i along the specified dimension, $B'_i = \text{interp}(B_i, l''_i)$ is the trasfored block obtained via bilinear interpolation, and $\tilde{x} = \text{concat}(B'_1, B'_2, \dots, B'_{M+1})$ is the final transformed image for one dimension. This process is applied sequentially along both height and width dimensions to generate transformed versions of the image.

3.4.3. Transformation Results

The BSS method first selects one spatial dimension at random and applies the block lever stretch and shrink operation to each image block. The same procedure is then performed along the orthogonal dimension. This process, as shown in Figure 3, is repeated N times (where N is a hyperparameter controlling the number of transformations) in the perturbed image $x+\delta_t$ (with t denoting the current iteration step), resulting in an extended transformation set.

$$\mathcal{T}(x+\delta_t) = \{\tilde{x}_t^{(k)}\}_{k=1}^{N} \tag{10}$$

where each $\tilde{x}_t^{(k)}$ is a different stretch and shrink variant of $x+\delta_t$.

These transformed inputs $\mathcal{T}(x+\delta_t)$ are then fed into the surrogate model for adversarial optimization, updating the perturbation to δ_{t+1}.

Attack	Inception-v3 ⇒										
	ResNet-18	ResNet-101	ResNext-50	DenseNet-121	Inception-v3	Inception-v4	ViT	PiT	Visformer	Swin	Cait
BSR	87.9	64.4	70.5	90.7	99.6*	89.2	37.4	47.4	59.4	62	46.4
SIA	93	82.1	85.3	95	**99.8***	94.8	48	63.8	75.9	76.7	62
DeCoWA	87.7	69.2	74.7	90.6	99.3*	91.5	43.7	51.2	63.3	67.2	52.6
OPS(10,5,4)	88.5	71.6	75.2	89.8	99.2*	93.5	39.5	48.5	61.8	63.9	52.4
BSS	**94.6**	**89.7**	**90.7**	**97.1**	99.7*	**96.8**	**60.5**	**71.3**	**83.5**	**83**	**73.8**
Attack	DenseNet-121 ⇒										
	ResNet-18	ResNet-101	ResNext-50	DenseNet-121	Inception-v3	Inception-v4	ViT	PiT	Visformer	Swin	Cait
BSR	97.5	85.3	89.4	**100***	90.8	92	48.9	61.6	77.7	75.1	58.6
SIA	98.9	91.5	95.6	**100***	93	94.2	53.3	67	84.6	82.6	66.3
DeCoWA	97.4	85.9	88.3	99.9*	91	93.9	55.7	64.1	78.3	78.8	66.8
OPS(10,5,4)	98.3	90.7	92.2	99.7*	**96**	96.7	64.1	70	83	82.6	73.4
BSS	**99.2**	**95.9**	**96.8**	**100***	95.5	**97.2**	**72.1**	**82.8**	**92.3**	**90.7**	**82.8**

Table 1
Attack success rates (%) of different attack methods on six CNNs and five ViTs models, using Inception-v3 and DenseNet-121 as the white-box model and setting the number scale to 21. The best results are highlighted in bold, and * indicates white-box attacks.

3.5. Perturbation Generation

The perturbation generation stage performs optimization using the transformation set $\mathcal{T}(x+\delta_t)$ generated from the Block Stretch and Shrink method. This stage aims to produce an incremental perturbation that maximizes the average loss across all transformed variants, enhancing robustness and transferability.

Formally, the optimization objective is to maximize the following:

$$\frac{1}{N}\sum_{k=1}^{N} J(f(\tilde{x}_t^{(k)}), y) \tag{11}$$

To solve this objective, we employ the Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [5], which incorporates a momentum term to stabilize gradient updates and enhance transferability.

The algorithm initializes the perturbation and the accumulated gradient as $\delta_0 = 0$ and $g_0 = 0$.

