RESEARCH-ARTICLE

I-C Attack: In-place and Cross-pixel Augmentations for Highly Transferable Transformation-based Attacks

JIAMING LIANG, University of Macau, Taipa, Macao

CHIMAN PUN, University of Macau, Taipa, Macao

Published: 27 October 2025

MM '25: The 33rd ACM International Conference on Multimedia
October 27 - 31, 2025
Dublin, Ireland

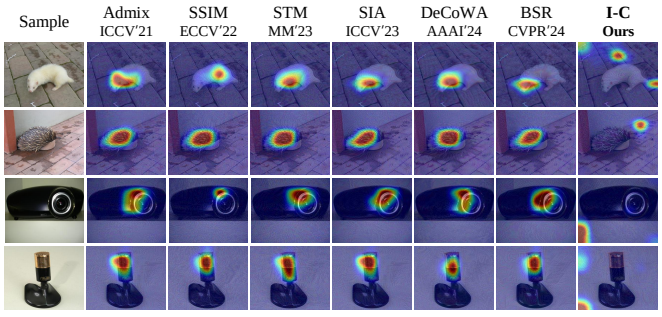
Conference Sponsors:
SIGMM

https://doi.org/10.1145/3746027.3754799
ISBN: 9798400720352

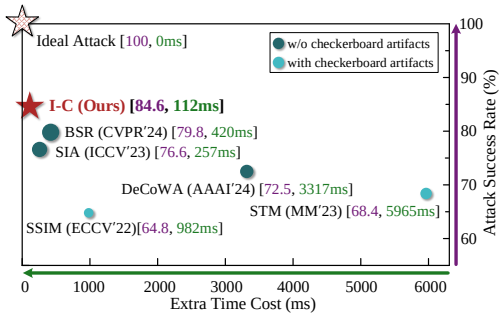
I-C Attack: In-place and Cross-pixel Augmentations for Highly Transferable Transformation-based Attacks

Jiaming Liang
University of Macau
Faculty of Science and Technology
Macau, China
chinaliangjm@gmail.com

Chi-Man Pun*
University of Macau
Faculty of Science and Technology
Macau, China
cmpun@um.edu.mo



(a) Grad-CAM Visualization



(b) Overview

Figure 1: (a) Grad-CAM visualizations of adversarial examples targeting hard cases. (b) Performance overview. The x-axis represents the average extra time cost required per sample compared to MI-FGSM on an NVIDIA RTX 3090 GPU.

Abstract

The efficiency and high transferability of transformation-based adversarial attacks (TAAs) make them a promising tool for robustness analysis. Despite the improvements in transferability brought by various image transformations, their underlying causes remain unclear, and there is still room for further improvement. We find that with attention-based models as surrogate models, adversarial examples generated by TAAs with relatively lower transferability tend to exhibit **checkerboard artifacts**, whereas those with higher transferability do not. This motivates us to explore the relationship between transferability and checkerboard artifacts. We confirm that checkerboard artifacts originate from the patching operation in attention-based surrogate models. Checkerboard artifacts vanish under the condition that spatial transformations are applied and gradients are calculated with respect to perturbations. Based on whether checkerboard artifacts are eliminated, we categorize model augmentations into **cross-pixel augmentations** and **in-place augmentations**. The former promotes interactions between pixels, breaks patch isolation, and thereby improves transferability while removing artifacts. The latter in-place augment the diversity of parameter features, enhancing transferability but failing to break isolation and remove artifacts. They constitute two distinct ways toward enhancing transferability. Integrating them enables higher transferability. Therefore, we propose an attack design paradigm to fully leverage both augmentations. To verify this paradigm, we design a basic **In-place and Cross-pixel Attack (I-C Attack)** with simple transformations. Extensive experiments demonstrate that, despite its simplicity, I-C attack can achieve much higher transferability while maintaining low computational cost. The code is available at https://github.com/chinaliangjiaming/I-C-Attack.git.

CCS Concepts

• **Computing methodologies** → **Computer vision**.

Keywords

Machine Learning, Neural Networks, Artificial Intelligence Security, Adversarial Attacks

ACM Reference Format:

Jiaming Liang and Chi-Man Pun. 2025. I-C Attack: In-place and Cross-pixel Augmentations for Highly Transferable Transformation-based Attacks. In *Proceedings of Proceedings of the 33rd ACM International Conference on Multimedia (MM '25).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746027.3754799

*Corresponding author.

MM '25, Dublin, Ireland.

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3754799

1 Introduction

Deep neural networks (DNNs) are widely used in safety-critical domains due to their exceptional pattern recognition capabilities [17, 21, 45, 60, 77]. Unfortunately, adversaries can deceive models with

imperceptible perturbations, known as adversarial attacks [56]. In white-box settings, iterative gradient-based attacks achieve satisfactory performance [30]. In contrast, under black-box settings where the adversary has limited or no knowledge of the target model, plain attacks perform unsatisfactorily. To improve performance, relying on the principle that the same adversarial example could be effective against multiple models, transfer-based strategies [28, 31, 66, 69] have been proposed. Transformation-based attacks, known for high transferability, low complexity, and plug-and-play flexibility, form a promising branch of transfer-based attacks.

Transformation-based attacks enhance transferability by transforming images before inputting them into surrogate models. As shown in Figure 1(b), while existing methods have made notable progress in black-box settings, significant room for improvement remains. In addition, pioneering work [66] suggests that image transformations effectively augment surrogate models, enriching perturbation features and boosting transferability. However, the detailed mechanism underlying model augmentation and transferability remains unclear. Therefore, this study attempts to explore the following problem: What is the intrinsic mechanism between model augmentations and adversarial transferability?

We find that some transformation-based attacks, when using attention-based models as surrogates, generate adversarial examples with checkerboard artifacts and generally exhibit low transferability. In contrast, other attacks do not produce such artifacts and achieve relatively higher transferability. This prompts us to consider: *Does the presence or absence of checkerboard artifacts indicate some inherent characteristic of different model augmentations? Is this characteristic closely related to transferability?*

To investigate this question, we conduct an in-depth review of transformation-based attacks and find that there exist two distinct differentiation modes: (a) Differentiate with respect to the perturbation of the preceding iteration; (b) Differentiate with respect to the transformed image of the current iteration. With attention-based models as surrogates, all attacks exhibit checkerboard artifacts under differentiation mode (b), whereas under mode (a), artifacts disappear only in attacks employing spatial transformations. This suggests a potential relation between checkerboard artifacts and differentiation modes. Theoretical analysis demonstrates that mode (a) includes additional inverse transformations[1] compared to mode (b). When spatial transformations are applied, mode (a) allows pixel interactions that merge perturbation information, breaking the patch-wise isolation from the attention mechanism and eliminating checkerboard artifacts. In contrast, mode (b) is equivalent to a pixel-wise constant addition transformation, regardless of the applied transformation, and thus cannot eliminate checkerboard artifacts.

Therefore, checkerboard artifacts serve as an indicator of whether perturbation information from interacting pixels across different positions is provided. The presence of checkerboard artifacts suggests a lack of such perturbations, resulting in low transferability. Based on whether checkerboard artifacts can be eliminated, we can categorize model augmentations that enhance transferability into two types: **in-place augmentations** and **cross-pixel augmentations**. In-place augmentations enhance transferability by augmenting the surrogates through non-pixel interactions, and are unable to eliminate artifacts. Cross-pixel augmentations augment the surrogates based on pixel interactions, with the core idea being the integration of perturbation information from different positions to improve transferability. Thereby, the transferability of transformation-based attacks stems from these two augmentations.

