

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

IMPORTANT

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

ENHANCING ADVERSARIAL
TRANSFERABILITY ACROSS NEURAL
NETWORKS

KAISHENG LIANG

PhD

The Hong Kong Polytechnic University

2026

The Hong Kong Polytechnic University
Department of Computing

Enhancing Adversarial Transferability across Neural Networks

Kaisheng Liang

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
Aug 2025

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: _____

Name of Student: Kaisheng Liang

Abstract

Despite their widespread success in numerous applications, deep neural networks (DNNs) remain susceptible to adversarial attacks. These attacks involve imperceptible input modifications that can mislead models into producing erroneous outputs. A particular concern is the phenomenon known as adversarial transferability, where adversarial inputs crafted using one white-box model can deceive unknown black-box models. This transferability raises serious security concerns, especially in scenarios where model access is restricted but reliability is critical. Studying adversarial transferability is key to understanding and improving the robustness of DNNs.

Numerous methods have been introduced to improve adversarial transferability, spanning both non-targeted and targeted attacks. Although non-targeted attacks have received more attention, many existing methods lack a deep understanding of the underlying mechanisms. In addition, they fail to transfer effectively across models with different architectures, primarily due to overfitting to the surrogate model. In contrast, targeted attacks seek to manipulate model predictions toward a specific class, posing a much greater challenge due to their stricter success criteria. Current targeted attacks are relatively limited in number and suffer from high computational cost and low success rates.

In this thesis, we first present a novel perspective on the underlying mechanism of adversarial transferability. We observe that the non-robust style features of input images can hinder transferability by inducing overfitting to the surrogate model. Mo-

tivated by this insight, we propose Style-less Perturbation (StyLess), which suppresses non-robust style features by attacking a set of stylized surrogate models. These stylized models are constructed by injecting adaptive instance normalization layers into the surrogate and perturbing intermediate features in a training-free manner. StyLess enhances transferability by replacing the original surrogate model with stylized variants that reduce overfitting to the original image style.

Beyond non-targeted attacks, we address the more challenging task of generating targeted adversarial examples. We introduce Feature Tuning Mixup (FTM), a novel method that augments the feature space instead of the input space. FTM introduces learnable attack-specific perturbations across multiple layers and employs a momentum-based stochastic update strategy to iteratively tune these feature perturbations. Empirical results show that FTM achieves state-of-the-art adversarial transferability with high efficiency in targeted attacks.

We further explore two general strategies to boost adversarial transferability. The first, Soft Dropout Adaptation (SODA), introduces structured, learnable randomness in the input space to enhance adversarial transferability. Our analysis shows that a patch-based soft mask outperforms traditional random dropout. Based on this insight, SODA jointly adapts internal image content and surrounding context through a soft mask and a visual prompt. The second, Perspective-Invariant Attack (PIA), addresses the limitations of fixed perspective in transformations-based attacks. By applying randomized perspective transformations during the attack process, PIA improves the transferability of adversarial examples across models and input variations. Overall, this thesis contributes both conceptual insights and practical techniques for enhancing adversarial transferability in deep learning.

Keywords: Adversarial transferability, Adversarial attacks, Transfer-based attacks, Adversarial robustness, Deep neural networks

Publications Arising from the Thesis

1. Kaisheng Liang, and Bin Xiao, “StyLess: Boosting the Transferability of Adversarial Examples”, in *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2023.
2. Kaisheng Liang, Xuelong Dai, Yanjie Li, Dong Wang, and Bin Xiao, “Improving Transferable Targeted Attacks with Feature Tuning Mixup”, in *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2025.
3. Kaisheng Liang, Yiming Cao, Liying Li, Ho Chen, and Bin Xiao, “SODA: Soft Dropout Adaptation for Transferable and Efficient Adversarial Attacks”, manuscript submitted to *IEEE Transactions on Image Processing (TIP)* (2025).
4. Kaisheng Liang, Yiming Cao, and Bin Xiao, “Perspective-Invariant Attack with Enhanced Adversarial Transferability”, manuscript submitted to *IEEE Transactions on Information Forensics and Security (TIFS)* (2025).

Acknowledgments

I am deeply grateful to my supervisor, Professor Bin Xiao, whose invaluable guidance and constant support played a pivotal role in my PhD journey. His passion for research and academic rigor has had a profound impact on me. He gave me the freedom to explore the academic world, while remaining consistently supportive and encouraging. Under his mentorship, I gradually learned how to independently conduct and publish research, and how to identify meaningful and impactful research topics. I am fortunate to have him as my PhD advisor, and I feel well-prepared and confident as I look ahead to the next stage of my academic journey.

I am also grateful to the members of our research group for their academic and personal support, especially during my early days in Hong Kong. Their kindness and willingness to share knowledge fostered a truly supportive and collaborative environment. I would also like to acknowledge my co-authors and discussion partners for their insightful feedback and collaborative spirit. Collaborating with them was a great pleasure.

Last but not least, I would like to thank my family for their unconditional love and support. The challenges of the COVID-19 pandemic have deepened my appreciation for my family's constant support. They have been my source of strength and resilience, enabling me to move forward with confidence.

Table of Contents

Abstract	i
Publications Arising from the Thesis	iii
Acknowledgments	iv
List of Figures	x
List of Tables	xv
1 Introduction	1
1.1 Motivation	1
1.2 Literature Review	5
1.2.1 Adversarial Attacks	5
1.2.2 Non-targeted Transferable Attacks	8
1.2.3 Targeted Transferable Attacks	11
1.2.4 Defenses against Adversarial Attacks	12
1.3 Contributions and Thesis Organization	13

2	StyLess: Boosting the Transferability of Adversarial Examples	16
2.1	Introduction	17
2.2	Literature Review	19
2.3	Methodology	22
2.3.1	Problem Formulation	22
2.3.2	Motivation	23
2.3.3	Stylized Networks	24
2.3.4	Our StyLess Attack	27
2.4	Experiments	28
2.4.1	Experimental Setup	28
2.4.2	Attacking Unsecured Models	30
2.4.3	Attacking Secured Models	32
2.4.4	Evaluating with Ensemble-Based Attacks	32
2.4.5	Real-World Evaluation	34
2.4.6	Ablation Study	35
2.5	Summary	39
3	Improving Transferable Targeted Attacks with Feature Tuning Mixup	41
3.1	Introduction	42
3.2	Literature Review	45
3.3	Methodology	48
3.3.1	Preliminaries	48

3.3.2	Analysis of CFM	49
3.3.3	Feature Tuning Mixup (FTM)	50
3.4	Experiments	53
3.4.1	Experimental Setup	54
3.4.2	Success Rates of Targeted Attacks	55
3.4.3	Combination with Existing Techniques	58
3.4.4	Attacking Multimodal LLMs	60
3.4.5	Further Analysis	61
3.4.6	Ablation Study	63
3.4.7	Visualization	66
3.4.8	Discussion of Targeted Attacks under Defenses	66
3.5	Summary	69
4	Soft Dropout Adaptation for Transferable and Efficient Attacks	71
4.1	Introduction	72
4.2	Literature Review	74
4.3	Methodology	77
4.3.1	Preliminaries	77
4.3.2	Motivation	78
4.3.3	Proposed Soft Dropout Adaptation	79
4.4	Experiments	82
4.4.1	Experimental Setup	82

4.4.2	Attacking Unsecured Models	83
4.4.3	Attacking Secured Models	86
4.4.4	Efficiency and Complexity Analysis	88
4.4.5	Ablation Study	89
4.4.6	Visualization of SODA’s Augmented Images	93
4.5	Summary	94
5	Perspective-Invariant Attack with Enhanced Adversarial Transfer-	
	ability	100
5.1	Introduction	101
5.2	Literature Review	104
5.3	Methodology	106
5.3.1	Problem Formulation	106
5.3.2	Preliminary	107
5.3.3	Proposed Method	109
5.4	Experiments	111
5.4.1	Experimental Setup	111
5.4.2	Adversarial Transferability of PIA	113
5.4.3	Attacking Defense Methods	115
5.4.4	Combining with L2T Framework	119
5.4.5	Results on Multimodal LLMs	121
5.4.6	Computational Complexity	122
5.4.7	Ablation Study	123