Given the total number of iterations T, step size $\alpha = \epsilon/T$, and momentum decay factor μ, the gradient is calculated by averaging over all transformed samples:

$$\nabla_\delta J = \frac{1}{N}\sum_{k=1}^{N} \nabla_\delta J(f(\tilde{x}_t^{(k)}), y). \tag{12}$$

The accumulated momentum and perturbation are updated as follows:

$$g_t = \mu \cdot g_{t-1} + \frac{\nabla_\delta J}{\|\nabla_\delta J\|_1} \tag{13}$$

$$\delta_t = \text{clip}(\delta_{t-1} + \alpha \cdot \text{sign}(g_t), -\epsilon, \epsilon), \text{s.t.} \quad \|\delta_t\|_\infty \le \epsilon. \tag{14}$$

After T iterations, the final adversarial example x_{adv} is generated as:

$$x_{\text{adv}} = x + \delta_T. \tag{15}$$

4. Experiments

4.1. Experimental Setup

Hardware: All experiments were conducted on an NVIDIA RTX 3090 GPU (24 GB VRAM).

Dataset: We randomly selected one image per category from the ILSVRC-2012 validation set, resulting in a subset of 1000 images.

Parameter Settings: All experiments use the following hyperparameters random seed = 42; attack iterations (per image) $T = 10$; batch size $B = 1$. perturbation constraints are set to $\|\delta\|_\infty \le \epsilon$ with $\epsilon = 16/255$. The attack size per-step is $\alpha = \epsilon/T = 1.6/255$. We adopt the Momentum Iterative Fast Gradient Sign Method with momentum decay factor $\mu = 1$. Unless otherwise specified, experiments perform untargeted white-box attacks to generate perturbations and then evaluate black-box transferability by applying the same perturbations to held-out models.

Attack	Inception-v3 ⇒							
	Inception-v3 (ens3)	Inception-v3 (ens4)	Inception-ResNet-v2 (ens2)	HGD	NPR	NPR (→ResNet101)	NPR (→ViT)	AT
BSR	78.6	78.9	64.3	72.4	40.4	24	36.5	59.4
SIA	90.9	89.4	80.4	89.4	51.7	30.1	37.8	75.9
DeCoWA	85.5	85.1	76.1	79.3	48.6	29.9	40.2	63.3
OPS(10,5,4)	86.5	88.1	79.7	82.5	53.9	33.7	41.6	61.8
BSS	**95.2**	**93.6**	**88.8**	**93.2**	**61.7**	**42**	**41.8**	**83.5**

Table 2
Attack success rates (%) of different attack methods on eight image classifiers with different defenses, using Inception-v3 as the white-box model and setting the number scale to 21. The best results are highlighted in bold.

Attack	ResNet-18 ⇒										
	ResNet-18	ResNet-101	ResNext-50	DenseNet-121	Inception-v3	Inception-v4	ViT	PiT	Visformer	Swin	Cait
BaseLine	**100***	41.3	45.8	73.8	47.7	48.9	19.2	24.1	32.9	41.4	21.5
1D-BSS-RP	99.9*	87.5	90.4	98.9	93.1	93.8	57.8	65.2	81.6	83.3	70.1
1D-BSS	**100***	88.6	90.4	99.1	92.4	93.7	55.2	64.5	82.2	83.6	68.5
2D-BSS-RP	99.9*	86.9	89.5	98.1	94.1	93.7	63.8	68.8	83.7	83.6	73.9
BSS	**100***	**94.5**	**95.8**	**99.6**	**96.7**	**96.8**	**71**	**78.4**	**90.7**	**91.0**	**81.2**

Table 3
Attack success rates (%) of ablated BSS attack methods on six CNNs and five ViTs models, using ResNet-18 as the white-box model and setting the number scale to 30. The best results are highlighted in bold, and * indicates white-box attacks.

Models: We validate the impact of the number scale on input transformation-based methods. We first use ResNet-18 as a white-box model to generate adversarial perturbations. To evaluate generalization, we further apply the method to three architectures: ResNet-18 [10], DenseNet-121 [13], and Inception-v3 [37]. To validate the transferability of black-box attacks, we transfer adversarial images to CNNs models (, ResNet-101 [10] , ResNext-50 [49] and Inception-v4 [36]) and test them on vision Transformer models (, ViT [7], PiT [11], Visformer [3], Swin [23], and CaiT [40]).

Subsequently, to validate the robustness of adversarial perturbations, we used Inception-v3 as a white-box model, utilizing three ensemble adversarial training models (, ens3_adv_Inception-v3, ens4_adv_Inception-v3, and ens2_adv_Inception-ResNet)[41] and three adversarial training methods: Adversarial Training (AT)[27], Gradient-based Defense (HGD)[17], and Neural Representation Purifier (NRP) [28].