Building on this mechanism, we propose a design paradigm to enhance transferability by exploiting in-place and cross-pixel augmentations. In-place augmentations focus on designing auxiliary functions for each pixel, whereas cross-pixel augmentations center on enabling pixel interactions, simplified as pixel relocation. To illustrate the performance baseline achievable by this design paradigm, we designed a basic **In-place and Cross-pixel Attack (I-C Attack)** using three simple transformations. The employed noise addition transformation provides in-place augmentations, while bilinear integration and block shuffle offer local and global interactions, respectively. Extensive experiments demonstrate that I-C attack not only has low computational complexity, but also outperforms existing methods in various scenarios.

Our main contributions are summarized as follows:

- To the best of our knowledge, this is the first study that identifies checkerboard artifacts in transformation-based attacks and uncovers their origin.
- Based on whether checkerboard artifacts are eliminated, we refine model augmentations into cross-pixel and in-place augmentations, and reveal their connection to transferability.
- We propose a design paradigm that integrates in-place and cross-pixel augmentations for higher transferability.
- We develop the I-C attack as the baseline for the paradigm. Extensive experiments demonstrate that I-C attack achieves state-of-the-art while maintaining low computational cost.

2 Related Work

2.1 Adversarial Attacks

2.1.1 Overview. Adversarial attacks aim to craft adversarial examples by adding imperceptible perturbations to clean samples to fool models. In a white-box setting, adversaries can derive precise perturbations by differentiating target model's output loss with respect to the input sample, allowing flawless attacks. However, in a black-box setting, adversaries cannot access precise gradients and have to rely on other strategies to carry out attacks.

Existing black-box attacks can be categorized into query-based and transfer-based strategies. Query-based strategies assume adversaries have access to target model's hard outputs (class labels) or soft outputs (logits), leading to decision-based query attacks (DQAs) [8, 9, 12, 29, 33, 44, 62] and score-based query attacks (SQAs) [1, 4, 13, 29, 43, 61]. By iteratively modifying the perturbations and observing the changes in the output, the adversaries update the perturbations accordingly. However, real-world deployed applications often impose query limits, and access to hard or soft outputs may not always be guaranteed. In contrast, transfer-based strategies offer greater flexibility.

Transfer-based attacks rely on the transferability of adversarial examples, where the same adversarial examples may be effective against multiple models. These attacks generate adversarial

[1]In this paper, forward propagation through the transformation module is called *transformation*, while backpropagation through it is termed *inverse transformation*.

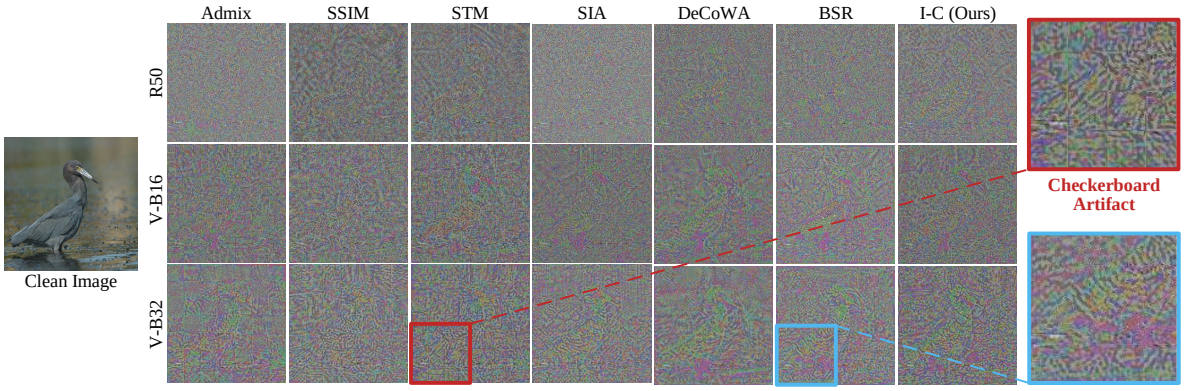


Figure 2: Illustrative example of checkerboard artifacts in perturbations. These perturbations are generated based on R50, V-B16, and V-B32, respectively. Each attack is combined with MI-FGSM. Every perturbation is min-max normalized.

examples on surrogate models to attack target models. Gradient-based strategies [22, 47, 63] enhance transferability by rectifying perturbations or altering their accumulation patterns. Model-related strategies [49, 65, 69, 74] enhance transferability by exploring similarities and differences between models. Ensemble-based strategies [11, 28] integrate multiple models as surrogates to bridge differences between models. Objective-optimized strategies [31, 68, 73, 75, 78] improve transferability by adjusting the objective function to prevent adversarial examples from overfitting surrogate models. Generation-based attacks [6, 32, 34] leverage generative structures such as GANs [24] and diffusion models [55] to learn the transferable patterns of perturbations. Transformation-based strategies [23, 36, 42, 64, 66, 67] improve transferability by augmenting samples before feeding them into surrogate models.

2.1.2 Transformation-based Attacks. Transformation-based strategies can be used independently or combined with other strategies to form stronger attacks. They enable plug-and-play generation of highly transferable adversarial examples, making them promising. Existing methods explored various possible image transformations. DIM [70], TIM [19], and SIM [35] enhance transferability by using resizing, shifting, and scaling for image augmentation, respectively. Admix [66] employs mixup for image augmentation. SSIM [42] utilizes discrete cosine transform to augment images in the frequency domain. STM [23] applies a stylization neural network. SIA [67] integrates multiple transformations, including shifting, flipping, rotation, scaling, noise addition, and blurring. DeCoWA [36] distorts images for augmentation. BSR [64] uses block shuffle and rotation for image augmentation. Unfortunately, the underlying causes of transformation-based transferability remain unclear, leading to a lack of design paradigm for guiding the development of more effective attacks.

2.2 Explanations of Transformation-based Transferability

Preliminary explanations for transformation-based transferability have been proposed. Jiadong Lin et al. proposed the concept of model augmentation [35]. They regard surrogate model together with image transformation with random parameters as a new unified model. Varying the transformation parameters is equivalent to changing the surrogate model, preventing the generated perturbations from fitting a single model and thus enhancing transferability. However, they provide limited discussion on how different transformations contribute to transferability enhancement. In addition, their definition of model augmentation relies on loss-preserving transformations [35], a prior assumption that empirical findings suggest may warrant further examination. Kunyu Wang et al. argue that transformation-based transferability stems from the consistency of attention heatmaps across different models for the same adversarial example. [64] However, significant differences in attention heatmaps across models for the same adversarial example do not necessarily impede transferability. Therefore, transformation-based transferability remains an open question. This work aims to further explore the fundamental causes of transformation-based transferability and how to refine its categorization.