5.4.8	Discussion of Non-targeted Attacks under Defenses	127
5.4.9	Comparison between SODA and PIA	130
5.5	Summary	134
6	Conclusion and Future Work	136
6.1	Conclusion	136
6.2	Future Work	137
	References	139

List of Figures

1.1	The contributions and organization of this thesis.	13
2.1	An overview of the proposed method. We construct stylized surrogate models by incorporating synthetic style features using an adaptive instance normalization (IN) layer, thereby reducing the surrogate’s dependence on non-robust style representations present in the original model.	18
2.2	Stylized loss gap visualization highlighting the difference between vanilla and stylized networks. The surrogate model is RN50. A higher adversarial loss here corresponds to a higher attack success rate. The expanding gap in baseline methods indicates that their attack effectiveness on our stylized surrogate models lags behind that on the vanilla model. In contrast, our proposed method, StyLess, effectively narrows this gap.	24
2.3	Comparing with other attack methods on various DNN models and defensive methods. RN50 is used as the surrogate model.	34
2.4	Ablation study on the insertion position of the IN layer. WRN101 or DN121 is used as the surrogate.	36
2.5	Impact of the number of stylized models, using RN50 as the surrogate.	37

2.6	Impact of synthesized style feature variation on clean loss and StyLess success rate. Star count indicates the degree of style modification. . .	38
2.7	Impact of different style statistics.	39
3.1	Effectiveness and efficiency of targeted attacks. Average attack success rates on 14 black-box models, along with the computation time required to generate an adversarial example. Our methods demonstrate superior performance with low computational cost, surpassing state-of-the-art targeted attacks.	43
3.2	Overview of our FTM attack. In the forward pass, the learnable perturbations are added to the output features. Only a small portion of network layers are randomly selected for perturbation updates and clean feature mixup in each attack iteration. In the backward pass, the gradients of the selected perturbations and adversarial image are computed together for update.	44
3.3	Success rates of targeted attacks under different iteration counts. . .	62
3.4	Attack success rates with different mixing ratio β . Each color represents the attack effectiveness on a specific target model.	63
3.5	Attack performance and computational costs with different numbers of ensemble models. Left: Our FTM vs. CFM using ensemble strategy. Right: Computational costs of our FTM and RAP. -ens2 means using 2 perturbed models, and so on.	64
3.6	Left: Attack success rates vs. p and α_{max} . Right: Attack success rates vs. p with/without clean features; The blue line is the success rates when the clean features from CFM are integrated with our FTM, while the orange line represents only using our learnable perturbations. . .	65

3.7	Visualization of targeted adversarial examples generated by different attack methods. The surrogate model used for attack generation is an ensemble of RN-50 and Inc-v3.	67
3.8	Visualization of targeted adversarial examples generated by different attack methods. The surrogate model used for attack generation is an ensemble of RN-50 and LeViT.	68
4.1	Overview of the proposed SODA. In each attack iteration, a soft mask and a visual prompt are jointly learned to modify the intermediate adversarial input. (a) Intermediate adversarial image; (b) Corresponding learned soft mask; (c) Input image after applying the soft mask; (d) Input image after applying the soft mask and the border prompt. . .	74
4.2	Illustration of visual prompt tuning. (a) Traditional learned prompt in pixel space; (b) Border prompt applied at a fixed position; (c) Border prompt applied at a random position.	80
4.3	Patch-based mask vs. pixel-based mask. Impact of different patch sizes of our mask on the success rate of SODA. When the patch size is 1, the mask becomes pixel-based.	91
4.4	Impact of different numbers of attack iterations on the attack success rate of our SODA.	92
4.5	Impact of different learning rates of our mask optimization on the attack success rate of our SODA.	93
4.6	Adversarial examples generated by SODA.	94

4.7	Visualization of the transformation process in our SODA. (a) The benign image; (b) the intermediate adversarial image during the iterative attack; (c) the learned soft mask corresponding to (b); (d) the masked intermediate adversarial image after applying the soft mask; (e) the final augmented input with both the soft mask and the visual prompt.	95
5.1	Our method significantly outperforms state-of-the-art attacks including SIA [133] and BSR [129]. SIA uses block-wise shuffling, while BSR uses block-wise shuffling combined with slight rotations, both of which are fixed perspective transformations. We consider altering the perspective of input images to increase the diversity of transformed images.	102
5.2	Visualization of perspective transformations applied to input images using our PIA method.	103
5.3	Overview of transformation-based attacks. Multiple transformed copies of the input image are used to calculate gradients for enhancing adversarial transferability.	108
5.4	Visualization of advanced input transformations for generating adversarial examples, including SIA and BSR.	108
5.5	Combining with L2T framework. Our perspective transformation can be added into the candidate pool of L2T to further improve its attack success rates.	121
5.6	Impact of the distortion parameter on the attack success rate of our PIA.	124
5.7	Impact of the number of image copies on the attack success rate of our PIA.	125

5.8	Impact of the number of attack iterations on the attack success rate of our PIA.	126
5.9	Impact of different combinations of transformation functions on the attack success rate of our PIA.	127
5.10	Impact of varying the number of image copies on attack success rates across different transformation function combinations.	128
5.11	Clean accuracy of the input images after block-wise shuffle, block-wise rotation, and perspective transformation.	129
5.12	Visualization of adversarial examples generated by the baseline attack and our PIA.	130
5.13	Comparison of cross-architecture gradient similarity between SODA and PIA. The curves illustrate the cosine similarity between the gradients computed on the source model (RN-50) and the target model (ViT) across attack iterations.	133

List of Tables

2.1	Attacking unsecured DNN models using StyLess and existing attack methods.	31
2.2	Attacking secured DNN models with StyLess and existing attack methods. RN50 or WRN101 is used as the surrogate model.	33
2.3	Combining with model ensemble attack methods.	35
3.1	Success rates (%) of targeted attacks evaluated across ten target models. The best-performing results are highlighted in bold, while the second-best are indicated with underlining. Time (s) represents the average computational time required to generate an adversarial example for a single image.	56
3.2	Success rates (%) of targeted attacks evaluated across five target models. The best-performing results are highlighted in bold, while the second-best are indicated with underlining. Time (s) represents the average computational time required to generate an adversarial example for a single image.	57
3.3	Targeted attack success rates (%) using DN-121 or LeViT as surrogate model. All methods are combined with MI-TI. The best results are shown in bold, and the second best results are underlined. The surrogate models are marked with * in the first column.	58

3.4	Success rates (%) of targeted attacks evaluated across ten target models, with ViT as the surrogate model. All methods are combined with MI-TI. The best results are shown in bold, and the second best results are underlined. Time (s) denotes the average computation time required to attack a single image. RDI-FTM-E-SI _{m₁=2} denotes using scaled input for RDI-FTM-E	59
3.5	Success rates (%) of targeted attacks evaluated across five transformer-based DNNs, with ViT as the surrogate model. All methods are combined with MI-TI. The best results are shown in bold, and the second best are underlined. Time (s) denotes the average time required to attack a single image. RDI-FTM-E-SI _{m₁=2} denotes using scaled input for RDI-FTM-E	59
3.6	Targeted attack success rates (%) of method combinations. The surrogate model used is an ensemble of RN-50 and Inc-v3. The best results are indicated in bold, while the second-best are underlined. Time (s) refers to the average time required to generate a targeted adversarial example for a single image.	60
3.7	Targeted attack success rates (%) using ensemble-based surrogate models. All methods are combined with MI-TI. The best results are shown in bold, and the second best results are underlined. The surrogate models are marked with * in the first column.	61
3.8	Evaluation on multimodal LLMs. The targeted adversarial examples are generated on ViT using RDI-FTM-E.	62
3.9	Accuracy (%) of clean images on the perturbed surrogate model with varying parameter p	66