Baselines: To validate the effectiveness of BSS, we compare it against two advanced input transformation operator methods (, BSR and DeCoWA) and two input transformation combination methods (, SIA and OPS). To ensure fairness, all input transformations were integrated into MI-FGSM using identical parameters.

4.2. Unified Number Scale Experiments

As discussed in Section 3.3, input transformation-based methods should be evaluated on identical number scales (N) to ensure a fair and meaningful comparison. We therefore conducted unified number scale experiments to examine how varying N affects attack performance between different transformation-based methods. The number scale was varied among { 10, 17, 21, 26, 37, 101, 401 }, and all attacks were implemented using MI-FGSM under identical hyperparameter settings.

The average attack success rates (%) on six CNNs and five ViT models are summarized in Figure 4. The results show that the success rates of all methods consistently increase with larger number scales, indicating that expanding N enhances input diversity and allows perturbations to better exploit model vulnerabilities. This observation highlights the importance of standardizing N when evaluating transformation-based methods, as variations in scale can inadvertently influence outcomes and affect the comparability of results.

Across all number scales, our proposed BSS method generally achieves high success rates. The advantage is particularly pronounced on smaller number scales ($N = 10, 17, 21, 26, 37$), where BSS significantly outperforms existing state-of-the-art methods. Even on larger number scales ($N = 101, 401$), BSS generally has superior transferability compared to other methods.

4.3. Model Attack Experiments

To evaluate the transferability of adversarial perturbations generated from different white-box models across various target black-box models, we conducted experiments using Inception-V3 and DenseNet-121 as white-box models. These complex architectures were chosen to assess performance under realistic computational constraints. We fixed the number of transformations at $N = 21$ to balance performance and computational efficiency.

All attacks were implemented using MI-FGSM under identical hyperparameter settings. The results, summarized in Table 1, demonstrate that the proposed BSS method substantially achieves the highest transferability compared to SOTA

input transformation-based attacks, including OPS, SIA, BSR, and DeCoWA. These results further validate the effectiveness of the BSS method in enhancing adversarial transferability when employing diverse white-box models under an unified number of transformation conditions.

4.4. Defense Experiments

Following the attack transferability experiments, we further evaluate the effectiveness of the proposed BSS method against multiple defense mechanisms. The evaluation plan is as described in Section 4.1.

All adversarial examples were generated on the Inception-V3 model using MI-FGSM as the base attack, with the number of transformations fixed at $N = 21$. The results are summarized in Table 2.

The proposed BSS method achieves the highest average success rate in most defense models, demonstrating strong generalization capability and robustness against diverse adversarial training strategies.

4.5. Ablation Experiments

We conducted ablation experiments on ResNet-18 with consistent hyperparameter settings and the number of transformations fixed at $N = 30$. Furthermore, we believe that the performance transferred to ViTs model is more important.

First, we performed parameter selection to determine the optimal values for the four hyperparameters defined in Section 3.4: the minimum boundary distance d_b, the distance between points d_p, the range-to-center ratio r, and the number of segmentation points M.

We varied the boundary distance d_b while fixing $d_p = 40$, $r = 1$, and $M = 2$, and based on the average attack success rate on six CNNs and five ViT models, the optimal value was determined as $d_b = 35$, as shown in Figure 5a.

Then, we analyzed the effect of distance between points d_p with $d_b = 35$, $r = 1$, and $M = 2$, where $d_p = 40$ achieved the best overall transfer performance, as shown in Figure 5b.

Next, we examine the influence of the range-to-center ratio r while fixing $d_b = 35$, $d_p = 40$, and $M = 2$. A ratio of $r = 1$ yielded the most balanced and stable attack performance between models, as shown in Figure 5c.

Finally, we evaluated the number of segmentation points M with fixed $d_b = 35$, $d_p = 40$, and $r = 1$. The optimal value was $M = 2$, which was also used for the other methods in the above experiments, as shown in Figure 5d.

Additionally, we assessed the contributions of the constrained point selection mechanism and single- versus dual-dimension processing. The results are reported in Table 3.