3 Proposed Method

3.1 Preliminaries

3.1.1 Notation Definition. Given a sample-label pair $(\boldsymbol{x}, y) \in (\mathcal{X}, \mathcal{Y})$ and a target model $M_T \in \mathcal{M}$, the adversary aims to find an attack $A \in \mathcal{A}$ with perturbation budget ϵ that satisfies

$$\arg\max M_T(A(\boldsymbol{x}); \boldsymbol{\theta}_{M_T}) \neq y,\ s.t. ||\tilde{\boldsymbol{x}} - \boldsymbol{x}||_p \leq \epsilon, \tag{1}$$

where $A(\boldsymbol{x})$ outputs adversarial example $\tilde{\boldsymbol{x}}$, and $\boldsymbol{\theta}_{\boldsymbol{M_T}}$ indicates parameters of target model. $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{M}$, and $\mathcal{A}$ represent the spaces of samples, labels, models, and attack algorithms, respectively. Consistent with previous work, this paper focuses on the perturbation budget ϵ of the L_∞ norm. Without incorporating advanced strategies, the basic iterative I-FGSM follows the recurrence equation

$$\tilde{\boldsymbol{x}}_{t+1} = Clip(\tilde{\boldsymbol{x}}_t + \alpha \cdot \frac{\partial J(M_S(\tilde{\boldsymbol{x}}_t; \boldsymbol{\theta}_{M_S}), y)}{\partial \tilde{\boldsymbol{x}}_t}),\ 0 \leq t \leq T-1, \tag{2}$$

where J is the loss function, M_S denotes the surrogate model, and $\tilde{\boldsymbol{x}}_{\boldsymbol{t}}$ is the adversarial example at iteration t. T is the number of iterations, and $\alpha = \epsilon/T$ is the step size per iteration. The $Clip$ function truncates the output within the range $(0, 1)$.

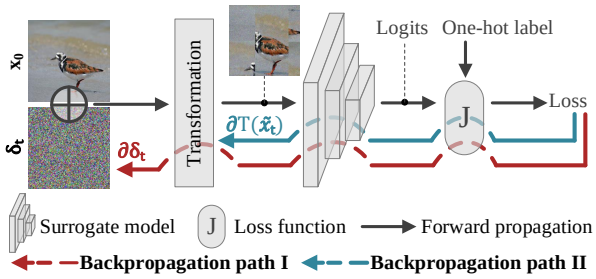


Figure 3: Illustration of two differentiation modes.

3.1.2 Framework of Transformation-based Attacks. Transformation-based attacks apply transformations N times to $\tilde{x}_t$ with varying parameters θ_T in each iteration, averaging the resulting N gradients to yield the perturbation for the current iteration. Thus, the complete recurrence equation of transformation-based attacks is

$$\tilde{x}_{t+1} = Clip(\tilde{x}_t + \frac{\alpha}{N}\sum_{i=1}^{N}\frac{\partial J(M_S(T(\tilde{x}_t;\theta_{T_i});\theta_{M_S}),y)}{\partial \tilde{x}_t}). \quad (3)$$

For clarity, no additional operation for other strategies is introduced to discuss. In addition, the *Clip* function will be omitted in the following discussions.

3.2 Motivation: Checkerboard Artifacts

We find that some attacks, such as Admix [66], SSIM [42], and STM [23] generate adversarial examples with checkerboard artifacts when using attention-based models as surrogates, and exhibit relatively low transferability. In contrast, other attacks such as SIA [67], DeCoWA [36], and BSR [64] do not produce checkerboard artifacts and exhibit higher transferability, as shown in Figure 2. This raises the following questions:

(1) What causes checkerboard artifacts to appear?
(2) How are checkerboard artifacts related to transferability?

With these questions in mind, we start from Equation 3 to analyze the causes of checkerboard artifacts.

3.3 Differentiation Modes

3.3.1 Definitions of Two Differentiation Modes. Recurrence Equation 3 represents a common differentiation mode in transformation-based attacks. In the Equation 3, $\tilde{x}_t = x + \delta_t$, where x is a constant and the variable δ_t represents the adversarial perturbation at iteration t. Because $\partial\delta_t/\partial(x+\delta_t) = I$, according to the chain rule, Equation 3 can be deduced as

$$\tilde{x}_{t+1} = \tilde{x}_t + \frac{\alpha}{N}\sum_{i=1}^{N}\frac{\partial J(M_S(T(\tilde{x}_t;\theta_{T_i});\theta_{M_S}),y)}{\partial \delta_t}. \quad (4)$$

This means that Equation 3 essentially generates new perturbations by differentiating with respect to the perturbations from the previous iteration. Although Equation 4 is more commonly used, there exists another mode that differentiates with respect to the transformed example $T(\tilde{x}_t;\theta_{T_i})$, with the recurrence equation by

$$\tilde{x}_{t+1} = \tilde{x}_t + \frac{\alpha}{N}\sum_{i=1}^{N}\frac{\partial J(M_S(T(\tilde{x}_t;\theta_{T_i});\theta_{M_S}),y)}{\partial T(\tilde{x}_t;\theta_{T_i})}. \quad (5)$$

The difference in backpropagation between the two differentiation modes is illustrated in Figure 3. For backpropagation path II corresponding to recurrence Equation 5, it passes only through the loss function and the surrogate model. In contrast, backpropagation path I, corresponding to recurrence Equation 4, additionally passes through the inverse transformation. For description, this paper defines the differentiation mode corresponding to Equation 4, which computes the gradient with respect to the perturbation, as **differentiation mode (a)**, while the mode corresponding to Equation 5, which differentiates with respect to the transformed image, is defined as **differentiation mode (b)**.

3.3.2 Equivalence Conditions of Two Differentiation Modes.

Theorem 3.1. *T being a pixel-wise addition transformation is a sufficient condition for the equivalence of two modes.*

Proof. When T is a transformation of pixel-wise constant addition, Equation 5 becomes

$$\tilde{x}_{t+1} = \tilde{x}_t + \frac{\alpha}{N}\sum_{i=1}^{N}\frac{\partial J(M_S(x+\delta_t+\theta_{T_i});\theta_{M_S}),y)}{\partial (x+\delta_t+\theta_{T_i})}. \quad (6)$$

Since the equation could be further simplified to

$$\tilde{x}_{t+1} = \tilde{x}_t + \frac{\alpha}{N}\sum_{i=1}^{N}\frac{\partial J(M_S(x+\delta_t+\theta_{T_i});\theta_{M_S}),y)}{\partial \delta_t}. \quad (7)$$

Similarly, substituting T into Equation 4 also leads to Equation 7. □

This indicates that when using a pixel-wise addition transformation, modes (a) and (b) are functionally equivalent. Moreover, mode (b) can reduce backpropagation latency.

3.3.3 Discrepancies of Two Differentiation Modes.

Theorem 3.2. *Any image transformation combined with differentiation mode (b) is equivalent to a pixel-wise addition transformation.*

Proof. In Equation 5, the partial derivative variable is $T(\tilde{x}_t;\theta_{T_i})$. We could replace this whole term with $T(\tilde{x}_t;\theta_{T_i}) = \delta_t + c$, where c is a constant, leading to

$$\tilde{x}_{t+1} = \tilde{x}_t + \frac{\alpha}{N}\sum_{i=1}^{N}\frac{\partial J(M_S(\delta_t+c;\theta_{M_S}),y)}{\partial (\delta_t+c)}. \quad (8)$$

Treating δ_t as the variable, Equation 8 can be further simplified to

$$\tilde{x}_{t+1} = \tilde{x}_t + \frac{\alpha}{N}\sum_{i=1}^{N}\frac{\partial J(M_S(\delta_t+c;\theta_{M_S}),y)}{\partial \delta_t}. \quad (9)$$

This essentially corresponds to the recurrence equation of transformation based attacks with a pixel-wise addition transformation. □

Thus, transformations combined with differentiation mode (b) always degenerate into pixel-wise addition transformations. As shown in Figure 3, this degradation stems from skipping the inverse transformation in backpropagation. In contrast, differentiation mode (a) fully leverages the augmentation introduced by the inverse transformation, making transformations take effect.