3.10	Success rates (%) of targeted attacks evaluated across three adversarially trained defenses. The surrogate model is the ensemble of RN-50 and LeViT. The best-performing results are highlighted in bold, while the second-best are indicated with underlining.	69
3.11	Success rates (%) of targeted attacks evaluated across three adversarially trained defenses. The surrogate model is the ensemble of RN-50 and Inc-v3. The best-performing results are highlighted in bold, while the second-best are indicated with underlining.	70
4.1	Attack success rates (%) on eight target models. The best results are shown in bold.	85
4.2	Attack success rate (%) of combining DI-TI with existing methods against eight target models. The best results are shown in bold. . . .	86
4.3	Attack success rate (%) using an ensemble of models as the surrogate model against eight target models. * indicates the surrogate models. The best results are shown in bold.	87
4.4	Success rates (%) against various defense methods. The best results are shown in bold.	88
4.5	Attack success rates (%) against three ensemble-based adversarially trained defenses. The best results are shown in bold.	98
4.6	Average time for generating an adversarial image.	98
4.7	Soft mask vs. hard mask: ablation study on the type of mask. The surrogate model is Inc-v3. μ and σ are the mean and standard deviation of the Gaussian distribution. The best results are shown in bold.	99

4.8	Ablation study on the components of SODA. The surrogate model is Inc-v3. The best results are shown in bold.	99
4.9	Ablation study on the canvas size of the visual prompt in our SODA. The surrogate model is Inc-v3.	99
5.1	Attack performance on eight target models. All methods are based on MI. The best results are shown in bold.	114
5.2	Attack performance of combining DI-TI with existing methods against eight target models. All methods are based on MI. The best results are shown in bold.	116
5.3	Attack performance using an ensemble of models as the surrogate model against eight target models. All methods are based on MI. * indicates the surrogate models. The best results are shown in bold.	117
5.4	Attack success rates (%) against three ensemble-based adversarially trained defenses. All methods are based on MI. The best results are shown in bold.	118
5.5	Attack success rates (%) against four target models on the CIFAR-10 dataset. All methods are based on MI. The best results are shown in bold.	119
5.6	Success rates (%) against various defense methods on ImageNet. All methods are based on MI. The best results are shown in bold.	120
5.7	Attack success rates (%) against multimodal LLMs. We randomly sample 100 images from the ImageNet-compatible dataset as the victim images, and use ResNet-50 as the surrogate model.	122
5.8	Average time (s) for each attack to generate an adversarial example on the ImageNet-compatible dataset. The surrogate model is RN-50.	123

5.9	Adversarial robustness (%) of current pretrained models and our adversarial training methods under various transfer-based black-box attacks. The surrogate model used for adversarial attacks is DN-121. The best results are shown in bold.	131
5.10	Attack performance of SODA and PIA on eight target models. . . .	133

Chapter 1

Introduction

1.1 Motivation

In recent years, deep neural networks (DNNs) have achieved breakthrough progress in numerous critical domains, including computer vision [44, 118, 24] and natural language processing [116, 127], demonstrating remarkable perceptual and reasoning capabilities. However, extensive research [40, 67, 14, 29, 86] has revealed that DNN models exhibit significant vulnerability to adversarial examples crafted by applying imperceptible perturbations to original data, which can mislead DNN models into generating erroneous outputs.

Early studies on adversarial attacks primarily concentrated on the white-box setting, where attackers have complete visibility of the model’s parameters and gradients [40, 67, 14]. However, real-world attacks frequently occur under the black-box setting, where adversaries lack access to the target model’s structure or parameters and may even lack query access. In this context, a critical phenomenon has drawn widespread attention: adversarial transferability across models [143, 22, 23]. This refers to the phenomenon where adversarial inputs crafted using a locally trained substitute model are still capable of misleading a different, previously unseen tar-

get model. Transferable attacks pose a realistic threat because they do not require access to the target model. This helps attackers avoid detection at the input level and increases their chances of success, making such attacks especially dangerous for commercial systems and online APIs.

Transferable adversarial attacks have emerged as a core research topic in adversarial machine learning. Numerous studies [22, 143, 23, 132, 43, 139, 51, 71, 134, 151] have sought to uncover their underlying mechanisms, influencing factors, and strategies to enhance adversarial transferability. Meanwhile, defenses against such attacks have become increasingly urgent, especially as DNNs are deployed in safety-critical applications such as autonomous driving [37], medical diagnosis [100], and facial recognition [105], where the potential risks carry profound implications. Thus, a deeper understanding of adversarial transferability and its impact is essential for improving the robustness of DNN models in real-world environments.

Existing transferable adversarial attacks can be broadly categorized into two types: non-targeted and targeted attacks. Non-targeted attacks [131, 134, 151] aim to craft adversarial examples that cause misclassification by the victim model, regardless of the specific misclassified label. Although extensive efforts have been made to improve non-targeted attacks, the underlying mechanisms that drive this improvement remain poorly understood. Moreover, their cross-architecture transferability is still limited. For example, adversarial examples crafted using Convolutional Neural Network (CNN)-based surrogate models often fail to transfer effectively to Vision Transformer (ViT) models. Targeted attacks [156, 137, 12], which aim to mislead the model into predicting a specific target class, pose even greater challenges. Compared to non-targeted attacks, targeted attacks generally exhibit much lower transferability. In addition, targeted attacks often incur prohibitively high computational costs, further limiting their practicality.

In this thesis, we start with interpreting adversarial transferability from a novel perspective: the original style features of input images may hinder adversarial transfer-

ability. Through empirical analysis, we observe that existing iterative attack methods tend to increasingly incorporate the style characteristics of the victim images during optimization, which may negatively impact their adversarial transferability. Motivated by this observation, we propose a novel attack method, named Style-less Perturbations (StyLess), which enhances transferability by explicitly suppressing the influence of non-robust style features. Specifically, we introduce stylized networks by injecting adaptive instance normalization layers into surrogate models, perturbing the mean and variance of intermediate feature representations. This stylization process is training-free and efficient, as the parameters of the inserted normalization layers are initialized via sampling and interpolating from reference style images. StyLess replaces the original surrogate model with a set of stylized models during the attack process. The core idea is to use diverse synthesized styles to compete against the original image style, thus mitigating the style-induced overfitting of adversarial perturbations. Our StyLess not only provides a new lens to understand adversarial transferability, but also achieves state-of-the-art performance in generating transferable non-targeted adversarial examples across various architectures.

In addition to non-targeted attacks, we also investigate the transferability of targeted adversarial examples, which aim to mislead models into predicting a specific target class. Unlike non-targeted attacks, targeted attacks often require the generation of visual patterns that closely resemble features of the desired class, making them significantly more difficult to construct within a constrained perturbation budget. Furthermore, existing targeted attack methods typically suffer from low transferability and often incur substantial computational overhead. To address these challenges, we propose a novel method called Feature Tuning Mixup (FTM), which enhances the transferability of targeted adversarial examples in a more effective and efficient manner. While prior efforts in data augmentation mainly focus on the input space, FTM takes a different approach by augmenting the feature space, which is the first attempt to directly optimize perturbations across multiple intermediate layers.

This unexplored direction reveals that injecting attack-specific feature perturbations, which have been largely overlooked in previous work, can substantially improve the transferability of existing targeted attacks. FTM leverages a combination of randomized and optimized feature perturbations, coupled with an efficient stochastic update strategy that preserves computational efficiency while significantly boosting transferability. We evaluate FTM in various scenarios, including commercial black-box models such as ChatGPT and Gemini. Experimental results demonstrate that FTM achieves state-of-the-art performance in generating transferable targeted adversarial examples across a wide range of black-box DNN models.