We evaluate four experimental variants:

- **1D-BSS-RP**: random segmentation points with single-dimension stretch and shrink.
- **1D-BSS**: constrained point selection combined with single-dimension stretch and shrink.
- **2D-BSS-RP**: random segmentation points with dual-dimension stretch and shrink.
- **BSS**: constrained point selection with dual-dimensional stretch and shrink (the full pipeline).

The results show that constrained point selection has minimal effect in the single-dimension case (1D-BSS vs. 1D-BSS-RP), but BSS significantly outperforms the 1D variants and 2D-BSS-RP. This shows that constrained segmentation and dual-dimension stretch and shrink work synergistically, improving the transferability of adversarial examples.

5. Summary

Generating high-transferability adversarial perturbations requires enriching diverse attention heatmaps while preserving the global semantic integrity of transformed inputs. We propose Block Stretch and Shrink (BSS), an input transformation-based attack using block level stretch and shrink to diversify attention and maintain semantic coherence. Compared with various SOTA methods, our approach substantially achieves the best transferability.

We categorize input transformation-based methods into two subdomains: individual operator design and operator combinations.

To ensure a fair and meaningful comparison, we introduce a standardized evaluation framework with a unified number scale N to ensure consistent input diversity between methods.

References

[1] Carlini, N., Wagner, D.A., 2016. Towards evaluating the robustness of neural networks. 2017 IEEE Symposium on Security and Privacy (SP) , 39–57URL: https://api.semanticscholar.org/CorpusID:2893830.

[2] Chen, J., Chen, H., Chen, K., Zhang, Y., Zou, Z., Shi, Z., 2025. Diffusion models for imperceptible and transferable adversarial attack. IEEE Transactions on Pattern Analysis and Machine Intelligence 47, 961–977. doi:10.1109/TPAMI.2024.3480519.

[3] Chen, Z., Xie, L., Niu, J., Liu, X., Wei, L., Tian, Q., 2021. Visformer: The vision-friendly transformer. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) , 569–578URL: https://api.semanticscholar.org/CorpusID:233394097.

[4] Cohen, J.M., Rosenfeld, E., Kolter, J.Z., 2019. Certified adversarial robustness via randomized smoothing. ArXiv abs/1902.02918. URL: https://api.semanticscholar.org/CorpusID:59842968.

[5] Dong, Y., Liao, F., Pang, T., Hu, X., Zhu, J., 2017. Discovering adversarial examples with momentum. ArXiv abs/1710.06081. URL: https://api.semanticscholar.org/CorpusID:195346264.

[6] Dong, Y., Pang, T., Su, H., Zhu, J., 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 4307–4316URL: https://api.semanticscholar.org/CorpusID:102350868.

[7] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. ArXiv abs/2010.11929. URL: https://api.semanticscholar.org/CorpusID:225039882.

[8] Goodfellow, I.J., Shlens, J., Szegedy, C., 2014. Explaining and harnessing adversarial examples. CoRR abs/1412.6572. URL: https://api.semanticscholar.org/CorpusID:6706414.

[9] Guo, Y., Liu, W., Xu, Q., Zheng, S., Huang, S., Zang, Y., Shen, S., Wen, C., Wang, C., 2025. Boosting adversarial transferability through augmentation in hypothesis space, in: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 19175–19185. doi:10.1109/CVPR52734.2025.01786.

[10] He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 770–778URL: https://api.semanticscholar.org/CorpusID:206594692.

[11] Heo, B., Yun, S., Han, D., Chun, S., Choe, J., Oh, S.J., 2021. Rethinking spatial dimensions of vision transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) , 11916–11925URL: https://api.semanticscholar.org/CorpusID:232417451.

[12] Hernandez-Garcia, A., König, P., 2018. Data augmentation instead of explicit regularization. ArXiv abs/1806.03852. URL: https://api.semanticscholar.org/CorpusID:51894279.

[13] Huang, G., Liu, Z., Weinberger, K.Q., 2016. Densely connected convolutional networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 2261–2269URL: https://api.semanticscholar.org/CorpusID:9433631.

[14] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Communications of the ACM 60, 84 – 90. URL: https://api.semanticscholar.org/CorpusID:195908774.

[15] Kurakin, A., Goodfellow, I.J., Bengio, S., 2016. Adversarial examples in the physical world. ArXiv abs/1607.02533. URL: https://api.semanticscholar.org/CorpusID:1257772.