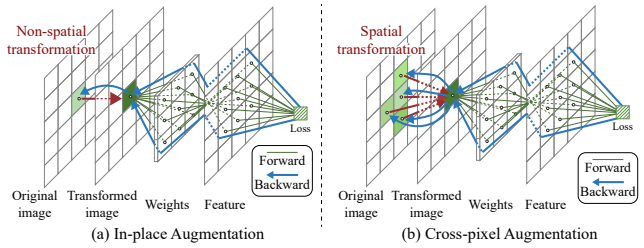


Figure 4: In-place and cross-pixel augmentations.

Theorem 3.3. *Differentiation mode (a) incorporates augmentation information from the inverse transformation based on different transformations.*

Proof. According to the chain rule, Equation 4 could be expanded as

$$\tilde{\boldsymbol{x}}_{t+1} = \tilde{\boldsymbol{x}}_t + \frac{\alpha}{N}\sum_{i=1}^{N}\frac{\partial J(M_S(T(\tilde{\boldsymbol{x}}_t;\boldsymbol{\theta}_{T_i});\boldsymbol{\theta}_{M_S}),y)}{\partial T(\tilde{\boldsymbol{x}}_t;\boldsymbol{\theta}_{T_i})}\frac{\partial T(\tilde{\boldsymbol{x}}_t;\boldsymbol{\theta}_{T_i})}{\partial \boldsymbol{\delta}_t}. \quad (10)$$

The first factor matches Equation 5, representing the shared back-propagation path of modes (a) and (b), while term $\partial T(\tilde{\boldsymbol{x}}_t;\boldsymbol{\theta}_{T_i})/\partial\boldsymbol{\delta}_t$ in mode (a) captures the information gain from the inverse transformation. □

At this point, we have a preliminary understanding of the possible outcomes when transformations are combined with differentiation modes (a) or (b). Any transformation combined with mode (b) is equivalent to a pixel-wise addition transformation, while transformations combined with mode (a) introduce a gain term $\partial T(\tilde{\boldsymbol{x}}_t;\boldsymbol{\theta}_{T_i})/\partial\boldsymbol{\delta}_t$. However, the contribution of the gain term remains unclear. In fact, as we will see next, when a spatial transformation is applied, the gain term helps eliminate checkerboard artifacts.

3.4 In-place and Cross-pixel Augmentations

3.4.1 Pixel-correlated Path and Spatial Transformations. For description, we first define the concepts of pixel-correlated path and spatial transformations.

Definition 3.4. **Pixel-correlated path** of a pixel $\boldsymbol{x}_{i,j}$ refers to the ordered sequence of neurons it passes through during forward propagation in the neural network.

Definition 3.5. **Non-spatial transformations** transform each pixel (i,j) independently as $T(\boldsymbol{x})_{i,j} = f_T(\boldsymbol{x}_{i,j})$. **Spatial transformations** make $T(x)_{i,j}$ dependent on multiple pixels, expressed as $T(\boldsymbol{x})_{i,j} = f_T(\boldsymbol{x}_{i_1,j_1}, \boldsymbol{x}_{i_2,j_2}, ..., \boldsymbol{x}_{i_n,j_n})$.

3.4.2 Gains Introduced by Non-spatial and Spatial Transformations. When differentiation mode (a) is combined with a non-spatial transformation, the gradient at (i,j) in the transformed image $T(\tilde{\boldsymbol{x}}_t;\boldsymbol{\theta}_T)$ propagates only back to (i,j) in $\tilde{\boldsymbol{x}}_t$ after the inverse transformation. This prevents pixel-correlated path information from spreading across different locations in (i,j) in $\tilde{\boldsymbol{x}}_t$, isolating perturbations. As shown in Figure 4(a), we define such a model augmentation, which produces isolated perturbations, as **In-place Augmentations**. In contrast, when differentiation mode (a) is combined with spatial transformations, the gradient at (i,j) in the transformed image $T(\tilde{\boldsymbol{x}}_t;\boldsymbol{\theta}_T)$ undergoes an inverse transformation, propagating to multiple locations in the image $\tilde{\boldsymbol{x}}_t$. This distributes pixel-correlated path information beyond (i,j), facilitating the fusion of pixel-correlated path information across different locations. As shown in Figure 4(b), we define such a model augmentation, which enhances the fusion of pixel-correlated path information across different locations, as **Cross-pixel Augmentations**.

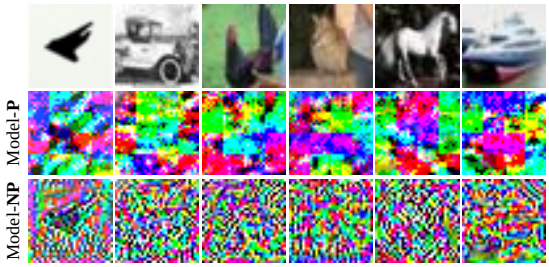


Figure 5: Perturbations generated on CIFAR-10.

Thus, from the perspective of pixel-correlated path information interaction, the gain term $\partial T(\tilde{\boldsymbol{x}}_t;\boldsymbol{\theta}_{T_i})/\partial\boldsymbol{\delta}_t$ enables information exchange across different locations. In addition, according to Theorem 3.2, any transformation combined with differentiation mode (b) degenerates into a pixel-wise addition transformation, meaning mode (b) can only achieve in-place augmentations.

3.4.3 Explanations of Checkerboard Artifacts and Transferability. Based on the concepts of in-place and cross-pixel augmentations, we could explain the conditions under which checkerboard artifacts occur and further clarify the mechanism of model augmentations for transformation-based transferability.

Checkerboard artifacts arise due to the patching operation typically performed in the first layer of attention-based surrogate models. This operation leads to gradient isolation between patches during back-propagation. To further illustrate, we conduct the following experiments. We design two models: Model-P and Model-NP. Both models have six intermediate convolution layers of $(K=3, S=1, P=1)$ with BN2 and ReLU, followed by a fully connected layer. The only difference between the two models is that the 1st layer of Model-P is a 8×8 patching operation of $(K=8, S=8, P=0)$, while the 1st layer of Model-NP is a standard convolution layer with $(K=3, S=1, P=1)$. These two models are trained on CIFAR-10, and adversarial perturbations are generated by Admix. The results after max-min normalization are shown in Figure 5. Perturbations generated by Model-P exhibit significant checkerboard artifacts, whereas those generated by Model-NP do not. This fact indicates that the checkerboard artifacts originate from the patching operation rather than the attention in the intermediate layers. If the inverse transformation enables interaction between perturbation information from different locations, this isolation is broken, and checkerboard artifacts disappear. Otherwise, they persist. In other words, in-place augmentations cannot eliminate checkerboard artifacts, whereas cross-pixel augmentations can. Checkerboard artifacts serve as an indicator of the use of cross-pixel augmentations to improve transferability.

Table 1: Summary of existing methods under the proposed classification framework. *Mode* indicates differentiation mode. *w/o* indicates whether the corresponding attack includes non-spatial transformations. *long-range* and *local* suggest whether the spatial transformation operates on distant pixels or within a local neighborhood. ↑ and ↓ indicate the flexibility of transformation. *Num* indicates the number of transformation types included. *All* indicates whether all transformations are used in each augmentation. *Artifact* indicates the presence of checkerboard artifacts.