We further investigate two general techniques for enhancing adversarial transferability: Soft Dropout Adaptation (SODA) and Perspective-Invariant Attack (PIA). Inspired by dropout’s role in generalization, we introduce structured, learnable randomness into the input space to enhance adversarial transferability. Our analysis shows that a differentiable, patch-based soft mask outperforms standard random dropout in achieving this goal. Building on this insight, we propose Soft Dropout Adaptation (SODA), which improves transferability by jointly adapting internal content and surrounding context. Specifically, SODA employs two complementary, co-adapted components: a content-modulating soft mask and a context-expanding visual prompt. SODA requires no additional training data and incurs minimal computational overhead. Next, we revisit widely used input transformation-based attack methods and identify a key limitation: most existing methods rely on fixed perspective transformations, which restrict the adaptability of adversarial examples to geometric variations and thereby limit their transferability. To address this limitation, we propose a new method termed Perspective-Invariant Attack (PIA), which enforces the invariance of adversarial examples to perspective transformations. Specifically, we construct a transformation pool comprising stochastic perspective transformations along with other input transformations. At each attack iteration, a transformation is randomly sampled from this pool and applied to the input image, thereby introducing ran-

domness and enhancing transferability. Our PIA is highly flexible and can be easily integrated into existing attack frameworks.

Overall, the methods proposed in this thesis are designed to improve the transferability of adversarial examples across diverse model architectures while preserving computational efficiency. Comprehensive experimental evaluations on a broad spectrum of DNN models, defense mechanisms, and vision language models demonstrate that our methods consistently and significantly enhance adversarial transferability across various attack scenarios.

1.2 Literature Review

1.2.1 Adversarial Attacks

Adversarial attacks can be categorized according to different criteria: (1) By knowledge level: white-box attacks vs. black-box attacks; (2) By application scenario: digital attacks vs. physical attacks; (3) By methodology: gradient-based attacks vs. generative modeling-based attacks; (4) By objective: non-targeted attacks vs. targeted attacks. The following sections will elaborate on each category.

White-box attacks vs. black-box attacks. White-box attackers have full access to the target model’s parameters and gradients, while black-box attackers do not. White-box attacks are commonly used to evaluate the robustness of deep neural networks (DNNs) and the effectiveness of defense mechanisms. When evaluating the robustness of defenses, white-box attacks are often referred to as adaptive attacks [2, 123], as the attacker can tailor the attack strategy based on feedback from the defense. Representative white-box attacks include the Fast Gradient Sign Method (FGSM)[40] and its iterative variant, the Iterative Fast Gradient Sign Method (I-FGSM)[67]. These methods generate adversarial examples by determining the gra-

dient sign of the loss function relative to the input data. However, the transferability of white-box attacks is limited, as they tend to overfit the surrogate models used during attack generation [22, 143]. In contrast, black-box attacks are more practical and challenging, as the attacker has no access to the target model’s parameters and gradients. Black-box attacks can generally be divided into two categories: transfer-based attacks[143] and query-based attacks[10]. Transfer-based attacks assume access to a surrogate model that performs a similar task to the target model. Adversarial examples are crafted on the surrogate model and directly transferred to the target model without querying it. Query-based attacks, on the other hand, do not rely on surrogate models but instead interact directly with the target model by submitting inputs and observing the outputs. These attacks often require thousands of queries to estimate gradients, making them more likely to be detected by the target system. Given these considerations, transfer-based adversarial attacks offer a better trade-off between stealth and practicality, and are the research topic of this thesis.

Digital attacks vs. physical attacks. Physical adversarial attacks [107, 29] aim to generate adversarial examples that remain effective when deployed in the physical world, requiring the adversarial perturbations to be physically realizable and printable. These attacks must consider specific physical scenarios, such as facial recognition systems and autonomous driving, which impose additional constraints. Representative physical attacks include adversarial eyeglass frames for deceiving facial recognition systems [107] and manipulated traffic signs for misleading autonomous vehicles [29]. Physical attacks are typically unrestricted [25, 26, 3, 50], meaning they are not confined to traditional ℓ_p norm constraints. Instead, they focus on crafting perturbations that are camouflaged or inconspicuous in the physical world, while still successfully misleading the target model. In addition, color-based attacks [47, 68, 155, 106, 152] are also a type of unrestricted attack, posing significant challenges to DNNs by manipulating the color of benign images without ℓ_p -norm constraints. In contrast, digital attacks [14, 86, 22, 23] are conducted purely in the digital

domain, without the need to consider physical devices such as cameras. This makes them more suitable for the systematic evaluation of model robustness. Moreover, digital attacks benefit from well-established threat models, typically incorporating ℓ_∞ or ℓ_2 norm constraints to ensure the stealthiness of perturbations. This thesis focuses on digital attacks, due to their clarity in formulation, reproducibility, and compatibility with standard evaluation practices.

Gradient-based attacks vs. generative modeling-based attacks. Current adversarial attack methods can be categorized into two types: gradient-based attacks [40, 22] and generative modeling-based attacks [96, 89]. Gradient-based attacks are typically built upon the Fast Gradient Sign Method (FGSM) framework [40, 67], which generates adversarial examples by iteratively using the gradient sign of the input. In contrast, generative modeling-based attacks rely on generative models [39] to directly generate adversarial examples. These methods generally require access to a complete dataset for training the generative model, which limits their practicality in real-world scenarios. However, once trained, generative models can produce adversarial examples with greater computational efficiency at inference time compared to gradient-based attacks. From a practical perspective, gradient-based attacks are more accessible, as they do not require training data and can leverage widely available pre-trained models, making them suitable for realistic attack scenarios. In addition, training a good surrogate model [70, 72] is an important topic in adversarial attacks. This thesis focuses on gradient-based attacks, due to their practicality and alignment with real-world attack settings.

Non-targeted attacks vs. targeted attacks. Non-targeted attacks [80, 132, 131, 134, 78] aim to generate adversarial examples that cause the model to produce incorrect predictions, regardless of the specific class. In contrast, targeted attacks [69, 156, 137, 11, 12] seek to mislead the model into predicting a specific target class. While non-targeted attacks have been widely studied, targeted attacks remain relatively underexplored. Non-targeted transferable attacks typically perform well

when the surrogate and target models share the same architecture. However, their effectiveness may drop significantly when the network architectures differ. For example, adversarial examples crafted using CNN-based surrogate models often fail to transfer effectively to Vision Transformers (ViTs). Moreover, most existing attack methods are tailored for CNNs. When attackers use ViTs as surrogate models, they often require different strategies [87, 136, 150]. Targeted attacks are more challenging than non-targeted ones and pose a greater threat in real-world applications. However, existing transferable targeted attacks often suffer from low transferability and high computational cost [77]. Compared to non-targeted attacks, targeted attacks typically require more optimization steps to generate adversarial perturbations that consistently induce the model to predict a specific target class. These perturbations are often semantically closer to the target class, making them more time-consuming to construct and more sensitive to the perturbation budget.

In summary, this thesis investigates adversarial transferability under black-box settings in the digital domain using gradient-based attack methods.

1.2.2 Non-targeted Transferable Attacks

This section reviews gradient-based non-targeted transferable attacks. The gradient-based attack framework typically consists of four components: the input data, surrogate model, loss function, and optimization method. We categorize existing attack methods according to these components. However, it is important to note that some attack methods involve multiple components simultaneously.

Input data. Input transformation-based attacks are among the most widely adopted methods for enhancing adversarial transferability. These methods apply various transformations to input images before feeding them into surrogate models. Diverse Input (DI) [143] applies random resizing and padding operations to generate diverse input images. Translation-Invariant (TI) [23] introduces a gradient-smoothing technique

using convolutional kernels to enhance transferability against spatial shifts. Resized Diverse Inputs (RDI) [160] modifies DI by setting the transformation probability $p = 1$ and resizing the augmented images back to their original resolution. Scale-Invariant (SI) [80] utilizes pixel scaling as a form of data augmentation and incorporates multiple scaled versions of the input to create the final augmented images. Admix [132] blends small portions of different images and integrates this approach with SI. Spectrum Simulation Attack (SSA) [84] manipulates image frequency components through Gaussian noise and random masking. Structure Invariant Attack (SIA) [133] and Block Shuffle and Rotation (BSR) [129] have achieved state-of-the-art performance by introducing block-wise transformations. SIA implements random shuffling and transformation of image blocks, while BSR extends this concept by incorporating slight rotations of individual blocks to further enhance diversity. Learn to Transform (L2T) [158] introduces a learning-based framework that identifies the most effective transformation strategy for each input image.