[16] Li, Z., Wang, W., Li, J., Chen, K., Zhang, S., 2024. Foolmix: Strengthen the transferability of adversarial examples by dual-blending and direction update strategy. IEEE Transactions on Information Forensics and Security 19, 5286–5300. URL: https://api.semanticscholar.org/CorpusID:269395288.

[17] Liao, F., Liang, M., Dong, Y., Pang, T., Zhu, J., Hu, X., 2017. Defense against adversarial attacks using high-level representation guided denoiser. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition , 1778–1787URL: https://api.semanticscholar.org/CorpusID:604742.

[18] Lin, J., Song, C., He, K., Wang, L., Hopcroft, J.E., 2019. Nesterov accelerated gradient and scale invariance for adversarial attacks. arXiv: Learning URL: https://api.semanticscholar.org/CorpusID:202750057.

[19] Lin, Q., Luo, C., Niu, Z., He, X., Xie, W., Hou, Y., Shen, L., Song, S., 2024. Boosting adversarial transferability across model genus by deformation-constrained warping, in: AAAI Conference on Artificial Intelligence. URL: https://api.semanticscholar.org/CorpusID:267500078.

[20] Liu, J., Wang, Z., Zhang, Z., Huang, C., Liang, S., Tang, Y., Xu, C., 2025a. Harnessing the computation redundancy in vits to boost adversarial transferability. URL: https://api.semanticscholar.org/CorpusID:277786663.

[21] Liu, R., Wu, H., Zhang, J., Cheng, X., Luo, X., Ma, B., Wang, J., 2025b. Pixel2Feature attack (P2FA): Rethinking the perturbed space to enhance adversarial transferability, in: Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., Zhu, J. (Eds.), Proceedings of the 42nd International Conference on Machine Learning, PMLR. pp. 39853–39870. URL: https://proceedings.mlr.press/v267/liu25cd.html.

[22] Liu, Y., Chen, X., Liu, C., Song, D.X., 2016. Delving into transferable adversarial examples and black-box attacks. ArXiv abs/1611.02770. URL: https://api.semanticscholar.org/CorpusID:17707860.

[23] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) , 9992–10002URL: https://api.semanticscholar.org/CorpusID:232352874.

[24] Lundberg, S.M., Lee, S.I., 2017. A unified approach to interpreting model predictions, in: Neural Information Processing Systems. URL: https://api.semanticscholar.org/CorpusID:21889700.

[25] Ma, X., Niu, Y., Gu, L., Wang, Y., Zhao, Y., Bailey, J., Lu, F., 2019. Understanding adversarial attacks on deep learning based medical image analysis systems. ArXiv abs/1907.10456. URL: https://api.semanticscholar.org/CorpusID:198229815.

[26] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A., 2017a. Towards deep learning models resistant to adversarial attacks. ArXiv abs/1706.06083. URL: https://api.semanticscholar.org/CorpusID:3488815.

[27] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A., 2017b. Towards deep learning models resistant to adversarial attacks. ArXiv abs/1706.06083. URL: https://api.semanticscholar.org/CorpusID:3488815.

[28] Naseer, M., Khan, S.H., Hayat, M., Khan, F.S., Porikli, F.M., 2020. A self-supervised approach for adversarial robustness. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 259–268URL: https://api.semanticscholar.org/CorpusID:219559020.

[29] Papernot, N., Mcdaniel, P., Goodfellow, I.J., Jha, S., Celik, Z.B., Swami, A., 2016. Practical black-box attacks against machine learning. Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security URL: https://api.semanticscholar.org/CorpusID:1090603.

[30] Papernot, N., Mcdaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A., 2015. The limitations of deep learning in adversarial settings. 2016 IEEE European Symposium on Security and Privacy (EuroS&P) , 372–387URL: https://api.semanticscholar.org/CorpusID:7004303.

[31] Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A., 2015. You only look once: Unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 779–788URL: https://api.semanticscholar.org/CorpusID:206594738.

[32] Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., Batra, D., 2016. Grad-cam: Visual explanations from deep networks via gradient-based localization. International Journal of Computer Vision 128, 336 – 359. URL: https://api.semanticscholar.org/CorpusID:15019293.

[33] Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K., 2016. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security URL: https://api.semanticscholar.org/CorpusID:207241700.