Attack	*Year*	*Mode*	*Transformation Type*			*Num*	*All*	*Artifact*
			long-range	*local*	*w/o*			
DIM [70]	19	(a)			✓	1	✓	✓
TIM [19]	19	(a)		✓		1	✓	✓
SIM [35]	19	(a)			✓	1	✓	✓
Admix [66]	21	(a)			✓	1	✓	✓
SSIM[42]	22	(b)	✓	✓		1	✓	✓
STM [23]	23	(b)		✓		1	✓	✓
SIA [67]	23	(a)	✓↓	✓	✓	8		
DeCoWA [36]	24	(a)		✓		1	✓	
BSR [64]	24	(a)	✓↑	✓		2	✓	
I-C (Ours)	-	(a)	✓↑	✓	✓	3	✓	

In-place and cross-pixel augmentations offer two complementary ways for improving transferability, and leveraging both is essential to achieving desirable performance. In-place augmentations augment the pixel-correlated paths in an isolated, pixel-wise manner using *external* auxiliary functions, with their transferability stemming from the new parameters and structures introduced by these functions. In contrast, cross-pixel augmentations boost transferability by integrating information from pixel-correlated paths across different spatial locations. By utilizing *internal* features from various locations within the model, it helps prevent overfitting.

4 The Attack Paradigm

Based on the identified mechanism for enhancing transferability, we propose a design paradigm that fully leverages in-place and cross-pixel augmentations to improve transferability. To demonstrate the paradigm's effectiveness, we follow it to design a novel attack using simple transformations to explore its lower bound. Extensive experiments in Section 5 will validate the superiority of the proposed paradigm.

4.1 Design Paradigm

4.1.1 Auxiliary Functions. To integrate the transferability of in-place augmentations, designers need to construct pixel-wise auxiliary functions. For an image $\boldsymbol{x}$ with channel count C, width W, and height H, we could design $C \times W \times H$ distinct pixel-wise auxiliary functions $f_{T_{c,i,j}} = (\boldsymbol{x}_{c,i,j}; \boldsymbol{\theta}_{T_{c,i,j}})$. They could be arbitrarily complex differentiable functions, and their optimal forms remain an open question for future research.

4.1.2 Pixel Interaction. To integrate the transferability of cross-pixel augmentations, we adopt the differentiation mode (a). Pixel interactions across different locations could be categorized into *long-range interactions* and *local interactions*. Long-range pixel-correlated paths exhibit greater variation, enriching features diversity, while local pixel-correlated paths have smaller differences, accelerating attack convergence.

Table 2: List of surrogate and target models.

	CNNs			*ViTs*		
Sur	(1) ResNet-50	(R50)	[25]	(4) ViT-B/32	(V-B32)	[20]
	(2) EfficientNet-B0	(EB0)	[59]	(5) ViT-B/16	(V-B16)	[20]
	(3) MobileNet-V2	(MV2)	[53]	(6) BeiT-B/16	(BT-B16)	[7]
Tar	(1) ConvNeXt-B	(ConNX-B)	[41]	(9) ViT-B/8	(V-B8)	[20]
	(2) WResNet50-2	(WR50)	[72]	(10) Swinformer-B	(Swin-B)	[40]
	(3) DenseNet-161	(D161)	[27]	(11) PiT-B	(PiT-B)	[26]
	(4) EfficientNet-B2	(EB2)	[59]	(12) ConvFormer-B	(Conv-B)	[37]
	(5) GoogLeNet	(GoogLeN)	[58]	(13) XCiT-S/12/8	(XCiT-S)	[2]
	(6) Xception-71	(Xcept71)	[15]	(14) Visformer-S	(Visf-S)	[14]
	(7) IncRes-V2	(IncResV2)	[57]	(15) Caformer-M/36	(Caf-M)	[76]
	(8) RegNetX800MF	(Reg-X)	[50]	(16) PoolFormer-M/36	(Pool-M)	[71]

4.2 The I-C Attack

We design a new transformation-based attack *In-place and Cross-pixel Attack (I-C Attack)* using simple auxiliary functions and pixel interaction transformations. The auxiliary function applies a basic noise addition transformation, i.e. $f_{T_{c,i,j}}(\boldsymbol{x}_{c,i,j}; \boldsymbol{\theta}_T) = \boldsymbol{x}_{c,i,j} + \boldsymbol{\theta}_{T_{c,i,j}}$, where $\boldsymbol{\theta}_{T_{c,i,j}}$ follows a uniform distribution $\mathcal{U}(-a, a)$. Long-range interactions are introduced by block shuffle, where the image is unevenly divided into $b \times b$ blocks and shuffled. Bilinear interpolation is employed for local interactions, where the image is up-sampled by an expansion ratio of r via bilinear interpolation and then center cropped. These three transformations are sequentially cascaded in the order of: block shuffle → noise addition → bilinear interpolation. Despite I-C attack is a simple design example, thanks to the effective utilization of in-place and cross-pixel augmentations, extensive experiments in Section 5 will demonstrate that I-C attack outperforms existing methods in both attack performance and computational efficiency across various scenarios.

5 Experiments and Results

In this section, we will empirically compare existing transformation-based attacks with the proposed I-C attack across various scenarios, including single and ensemble surrogate models, non-targeted and targeted attacks, undefended and defended targets, different perturbation budgets, and combinations with various gradient-based attacks. In addition, we will conduct ablation studies on the components of the I-C attack and further analyze the relationship between checkerboard artifacts and transferability.

5.1 Setup

Dataset. Following previous works, this paper adopts a subset of the ImageNet validation set [51] for attacks. This subset consists of one image per class across 1,000 classes.

Baselines and Parameter Settings. This paper compares I-C attack with various advanced baselines, including Admix (ICCV'21), SSIM (ECCV'22), STM (ACMMM'23), SIA (ICCV'23), DeCoWA (AAAI'24) and BSR (CVPR'24). For Admix, the number of scaled copies $m_1 = 5$, admixed image number $m_2 = 4$ and the admix ratio $\eta = 0.2$. For SSIM, the tuning factor $\rho = 0.5$. For STM, the mixing ratio $\gamma = 0.5$ and the noise upper bound $\beta = 2$. For SIA, the splitting number $s = 3$. For DeCoWA, the number of control points $M = 9$ and the learning rate $\beta = 0.01$. For BSR, images are split into 2×2 blocks with the maximum rotation angle $\tau = 24°$. For I-C, when attacking

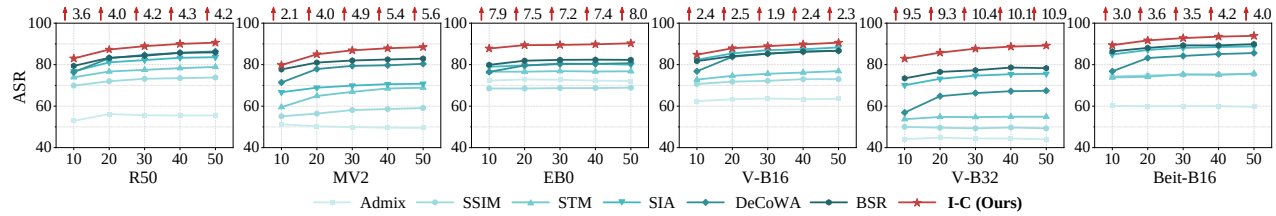


Figure 6: Non-targeted attacks on single surrogate with varying iterations T. Each point represents the average ASR (%) across 16 target models. ↑ indicates the performance gain of I-C over the suboptimal method.