Surrogate model. This section reviews two main types of methods designed to improve adversarial transferability: model enhancement and model ensemble. Model enhancement methods aim to refine a given surrogate model to generate more transferable adversarial examples. Different network architectures and feature representations influence how adversarial perturbations behave. For instance, the linearity of DNNs is considered a key factor in adversarial transferability [40]. Based on this observation, LinBP [43] removes non-linear activations during backpropagation to improve adversarial transferability. SGM [139] adjusts the weights of the gradients from skip connections in residual networks. To better exploit intermediate representations, some methods introduce auxiliary classifiers trained on hidden feature spaces [55, 56], providing more effective gradient signals during attack generation. Adversarially trained models have also been shown to serve as better surrogates for generating transferable attacks [102, 126, 113], due to their smoother feature representations [54, 159]. Ghost Networks (GN) [76] introduces model diversity by perturbing skip connections and

dropout layers, effectively generating multiple virtual surrogate models from a single surrogate model. In model ensemble-based attacks, multiple surrogate models are used simultaneously to improve transferability [81]. Recent studies have further explored optimization strategies to enhance these ensemble attacks [144, 15, 16], making them more effective and adaptive.

Loss function. This section reviews intermediate-level adversarial attacks. While cross-entropy loss is widely used in adversarial attacks, feature-based loss functions have been shown to be more effective. Some methods aim to increase the feature-space discrepancy between benign and adversarial images [157], while others fine-tune adversarial examples at intermediate layers using iterative methods like Intermediate Level Attack (ILA) [51, 71]. These methods typically use the ℓ_2 norm to measure the distance between feature representations but do not consider the relative importance of different features. To address this, feature importance-based attacks [134, 151] identify and perturb the most critical features in the intermediate layers of surrogate models. Additionally, several refined loss functions have been proposed specifically for targeted attacks [57, 69, 156, 137], which will be discussed in the following subsection.

Optimization method. MI [22] employs momentum-based optimization to enhance gradient updates. Similarly, NI [80] leverages Nesterov acceleration to further refine the optimization process. Variance Tuning (VT) [131] incorporates gradient variance control to stabilize localized gradients. The core objective of these methods is to improve the stability of gradient updates during optimization. Additionally, Reverse Adversarial Perturbation (RAP) [97] generates adversarial examples situated in flat loss regions, while Sharpness-Aware Minimization (SAM) [33] has demonstrated effectiveness in enhancing DNN training and has also been adapted for adversarial attack optimization [153]. Further studies [36, 30] explore the connection between adversarial transferability and loss landscape flatness.

1.2.3 Targeted Transferable Attacks

Targeted attacks are more challenging than non-targeted ones, as they aim not only to cause misclassification but also to force the target model to produce a predefined class. These attacks can be broadly categorized into two types: generative modeling-based and gradient-based methods. Generative modeling methods typically require access to training datasets, either to train generative models directly or to construct auxiliary modules that support gradient-based attacks. Prior works have utilized datasets to build such auxiliary modules [55, 56], or to train generative models that produce targeted adversarial examples [91, 148, 135]. More recently, a study [31] demonstrates the use of CLIP [98] as guidance to train generative models for effective multi-target attacks. In contrast, gradient-based targeted attacks do not require any training data, making them more practical in black-box settings. These methods will be reviewed in detail in the following.

Gradient-based targeted attacks do not require access to the target model’s architecture or parameters, nor do they rely on training datasets. The Activation Attack (AA) [57] introduces a feature-space loss based on the distance between adversarial and benign images. Po+Trip [69] improves the loss formulation by incorporating Poincaré distance to adjust gradient magnitudes and by using triplet loss to address insufficient inter-class separability. The Logit Attack (Logit) [156] shows that a simple logit-based loss can outperform cross-entropy loss in generating more transferable targeted adversarial examples, thus significantly improving attack success rates. SU [137] demonstrates that more universal perturbations optimized in different local regions can lead to stronger targeted transferability. ODI [11] applies data augmentation in the image space by using randomly rendered 3D objects to enhance input diversity during attacks. At the feature level, CFM [12] proposes storing clean features from other images as random feature perturbations, thereby effectively mitigating overfitting to the surrogate model. CFM achieves state-of-the-art performance in targeted transfer-based attacks.

1.2.4 Defenses against Adversarial Attacks

Adversarial purification methods aim to remove adversarial perturbations from input images before feeding them to the model. These techniques have been extensively studied as a pre-processing defense mechanism. One of the earliest methods is JPEG compression, which reduces adversarial perturbations using image compression [42]. R&P [142] applies random resizing and padding to mitigate adversarial effects. Bit-Red [146] introduces feature squeezing techniques, such as bit-depth reduction and spatial smoothing. HGD [79] trains a denoising network using high-level supervision. FD [82] and ComDefend [60] are also based on image compression to construct defenses. NRP [90] employs self-supervised learning to train a robust denoising model. DiffPure [93] utilizes pre-trained diffusion models to effectively remove adversarial noise through iterative denoising. Although many of these defenses are vulnerable to adaptive attacks [2, 123], they are plug-and-play and model-agnostic, making them attractive for real-world applications. These purification methods can be easily integrated into existing systems without modifying the underlying neural network.

In addition to purification-based methods, Adversarial Training (AT) [86] is a widely adopted defense strategy that improves the robustness of DNNs by incorporating adversarial examples into the training process. Ensemble Adversarial Training [124] extends this method by utilizing multiple surrogate models, enhancing both robustness and resistance to transfer-based attacks. To reduce the high computational cost of standard adversarial training, Fast Adversarial Training [138] demonstrates that single-step attacks can be used to achieve competitive robustness with significantly lower training overhead. Numerous studies [94, 1, 63, 114, 154] have been proposed to enhance the effectiveness and efficiency of adversarial training. Another line of work focuses on certified defenses, with Randomized Smoothing (RS) [19] representing a major breakthrough. RS provides provable robustness guarantees against adversarial perturbations under the ℓ_2 norm, although extending these guarantees to ℓ_∞ norm remains more challenging. Recent work [13] shows that RS can be integrated with

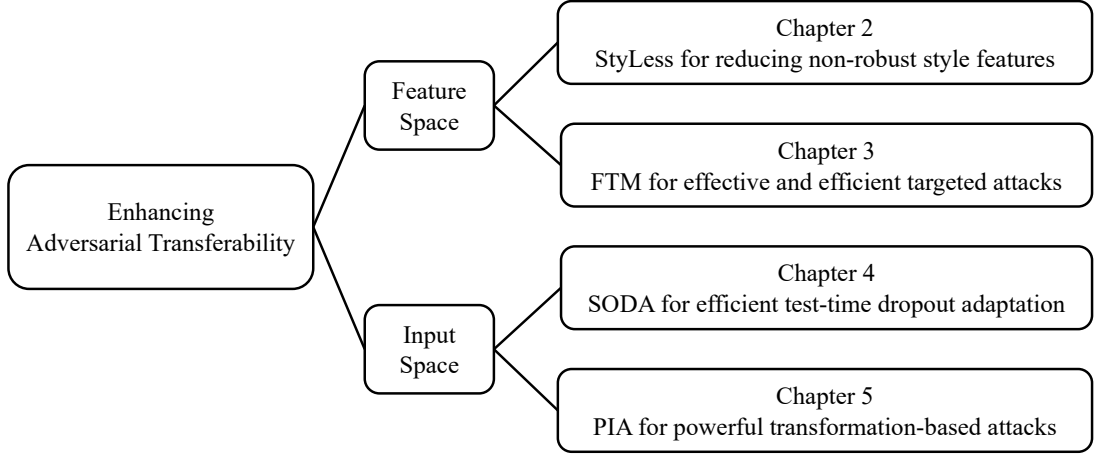


Figure 1.1: The contributions and organization of this thesis.

diffusion models to further improve certified robustness.