[34] Shelhamer, E., Long, J., Darrell, T., 2014. Fully convolutional networks for semantic segmentation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 3431–3440URL: https://api.semanticscholar.org/CorpusID:1629541.

[35] Shorten, C., Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. Journal of Big Data 6, 1–48. URL: https://api.semanticscholar.org/CorpusID:195811894.

[36] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A., 2016. Inception-v4, inception-resnet and the impact of residual connections on learning. ArXiv abs/1602.07261. URL: https://api.semanticscholar.org/CorpusID:1023605.

[37] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2015. Rethinking the inception architecture for computer vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 2818–2826URL: https://api.semanticscholar.org/CorpusID:206593880.

[38] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R., 2013. Intriguing properties of neural networks. CoRR abs/1312.6199. URL: https://api.semanticscholar.org/CorpusID:604334.

[39] Tang, B., Wang, Z., Bin, Y., Dou, Q., Yang, Y., Shen, H.T., 2024. Ensemble diversity facilitates adversarial transferability, in: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 24377–24386. doi:10.1109/CVPR52733.2024.02301.

[40] Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., J'egou, H., 2021. Going deeper with image transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) , 32–42URL: https://api.semanticscholar.org/CorpusID:232428161.

[41] Tramèr, F., Kurakin, A., Papernot, N., Boneh, D., Mcdaniel, P., 2017. Ensemble adversarial training: Attacks and defenses. ArXiv abs/1705.07204. URL: https://api.semanticscholar.org/CorpusID:21946795.

[42] Wang, K., He, X., Wang, W., Wang, X., 2023a. Boosting adversarial transferability by block shuffle and rotation. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 24336–24346URL: https://api.semanticscholar.org/CorpusID:261048885.

[43] Wang, X., He, X., Wang, J., He, K., 2021a. Admix: Enhancing the transferability of adversarial attacks. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) , 16138–16147URL: https://api.semanticscholar.org/CorpusID:231741167.

[44] Wang, X., Yin, Z., Zhang, Z., Wang, K., Ge, Z., Luo, Y., 2023b. Transferattack https://github.com/trustworthy-ai-group/transferattack. GitHub repository.

[45] Wang, X., Zhang, Z., Zhang, J., 2023c. Structure invariant transformation for better adversarial transferability. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) , 4584–4596URL: https://api.semanticscholar.org/CorpusID:262827690.

[46] Wang, Y., Liu, J., Chang, X., Wang, J., Rodr'iguez, R.J., 2021b. Di-aa: An interpretable white-box attack for fooling deep neural networks. ArXiv abs/2110.07305. URL: https://api.semanticscholar.org/CorpusID:238857225.

[47] Wong, E., Rice, L., Kolter, J.Z., 2020. Fast is better than free: Revisiting adversarial training. ArXiv abs/2001.03994. URL: https://api.semanticscholar.org/CorpusID:210164926.

[48] Xie, C., Zhang, Z., Wang, J., Zhou, Y., Ren, Z., Yuille, A.L., 2018. Improving transferability of adversarial examples with input diversity. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 2725–2734URL: https://api.semanticscholar.org/CorpusID:3972825.

[49] Xie, S., Girshick, R.B., Dollár, P., Tu, Z., He, K., 2016. Aggregated residual transformations for deep neural networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 5987–5995URL: https://api.semanticscholar.org/CorpusID:8485068.

[50] Xu, W., Evans, D., Qi, Y., 2017. Feature squeezing: Detecting adversarial examples in deep neural networks. ArXiv abs/1704.01155. URL: https://api.semanticscholar.org/CorpusID:3851184.

[51] Yang, J., Boloor, A., Chakrabarti, A., Zhang, X., Vorobeychik, Y., 2020. Finding physical adversarial examples for autonomous driving with fast and differentiable image compositing. URL: https://api.semanticscholar.org/CorpusID:224705298.

[52] Yuan, Z., Zhang, J., Shan, S., 2021. Adaptive image transformations for transfer-based adversarial attack. ArXiv abs/2111.13844. URL: https://api.semanticscholar.org/CorpusID:244714851.

[53] Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., Torralba, A., 2015. Learning deep features for discriminative localization. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 2921–2929URL: https://api.semanticscholar.org/CorpusID:6789015.