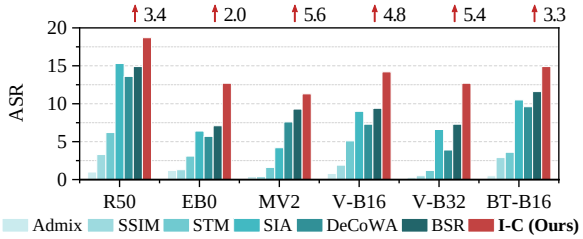


Figure 7: Cross-genus targeted attacks. Each bar represents the average ASR (%) across eight cross-genus target models. ↑ indicates the performance gain of I-C over the suboptimal method.

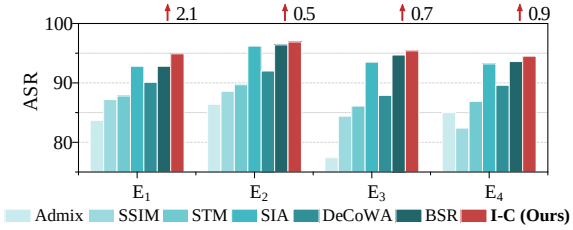


Figure 8: Non-targeted attacks on ensemble surrogate. Each bar represents the average ASR (%) across 16 target models. ↑ indicates the performance gain of I-C over the suboptimal method.

an undefended target, $a = 0.07$, $b = 3$ and $r = 1.40$. When attacking a defended target, this paper sets $a = 0.15$, $b = 3$ and $r = 1.10$. By default, the number of iteration $T = 10$, the perturbation budget $\epsilon = 16/255$, the step size $\alpha = 1.6/255$, and the number of image transformations is $N = 20$. All attacks are combined with MI-FGSM.

Surrogates and Targets. For comprehensive experimental results, various CNN and ViT models are selected as surrogates and targets, as listed in Table 2.

5.2 Comparisons with SOTA Methods

5.2.1 Non-targeted Attacks on Single Surrogate and Convergence Analysis. In this experiment, we compare the black-box performance of different attacks under a single surrogate model setting. Meanwhile, we analyze the convergence of different attacks by varying the number of iterations T. Specifically, we set $T = 10, 20, 30, 40, 50$ with a step size of $\alpha = 16/255/T$. Each attack generates adversarial examples using the 6 surrogate models in Table 2 and attacks 16 target models. The average attack success rates (ASRs) are presented in Figure 6. The results show that I-C attack consistently outperforms state-of-the-art methods across different iterations T. Additionally, at $T = 10$, all attacks are nearly converged, with only slight improvements as T increases. Unless stated otherwise, subsequent experiments of non-targeted attacks follow previous works [23, 64, 66] and set $T = 10$.

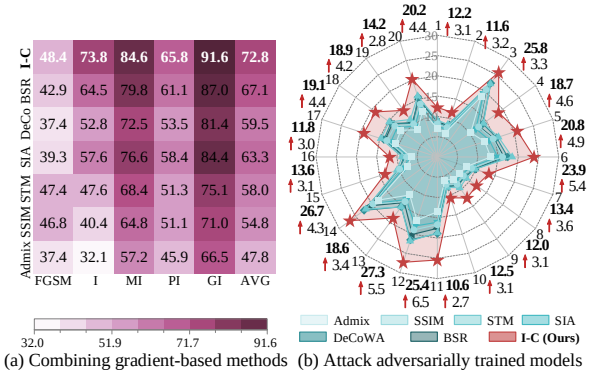
5.2.2 Cross-genus Targeted Attacks. This experiment evaluates different attacks on the challenging cross-genus targeted attacks, where CNNs attack ViTs or vice versa, using the models in Table 2. Since targeted attacks require more iterations to converge than non-targeted attacks, this experiment sets the iterations $T = 50$. The experimental results are presented in Figure 7. Across all surrogate models, I-C significantly outperforms existing transformation-based attacks.

5.2.3 Non-targeted Attacks on Ensemble Surrogate. Beyond using transformation-based attacks alone, we also evaluate their performance when combined with other strategies. This experiment examines these attacks by integrating them with ensemble-based strategies. For a comprehensive evaluation, we construct four ensemble models: E_1={R18, R34, R50}, E_2={R34, MV2, V-S16, V-S32}, E_3={V-S16, V-S32, BT-B16} and E_4={EB0, MV2, IncV3, Reg-Y}. Consistent with [39], the ensemble loss is

$$Loss = J(\sum_{i=1}^{|\Phi|} \alpha_i M_i(x), y), \tag{11}$$

where Φ is the model set for each ensemble model. The weighting coefficient α_i is set to $\frac{1}{|\Phi|}$ in this experiment. Figure 8 shows that I-C outperforms existing attacks under ensemble settings.

5.2.4 Combined with Different Gradient-based Strategies. Additionally, we combine transformation-based attacks with different gradient-based strategies to compare their performance. We integrate each transformation-based attack with FGSM, I-FGSM [30], MI-FGSM [18], PI-FGSM [22] and GI-FGSM [63] to evaluate their non-targeted attack performance under single surrogate model. The experimental results are presented in Figure 9(a). The results show that when integrated with these gradient-based attacks, I-C attack consistently outperforms existing transformation-based attacks. This suggests that for a new gradient-based strategy, I-C attack is more likely to achieve better performance when combined compared to other attacks.



(a) Combining gradient-based methods (b) Attack adversarially trained models

Figure 9: (a) Average ASR (%) of each transformation-based attack when integrated with different gradient-based strategies. (b) Average ASRs (%) against advanced adversarially trained models. ↑ indicates the performance gain of I-C over the suboptimal method.

5.2.5 Attacks under Different Perturbation Budgets. The above experiments are conducted with a perturbation budget of $\epsilon = 16/255$. In practice, an adversary may adjust the budget for stealth or effectiveness. This study evaluates non-targeted attack performance under $\epsilon = 8/255, 12/255, 16/255, 20/255, 24/255, 28/255$. The results are shown in Figure 10. Across all perturbation budgets, I-C outperforms existing methods, especially when attacking attention-based models.

5.2.6 Attacking Adversarially Trained Models. Additionally, we are curious about the performance of these transformation-based attack against defense mechanisms. Therefore, this experiment evaluates them to attack 20 advanced adversarially trained models: (1) ConvNX-L [3], (2) ConvNXV2-L + SwinT-L [5], (3)WR50 [10], (4) XCiT-L [16], (5) XCiT-M [16], (6) XCiT-S [16], (7) ConvNX-B [38], (8) ConvNX-L [38], (9) Swin-B [38], (10) Swin-L [38], (11) Swin-B [46], (12) ViT-B [46], (13) RaWR101 [48], (14) WR50 [52], (15) ConvNX-B + ConvStem [54], (16) ConvNX-L + ConvStem [54], (17) ConvNX-S + ConvStem [54], (18) ConvNX-T + ConvStem [54], (19) ViT-B + ConvStem [54], (20) ViT-S + ConvStem [54]. The results in Figure 9(b) demonstrate the superiority of I-C attack in attacking adversarially trained models.