1.3 Contributions and Thesis Organization

Figure 1.1 illustrates the overall structure and main contributions of this thesis. All the proposed methods are developed under a unified objective: enhancing adversarial transferability. To this end, the thesis explores two complementary perspectives: feature space and input space. Specifically, Chapters 2 and 3 aim to improve transferability by manipulating internal representations in the feature space. In contrast, Chapters 4 and 5 address transferability enhancement in the input space, where adversarial examples are optimized through adaptive input transformations. The details are summarized as follows.

Chapter 2 presents a novel perspective on the mechanism of adversarial transferability and introduces Style-less Perturbations (StyLess), a method designed to enhance transferability. We investigate the role of image style in adversarial attacks and show that existing methods tend to increase the use of the style features of benign images during optimization, which hinders transferability. To address this, StyLess reduces

reliance on non-robust style features by employing stylized surrogate models, achieved by integrating adaptive instance normalization layers into vanilla surrogate models.

Chapter 3 addresses the challenging task of improving transferability in targeted attacks effectively and efficiently. Our analysis highlights the potential of feature-level augmentation for enhancing targeted attacks. However, existing methods largely ignore attack-specific augmentations. To bridge this gap, we propose Feature Tuning Mixup (FTM), a novel method that integrates attack-specific feature perturbations with clean features in the feature space. To optimize the learnable perturbations efficiently, we introduce a momentum-based stochastic update strategy. Extensive experiments demonstrate that FTM achieves state-of-the-art performance in targeted attacks while maintaining computational efficiency.

Chapter 4 explores adapting the dropout mechanism to the input space for improving adversarial transferability. While pixel-wise binary dropout is ineffective, a patch-wise soft mask with continuous values significantly enhances transferability. We propose Soft Dropout Adaptation (SODA), a unified framework that dynamically co-adapts a content-modulating soft mask and a context-expanding border prompt to enhance adversarial transferability. This test-time adaptation requires no additional training data and introduces only negligible computational overhead, making it a highly effective and practical method for transfer-based attacks.

Chapter 5 studies the limitations of existing input transformation-based attacks, which typically rely on fixed perspective transformations, thereby restricting adversarial transferability. To address this, we propose Perspective-Invariant Attack (PIA), a novel method that applies randomized perspective transformations during the attack process. By promoting invariance to viewpoint changes, PIA generates adversarial examples that are more transferable across different DNN architectures. In addition, PIA is computationally efficient and easily integrates into various attack frameworks, making it practical for real-world scenarios.

Chapter 6 concludes the thesis by summarizing the key contributions and discussing potential directions for future research.

Chapter 2

StyLess: Boosting the Transferability of Adversarial Examples

In this chapter, we analyze adversarial transferability from a novel perspective: original style features may act as non-robust features that hamper adversarial transferability. We identify that existing attacks’ inability to differentiate between style and content representations limits adversarial transferability. To address this, we propose Style-less Perturbations (StyLess), a novel attack method that improves transferability by using stylized surrogate models, constructed by inserting adaptive instance normalization layers into standard surrogate models. Unlike previous methods that rely on vanilla surrogate models and inadvertently exploit non-robust style cues, StyLess reduces this dependency without requiring model retraining. The inserted normalization layers are initialized through sampling and interpolation from reference style images without training. Experiments show that StyLess significantly improves adversarial transferability and outperforms existing transfer-based attacks.

2.1 Introduction

Although deep neural networks (DNNs) demonstrate impressive accuracy, they are notoriously susceptible to adversarial attacks [119, 40]. These attacks involve subtly modifying benign images to produce adversarial examples that appear unchanged to humans but mislead the model’s predictions. One notable property of adversarial examples is their transferability: a single adversarial input can often deceive multiple unseen, black-box models [95, 81, 22, 143], making them a practical threat in real-world applications.

It is well established that the style of an image can be separated from its core content, and existing style transfer methods enable the creation of stylized versions of images based on arbitrary reference styles [52]. The term “image style” captures the distinct visual traits of an image such as the textures, color palette, and lighting conditions. For example, two photographs of the same subject may exhibit different visual styles. Ideally, robust models should rely more heavily on content-based features rather than stylistic ones. This observation motivated us to explore adversarial transferability from the viewpoint of reducing reliance on non-robust features. We hypothesize that style features are inherently non-robust and hinder the generation of transferable adversarial examples. However, most existing adversarial attack methods treat content and style features within surrogate models indistinguishably, potentially limiting their transferability.

Our key idea is to enhance transferability by explicitly managing style features through stylized surrogate models. The unmodified surrogate model is termed the “vanilla model.” To generate its stylized counterpart, we insert an adaptive instance normalization (IN) layer into the vanilla model. The IN layer is initialized using the mean and variance of the original input features, preserving the model’s basic functionality without requiring retraining. During the attack, we perturb the IN layer’s mean and variance, effectively generating stylized feature representations. This perturbation

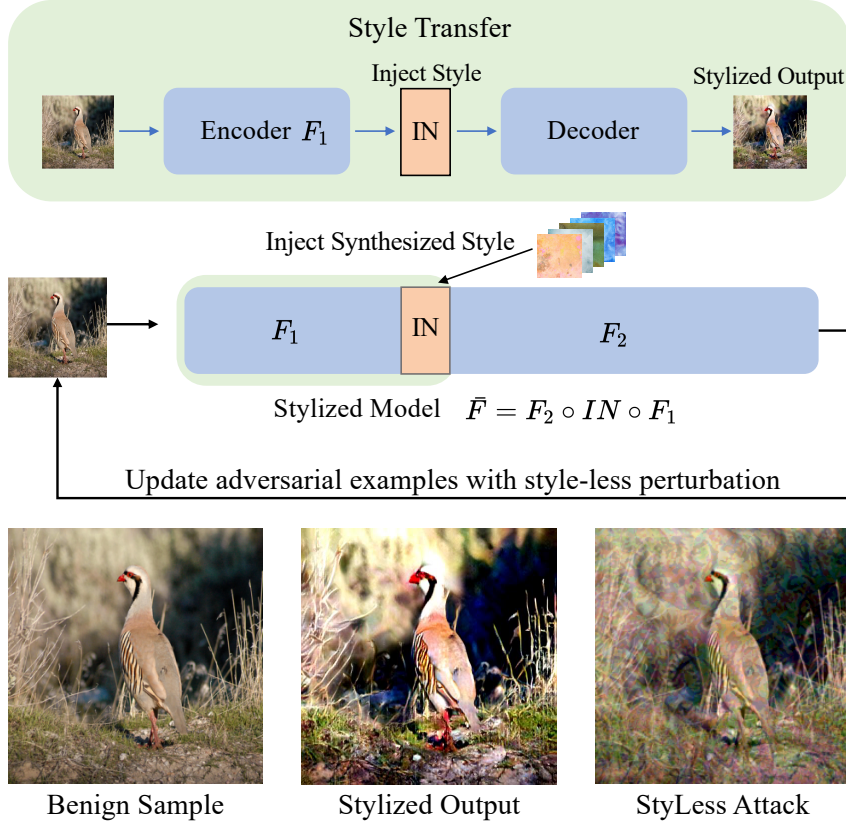


Figure 2.1: An overview of the proposed method. We construct stylized surrogate models by incorporating synthetic style features using an adaptive instance normalization (IN) layer, thereby reducing the surrogate’s dependence on non-robust style representations present in the original model.

can be driven by sampling or interpolating from external style reference images. By tuning the parameters of the IN layer, we can flexibly control the stylization of the surrogate model. To better understand the behavior of stylized versus vanilla surrogate models, we examined their loss dynamics during attack optimization. Interestingly, we observed that the loss in the vanilla model rose more rapidly than in the stylized models, which led to a growing gap. This indicates that conventional attacks tend to overly optimize the loss of the vanilla surrogate, inadvertently exploiting its style-related features. Given our hypothesis that style features are non-robust, we argue that this excessive reliance may undermine adversarial transferability. Therefore, to

promote transferability, it is necessary to mitigate the influence of non-robust style cues and reduce the gap between model variants.