[54] Zhu, R., Zhang, Z., Liang, S., Liu, Z., Xu, C., 2024. Learning to transform dynamically for better adversarial transferability. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 24273–24283URL: https://api.semanticscholar.org/CorpusID:269983246.

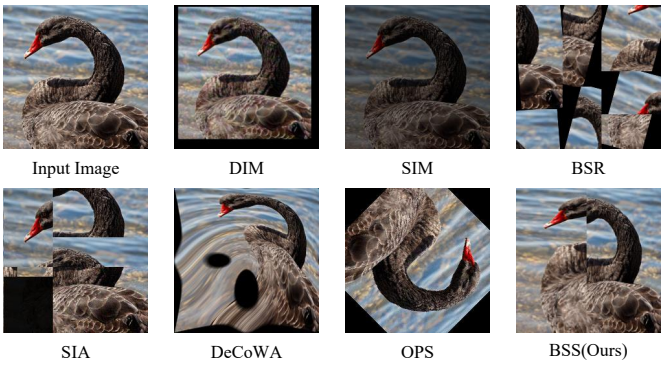


Figure 1: The input image and its transformed versions generated by different methods. Except for BSS and OPS, existing advanced methods (, BSR, SIA) fail to maintain global semantic integrity. In contrast, traditional methods (, SIM, DIM) cannot produce rich feature representations.

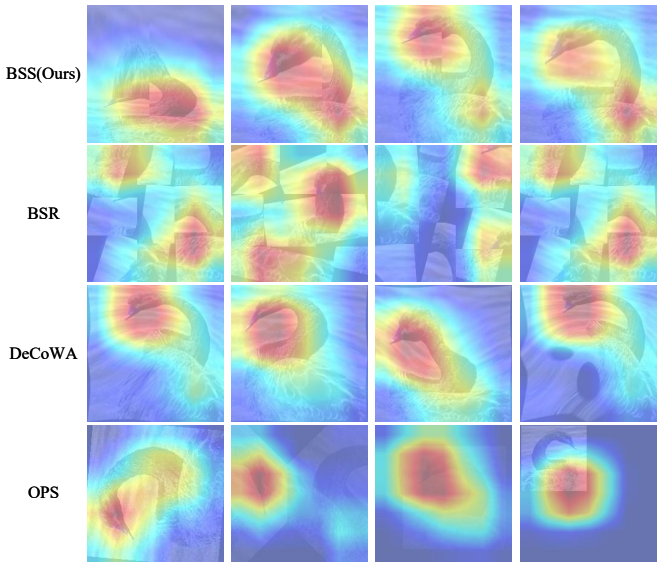


Figure 2: Attention heatmaps of the transformed images generated by BSS, OPS, DeCoWA, and BSR, computed on ResNet-18.