5.3 Ablation Studies

5.3.1 Parameter Ablation. This experiment conducts an ablation study on the three transformations that constitute the I-C attack by individually analyzing their parameters. This aims to further investigate the characteristics of I-C and the roles played by in-place and cross-pixel augmentations. We fix two parameters while varying the other to examine the relationship between parameter changes and the success rate curve of single-surrogate non-targeted attacks. The surrogate and target models are presented in Table 2. The average ASR results are shown in Figure 11. It can be observed that when the noise addition parameter $a = 0$, the block shuffle parameter $b = 1$, or the bilinear interpolation parameter $c = 1$, the attack success rate drops significantly. This indicates that in-place augmentations, long-range cross-pixel augmentations, and local

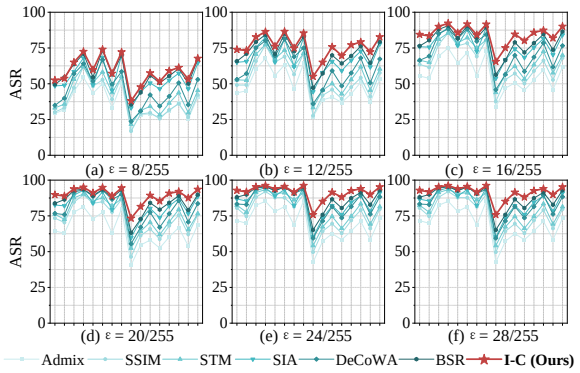


Figure 10: Average ASRs (%) on six surrogate models under different perturbation budgets. The x-axis represents target models (1) to (16) in Table 2.

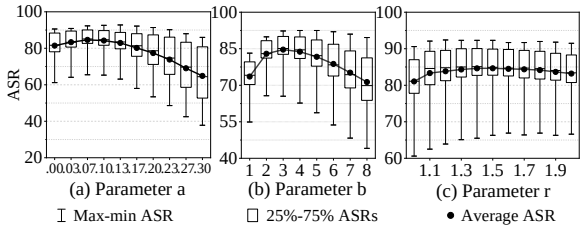


Figure 11: I-C performance with (a) varying noise addition parameter a, (b) different block shuffle parameter b, (c) varying bilinear interpolation parameter r.

cross-pixel augmentations each provide unique features that cannot be substituted by the other two transformations. Additionally, as the parameters a, b and c increase beyond their optimal values, the ASRs begin to decline significantly. This suggests that, for a fixed number of iterations, excessive transformations introduce features that hinder adversarial transferability.

6 Conclusion

Inspired by checkerboard artifacts, this paper classifies model augmentations into in-place and cross-pixel augmentations and proposes a transformation-based attack design paradigm to enhance transferability. Instead of relying on traditional image transformations, our paradigm emphasizes designing auxiliary functions and pixel interaction mechanisms, reducing trial-and-error and improving efficiency. Despite its simplicity, the proposed I-C attack surpasses existing transformation-based attacks across various scenarios. In future work, further exploring the optimal forms of auxiliary functions and pixel interaction mechanisms to fully exploit in-place and cross-pixel augmentations would be promising.

Acknowledgments

This work was supported in part by the University of Macau under Grants MYRG-GRG2023-00131-FST and MYRG-GRG2024-00065-FST-UMDF, and in part by the Science and Technology Development Fund, Macau SAR, under Grants 0141/2023/RIA2 and 0193/2023/RIA3.

References

[1] Abdullah Al-Dujaili and Una-May O'Reilly. 2020. Sign bits are all you need for black-box attacks. In *International Conference on Learning Representations*.

[2] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. 2021. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems* 34 (2021), 20014–20027.

[3] Sajjad Amini, Mohammadreza Teymoorianfard, Shiqing Ma, and Amir Houmansadr. 2024. MeanSparse: Post-Training Robustness Enhancement Through Mean-Centered Feature Sparsification. *arXiv preprint arXiv:2406.05927* (2024).

[4] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. 2020. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*. Springer, 484–501.

[5] Yatong Bai, Mo Zhou, Vishal M Patel, and Somayeh Sojoudi. 2024. MixedNUTS: Training-Free Accuracy-Robustness Balance via Nonlinearly Mixed Classifiers. *arXiv preprint arXiv:2402.02263* (2024).

[6] Shumeet Baluja and Ian Fischer. 2017. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387* (2017).

[7] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2021. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254* (2021).

[8] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2017. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248* (2017).

[9] Thomas Brunner, Frederik Diehl, Michael Truong Le, and Alois Knoll. 2019. Guessing smart: Biased sampling for efficient black-box adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4958–4966.

[10] Erh-Chung Chen and Che-Rung Lee. 2024. Data filtering for efficient adversarial training. *Pattern Recognition* 151 (2024), 110394.

[11] Huanran Chen, Yichi Zhang, Yinpeng Dong, Xiao Yang, Hang Su, and Jun Zhu. 2023. Rethinking model ensemble in transfer-based adversarial attacks. *arXiv preprint arXiv:2303.09105* (2023).

[12] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. 2020. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*. IEEE, 1277–1294.

[13] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*. 15–26.

[14] Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. 2021. Visformer: The vision-friendly transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*. 589–598.

[15] François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1251–1258.

[16] Edoardo Debenedetti, Vikash Sehwag, and Prateek Mittal. 2023. A light recipe to train robust vision transformers. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 225–253.

[17] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. 2022. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature* 602, 7897 (2022), 414–419.

[18] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 9185–9193.

[19] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4312–4321.

[20] Alexey Dosovitskiy. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[21] Diogo R Ferreira and Pedro J Carvalho. 2020. Deep learning for plasma tomography in nuclear fusion. In *NeurIPS 2020 Workshop on Deep Learning and Inverse Problems*. Curran Associates., NY, USA.

[22] Lianli Gao, Qilong Zhang, Jingkuan Song, Xianglong Liu, and Heng Tao Shen. 2020. Patch-wise attack for fooling deep neural network. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*. Springer, 307–322.

[23] Zhijin Ge, Fanhua Shang, Hongying Liu, Yuanyuan Liu, Liang Wan, Wei Feng, and Xiaosen Wang. 2023. Improving the transferability of adversarial examples with arbitrary style transfer. In *Proceedings of the 31st ACM International Conference on Multimedia*. 4440–4449.

[24] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[26] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. 2021. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*. 11936–11945.

[27] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[28] Hao Huang, Ziyan Chen, Huanran Chen, Yongtao Wang, and Kevin Zhang. 2023. T-sea: Transfer-based self-ensemble attack on object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20514–20523.

[29] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018. Black-box adversarial attacks with limited queries and information. In *International conference on machine learning*. PMLR, 2137–2146.

[30] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*. Chapman and Hall/CRC, 99–112.

[31] Maosen Li, Cheng Deng, Tengjiao Li, Junchi Yan, Xinbo Gao, and Heng Huang. 2020. Towards transferable targeted attack. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 641–649.

[32] Teng Li, Xingjun Ma, and Yu-Gang Jiang. 2025. AIM: Additional Image Guided Generation of Transferable Adversarial Attacks. *arXiv preprint arXiv:2501.01106* (2025).

[33] Xiu-Chuan Li, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. 2022. Decision-based adversarial attack with frequency mixup. *IEEE Transactions on Information Forensics and Security* 17 (2022), 1038–1052.

[34] Zhankai Li, Weiping Wang, Jie Li, Kai Chen, and Shigeng Zhang. 2024. UCG: A Universal Cross-Domain Generator for Transferable Adversarial Examples. *IEEE Transactions on Information Forensics and Security* (2024).