Building on these insights, we introduce StyLess, a novel approach aimed at improving adversarial transferability. Our method leverages multiple synthetic style features to compete against the original style representations during optimization. This framework is visualized in Figure 2.1. We embed various stylized representations into the surrogate model using an IN layer, whose parameters are dynamically perturbed throughout the attack. Instead of relying solely on the vanilla model, we incorporate gradient information from both the vanilla and the stylized models to guide adversarial example updates. The shallow part of the model serves as a style feature encoder, while the inserted IN layer generates alternative style distributions in a training-free manner. Although it is possible to use a decoder to generate explicit stylized images, our method does not require this step. Experimental results confirm that StyLess substantially boosts the transferability of leading adversarial attacks against both unsecured and defense-equipped black-box DNNs.

2.2 Literature Review

Adversarial Attacks indicate the inherent weaknesses of modern DNNs [119]. Classic attack strategies are gradient-based, including methods such as the Fast Gradient Sign Method (FGSM) [40] and its iterative variant, I-FGSM [67]. C&W [14] introduced optimization-based attacks that minimize the distance between adversarial and benign samples. These attacks are not limited to the digital domain; adversarial examples have also been successfully demonstrated in the physical world [29, 107]. To counter such attacks, adversarial training has emerged as a widely adopted defense technique, wherein adversarial examples are incorporated into the training process to enhance model robustness [86].

Increasing Adversarial Transferability. A notable characteristic of adversarial examples is their transferability, the ability to deceive unseen models. Ensemble-based attacks [81] improve transferability by leveraging multiple surrogate networks instead of relying on a single one. Ghost networks [76] simulate diverse surrogates by perturbing structural components like skip connections and dropout layers. Several optimization techniques also aim to boost transfer: MI [22] employs momentum in the optimization process, whereas Variance Tuning (VT) [131] regulates gradient stability by introducing gradient variance. RAP [97] enhances transferability by generating adversarial examples situated in flat regions of the loss landscape. Data augmentation techniques further contribute: DI [143] applies transformations like resizing and padding; TI [23] shift pixels spatially; SI [80] computes gradients using scaled input variants. Admix [132] generates adversarial gradients by mixing target images with randomly sampled inputs during iterations.

Different architectures and latent representations affect adversarial vulnerability. The linearity of DNNs has been recognized as a contributing factor to this vulnerability [40]. To address this, LinBP [43] omits nonlinear activations during backpropagation. SGM [139] improves transferability by routing more gradients through residual connections. To better exploit intermediate representations, several methods have been proposed: training auxiliary classifiers on internal features [55, 56], maximizing the feature-space distance between natural and adversarial inputs [157], and refining adversarial examples at intermediate layers using ILA [51, 71].

Style Transfer aims to modify the visual style of an image to match that of another image [35, 61, 27]. Fast feedforward models can apply arbitrary styles in a single pass [52, 75]. This capability has found diverse applications in adversarial attacks: AdvCam [25] uses natural image styles to camouflage unrestricted adversarial perturbations, while FSA [145] generates adversarial examples with natural appearances via optimized style transformations. Style-based approaches have also been used to enhance robustness by enriching feature diversity [92]. Latent style variations can

aid in detecting adversarial inputs [130], and AMT-GAN [48] employs style transfer techniques like adversarial makeup to safeguard facial privacy.

Instance normalization (IN) [125] belongs to a broader family of normalization methods, including batch normalization (BN) [58], layer normalization [5], and group normalization [141]. These techniques primarily reduce internal covariate shift and facilitate faster training. Recently, their connection to robustness has attracted attention. For instance, BN has been shown to encourage the use of non-robust yet useful features [9, 54]. AdvBN [111] introduces additional BN layers during training to adversarially increase the loss and enhance robustness across domain shifts. Adjusting BN statistics such as running mean and variance at inference time has also been shown to improve resistance to adversarial and common corruptions [104, 8].

Among style-driven adversarial attacks, FSA [145] explicitly separates style and content features, which is conceptually related to our method. However, our approach significantly differs from FSA in three key aspects: 1) While FSA hides adversarial perturbations within optimized styles, our method aims to completely avoid dependence on style features. 2) FSA enhances the natural appearance of unrestricted (non- ℓ_p) attacks, whereas we aim to improve the transferability of ℓ_p -bounded adversarial images. 3) Though both methods are inspired by AdaIN [52], their use of IN layers is different. FSA applies perturbations to the IN layers to discover malicious image styles and employs a style decoder to generate the final output. In contrast, our method introduces randomized IN layers to augment attack diversity without any need for decoder training.

2.3 Methodology

2.3.1 Problem Formulation

Given an image x , transfer-based adversarial attacks aim to optimize an imperceptible perturbation using a surrogate model F . We formulate the objective function as follows:

$$\max_{\delta} \mathcal{L}(F(x + \delta), y) \quad \text{s.t.} \quad \|\delta\| \leq \epsilon, \quad (2.1)$$

where δ denotes the optimized perturbation that is added to x , y is the ground-truth label of x , $\mathcal{L}$ represents the attack loss, and ϵ is the maximum perturbation budget.

A widely used method to resolve the problem described above is I-FGSM [40, 67], which is formulated as:

$$x_{adv}^{t+1} = x_{adv}^t + \alpha \cdot \text{sign} \left(\nabla_x \mathcal{L} \left(F(x_{adv}^t), y \right) \right), \quad (2.2)$$

where α is the step size, and x_{adv}^{t+1} will be clipped to ensure $\|x_{adv}^{t+1} - x\| \leq \epsilon$.

Attacker capability. Consistent with prior work, we assume the attacker has access to a surrogate model and some test images but lacks knowledge of the target model’s architecture and training data. Notably, our method does not depend on any external or additional datasets. Instead, it leverages style representations, which can be obtained either from arbitrary source images or generated directly without requiring a reference style image.

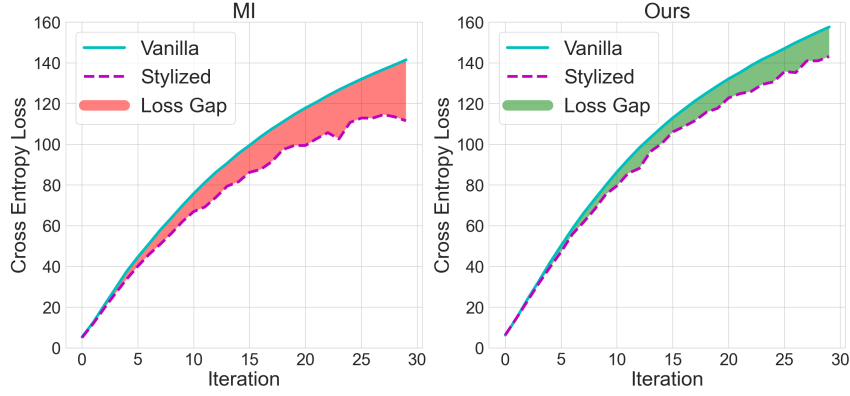
Transfer-based black-box attacks generate adversarial examples on a substitute model F and expect them to transfer and remain effective on unseen target models. This adversarial transferability enables their use in black-box scenarios, where direct access to the victim model is unavailable.