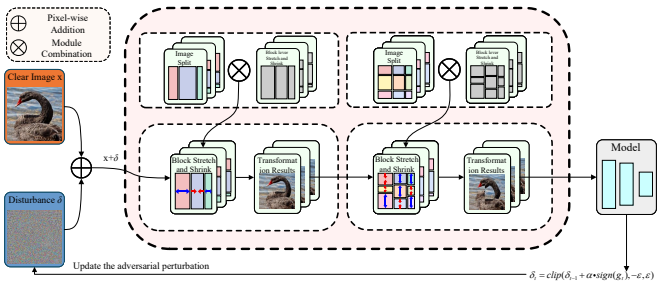
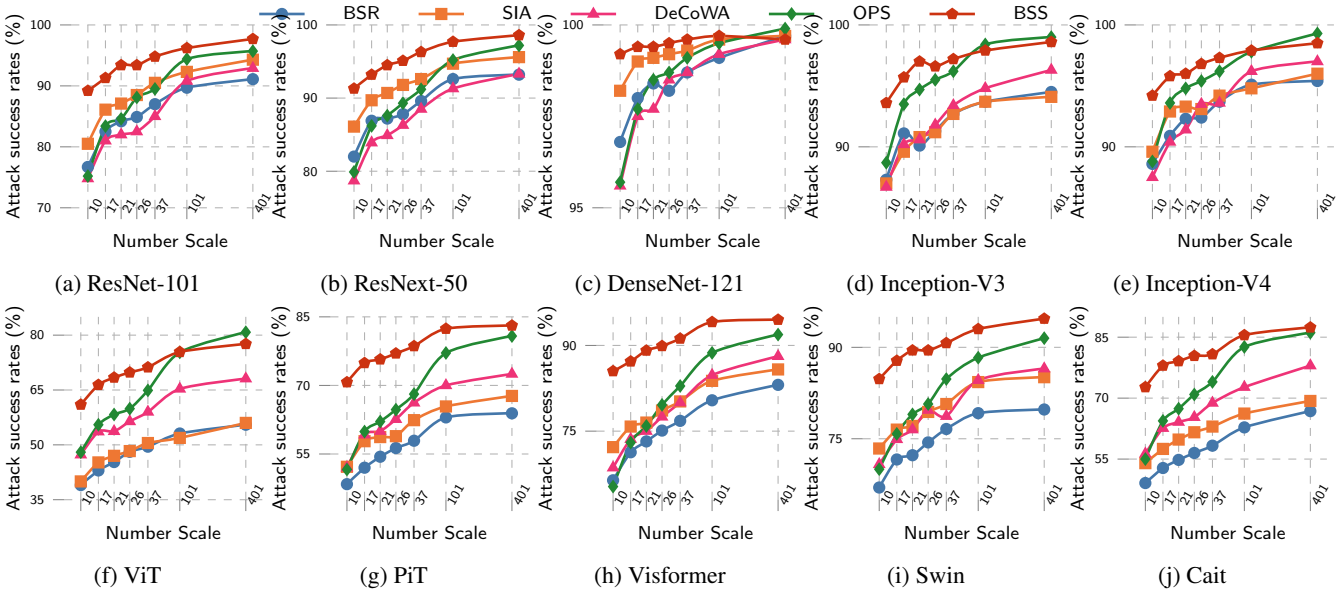


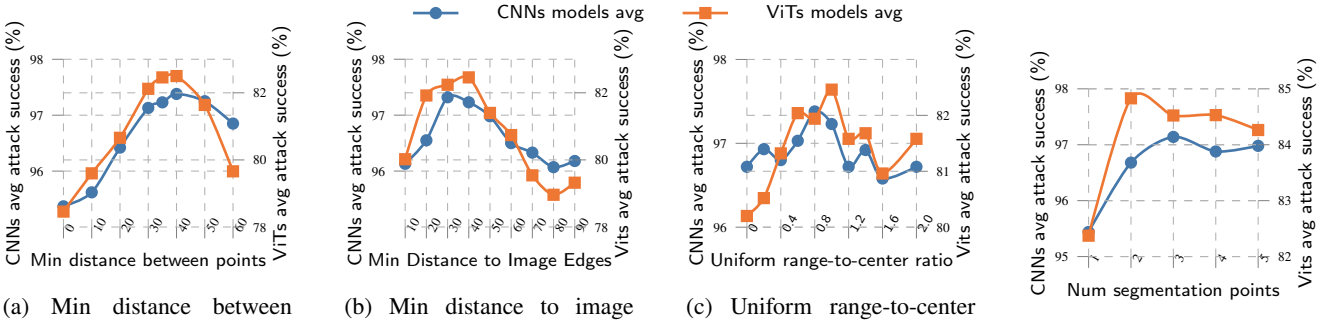
Figure 3: This figure shows the specific principle of Block Stretch and Shrink. After adding perturbation δ to the input image x, the image split and stretch and shrink modules operate along two dimensions to produce a transformed input, which is then fed into the model to generate adversarial perturbations.



(a) ResNet-101 (b) ResNext-50 (c) DenseNet-121 (d) Inception-V3 (e) Inception-V4

(f) ViT (g) PiT (h) Visformer (i) Swin (j) Cait

Figure 4: Attack success rates (%) of different attack methods in different number scale on five CNNs and five ViTs models, using ResNet-18 as the white-box model, with input transformation methods BSR, SIA, DeCoWA, OPS, and BSS.



(a) Min distance between points (b) Min distance to image edges (c) Uniform range-to-center ratio (d) Num segmentation points

Figure 5: Average attack success rates (%) on six CNNs and five ViTs models under different parameter configurations, using ResNet-18 as the white-box model and setting the number scale to 30.