[35] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. 2019. Nesterov accelerated gradient and scale invariance for adversarial attacks. *arXiv preprint arXiv:1908.06281* (2019).

[36] Qinliang Lin, Cheng Luo, Zenghao Niu, Xilin He, Weicheng Xie, Yuanbo Hou, Linlin Shen, and Siyang Song. 2024. Boosting Adversarial Transferability across Model Genus by Deformation-Constrained Warping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 3459–3467.

[37] Xian Lin, Zengqiang Yan, Xianbo Deng, Chuansheng Zheng, and Li Yu. 2023. ConvFormer: Plug-and-play CNN-style transformers for improving medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 642–651.

[38] Chang Liu, Yinpeng Dong, Wenzhao Xiang, Xiao Yang, Hang Su, Jun Zhu, Yuefeng Chen, Yuan He, Hui Xue, and Shibao Zheng. 2024. A comprehensive study on robustness of image classification models: Benchmarking and rethinking. *International Journal of Computer Vision* (2024), 1–23.

[39] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2022. Delving into Transferable Adversarial Examples and Black-box Attacks. In *International Conference on Learning Representations*.

[40] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*. 10012–10022.

[41] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11976–11986.

[42] Yuyang Long, Qilong Zhang, Boheng Zeng, Lianli Gao, Xianglong Liu, Jian Zhang, and Jingkuan Song. 2022. Frequency domain model augmentation for adversarial attack. In *European conference on computer vision*. Springer, 549–566.

[43] Chen Ma, Li Chen, and Jun-Hai Yong. 2021. Simulating unknown target models for query-efficient black-box attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11835–11844.

[44] Thibault Maho, Teddy Furon, and Erwan Le Merrer. 2021. Surfree: a fast surrogate-free black-box attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10430–10439.

[45] Xiangxi Meng, Kaicong Sun, Jun Xu, Xuming He, and Dinggang Shen. 2024. Multi-modal Modality-masked Diffusion Network for Brain MRI Synthesis with Random Modality Missing. *IEEE Transactions on Medical Imaging* (2024).

[46] Yichuan Mo, Dongxian Wu, Yifei Wang, Yiwen Guo, and Yisen Wang. 2022. When adversarial training meets vision transformers: Recipes from training to architecture. *Advances in Neural Information Processing Systems* 35 (2022), 18599–18611.

[47] Anjie Peng, Zhi Lin, Hui Zeng, Wenxin Yu, and Xiangui Kang. 2023. Boosting Transferability of Adversarial Example via an Enhanced Euler's Method. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[48] ShengYun Peng, Weilin Xu, Cory Cornelius, Matthew Hull, Kevin Li, Rahul Duggal, Mansi Phute, Jason Martin, and Duen Horng Chau. 2023. Robust principles: Architectural design principles for adversarially robust cnns. *arXiv preprint arXiv:2308.16258* (2023).

[49] Yunxiao Qin, Yuanhao Xiong, Jinfeng Yi, and Cho-Jui Hsieh. 2023. Training meta-surrogate model for transferable adversarial attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 9516–9524.

[50] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. 2020. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10428–10436.

[51] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115 (2015), 211–252.

[52] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. 2020. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems* 33 (2020), 3533–3545.

[53] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4510–4520.

[54] Naman Deep Singh, Francesco Croce, and Matthias Hein. 2024. Revisiting adversarial training for imagenet: Architectures, training and generalization across threat models. *Advances in Neural Information Processing Systems* 36 (2024).

[55] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*. pmlr, 2256–2265.

[56] C Szegedy. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[57] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31.

[58] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[59] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, 6105–6114.

[60] Daiju Ueda, Taichi Kakinuma, Shohei Fujita, Koji Kamagata, Yasutaka Fushimi, Rintaro Ito, Yusuke Matsui, Taiki Nozaki, Takeshi Nakaura, Noriyuki Fujima, et al. 2024. Fairness of artificial intelligence in healthcare: review and recommendations. *Japanese Journal of Radiology* 42, 1 (2024), 3–15.

[61] Viet Quoc Vo, Ehsan Abbasnejad, and Damith C Ranasinghe. 2024. BruSLeAttack: A Query-Efficient Score-Based Black-Box Sparse Adversarial Attack. *arXiv preprint arXiv:2404.05311* (2024).

[62] Jie Wan, Jianhao Fu, Lijin Wang, and Ziqi Yang. 2024. Bounceattack: A query-efficient decision-based adversarial attack by bouncing into the wild. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1270–1286.

[63] Jiafeng Wang, Zhaoyu Chen, Kaixun Jiang, Dingkang Yang, Lingyi Hong, Pinxue Guo, Haijing Guo, and Wenqiang Zhang. 2024. Boosting the transferability of adversarial attacks with global momentum initialization. *Expert Systems with Applications* 255 (2024), 124757.

[64] Kunyu Wang, Xuanran He, Wenxuan Wang, and Xiaosen Wang. 2024. Boosting adversarial transferability by block shuffle and rotation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 24336–24346.

[65] Ruikui Wang, Yuanfang Guo, and Yunhong Wang. 2024. AGS: Affordable and Generalizable Substitute Training for Transferable Adversarial Attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 5553–5562.

[66] Xiaosen Wang, Xuanran He, Jingdong Wang, and Kun He. 2021. Admix: Enhancing the transferability of adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 16158–16167.

[67] Xiaosen Wang, Zeliang Zhang, and Jianping Zhang. 2023. Structure invariant transformation for better adversarial transferability. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4607–4619.

[68] Juanjuan Weng, Zhiming Luo, Shaozi Li, Nicu Sebe, and Zhun Zhong. 2023. Logit margin matters: Improving transferable targeted adversarial attack by logit calibration. *IEEE Transactions on Information Forensics and Security* (2023).

[69] Wang Xiaosen, Kangheng Tong, and Kun He. 2023. Rethinking the Backward Propagation for Adversarial Transferability. *Advances in Neural Information Processing Systems* 36 (2023), 1905–1922.

[70] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. 2019. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2730–2739.

[71] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. 2022. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10819–10829.

[72] Sergey Zagoruyko. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* (2016).

[73] Hui Zeng, Biwei Chen, and Anjie Peng. 2024. Enhancing targeted transferability via feature space fine-tuning. *arXiv preprint arXiv:2401.02727* (2024).

[74] Jianping Zhang, Yizhan Huang, Weibin Wu, and Michael R Lyu. 2023. Transferable adversarial attacks on vision transformers with token gradient regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16415–16424.

[75] Jianping Zhang, Weibin Wu, Jen-tse Huang, Yizhan Huang, Wenxuan Wang, Yuxin Su, and Michael R Lyu. 2022. Improving adversarial transferability via neuron attribution-based attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14993–15002.

[76] Kexuan Zhang, Xiaobei Zou, and Yang Tang. 2024. Caformer: Rethinking time series analysis from causal perspective. *arXiv preprint arXiv:2403.08572* (2024).

[77] Jingyuan Zhao, Wenyi Zhao, Bo Deng, Zhenghong Wang, Feng Zhang, Wenxiang Zheng, Wanke Cao, Jinrui Nan, Yubo Lian, and Andrew F Burke. 2024. Autonomous driving system: A comprehensive survey. *Expert Systems with Applications* 242 (2024), 122836.

[78] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. 2021. On success and simplicity: A second look at transferable targeted attacks. *Advances in Neural Information Processing Systems* 34 (2021), 6115–6128.