2.3.2 Motivation

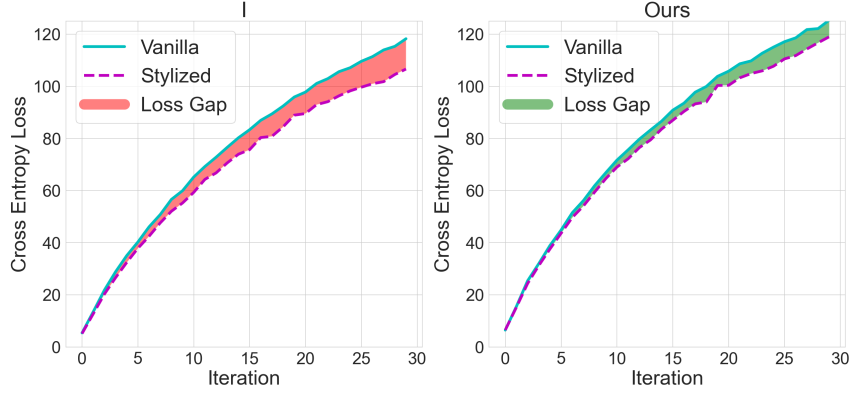
Existing transfer-based adversarial attacks typically depend on the gradient with respect to the loss $\mathcal{L}$ (Equation 2.2), but they often overlook the influence of individual components within $\mathcal{L}$. This oversight can be limiting, as improving transferability requires reducing reliance on non-robust features extracted by the surrogate model. Interestingly, in computer vision tasks, style features tend to be less robust than content features. Motivated by this, we propose to enhance adversarial transferability by explicitly suppressing the surrogate model’s dependence on style features within the loss function $\mathcal{L}$.

The core idea behind our method is to simulate multiple surrogate models that discard the original style characteristics of the original surrogate model, i.e., the vanilla model. We find that by introducing an instance normalization (IN) layer inside the vanilla model, we can generate new variants dubbed stylized surrogate models. For brevity, we sometimes drop the word “surrogate.” These stylized models allow for explicit control over style features while maintaining predictive accuracy. Our ultimate goal remains to craft adversarial examples that effectively deceive unseen black-box models, including the stylized ones. However, existing methods like MI and I-FGSM concentrate solely on increasing the loss of the vanilla model, ignoring stylized variants.

As shown in Figure 2.2, MI and I(-FGSM) exhibit poor transferability to stylized models. This is because the adversarial loss escalates much more rapidly in the vanilla model than in the stylized ones, leading to a growing loss gap. This imbalance indicates that the baseline attacks primarily exploit features unique to the vanilla model, but these features fail to generalize to the stylized variants. We proceed to show how the proposed approach overcomes this limitation by jointly increasing the adversarial loss for both the vanilla and stylized models, thereby substantially boosting transferability.



(a) Baseline attack MI-FGSM vs our MI+StyLess



(b) Baseline attack I(-FGSM) vs our I+StyLess

Figure 2.2: Stylized loss gap visualization highlighting the difference between vanilla and stylized networks. The surrogate model is RN50. A higher adversarial loss here corresponds to a higher attack success rate. The expanding gap in baseline methods indicates that their attack effectiveness on our stylized surrogate models lags behind that on the vanilla model. In contrast, our proposed method, StyLess, effectively narrows this gap.

2.3.3 Stylized Networks

In this section, we formally define our stylized surrogate models, which encode diverse style features by inserting an IN layer. Following this, we explore the stylized loss gap ($\Delta\mathcal{L}$), which quantifies the difference in loss values between the vanilla and

stylized surrogate models. We demonstrate that a growing $\Delta\mathcal{L}$ negatively affects transferability, and we introduce our method to reduce this gap.

Stylized Surrogate Models

Given a surrogate model $F = F_2 \circ F_1$, which is a classifier in this work, we can define a stylized model as follows:

$$\bar{F}_{x_s} = F_2 \circ \text{IN}_{x_s} \circ F_1, \quad (2.3)$$

where x_s is a reference style image, IN_{x_s} is an IN layer parameterized by x_s . Specifically, an IN layer is formulated as follows:

$$\text{IN}(x; \mu, \sigma) = \sigma \cdot \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \mu, \quad (2.4)$$

where μ and σ are the IN layer’s parameters, and $\mu(x), \sigma(x)$ refer to the mean and variance of the IN feature map when fed with input x . According to adaptive instance normalization (AdaIN) [52], to stylize an image x with a given reference style image x_s , it is sufficient to configure the IN layer as

$$\text{IN}_{x_s}(x) = \text{IN}(x) \big|_{\mu=\mu(x_s), \sigma=\sigma(x_s)} \quad (2.5)$$

The proposed stylized model $\bar{F}_{x_s}$ has encoded style features when we instantiate the IN layer using the above equation without training. Specifically, the F_1 from the classifier $F = F_2 \circ F_1$ serves as an encoder for style transfer, and given a style input x_s , we can obtain a stylized image via $\bar{x} = D \circ \text{IN}_{x_s} \circ F_1(x)$, where D denotes a decoder. Thus, $\text{IN}_{x_s} \circ F_1(x)$ has encoded the style features of x_s to $\bar{F}_{x_s}$. While this decoding process is feasible, it is not required during the attack phase of our method.

Stylized Loss Gap

In order to evaluate the effectiveness of using stylized models as surrogates, we introduce the concept of a *stylized loss gap*, defined as follows:

$$\Delta\mathcal{L} = \mathbb{E}_{x_s \in \mathcal{D}}[\mathcal{L}(F(x), y) - \mathcal{L}(\bar{F}_{x_s}(x), y)], \quad (2.6)$$

where F is the vanilla model, and $\bar{F}_{x_s}$ is the stylized one, x_s is a reference style image.

As shown in Figure 2.2, the loss gap $\Delta\mathcal{L}$ increases over time, which hinders the transferability of adversarial attacks. To better understand this, we can assume that the total adversarial loss $\mathcal{L}$ can be decomposed into content-dependent and style-dependent components:

$$\begin{aligned} \mathcal{L} &= \mathcal{L}^c + \mathcal{L}_x^s, \\ \bar{\mathcal{L}} &= \mathcal{L}^c + \mathcal{L}_{x_s}^s, \end{aligned} \quad (2.7)$$

where $\mathcal{L}$ and $\bar{\mathcal{L}}$ are the losses of the vanilla model and the stylized one, respectively; $\mathcal{L}^c$ is the common content-dependent loss, which captures the robust features shared by the vanilla and stylized models; $\mathcal{L}_x^s$ is the input image x -specific style-dependent loss, while $\mathcal{L}_{x_s}^s$ is the reference style image x_s -specific style-dependent loss. Here, the previously defined stylized loss gap $\Delta\mathcal{L} = \mathcal{L}_x^s - \mathcal{L}_{x_s}^s$.

$\mathcal{L}_x^s$ and $\mathcal{L}_{x_s}^s$ represent losses attributed to style features, while $\mathcal{L}^c$ captures content-based components that are likely to generalize across unseen models. The increasing gap observed with methods such as MI and I-FGSM implies that the optimization process gradually emphasizes $\mathcal{L}_x^s$, which is specific to the vanilla model F . As a result, this focus restricts the transferability of adversarial examples when applied to stylized surrogate models.

To reduce the stylized loss gap $\Delta\mathcal{L}$ and enhance adversarial transferability, we propose integrating $\mathcal{L}_{x_s}^s$ as a competing term for $\mathcal{L}_x^s$ during optimization to mitigate the increase of $\mathcal{L}_x^s$. It is natural to assume that all terms in Equation 2.7 are non-negative

and that the content-dependent loss $\mathcal{L}^c$ dominates, i.e., $\mathcal{L}^c \gg \mathcal{L}_x^s$, $\mathcal{L}^c \gg \mathcal{L}_{x_s}^s$. Furthermore, it is natural to consider that there exists a constant B bounding these losses: $\mathcal{L} < B$, $\bar{\mathcal{L}} < B$, since perturbations added to the input must remain within a norm bound ϵ . If we only optimize to maximize the loss of the vanilla model $\mathcal{L}$, both $\mathcal{L}^c$ and $\mathcal{L}_x^s$ may increase. To counteract this, we propose maximizing the composite loss: $\mathbb{E}_{x_s \in \mathcal{D}} \mathcal{L}_{x_s}^s + \mathcal{L}_x^s + \mathcal{L}^c$, which introduces a variety of style-dependent losses $\mathcal{L}_{x_s}^s (x_s \in \mathcal{D})$ to compete with $\mathcal{L}_x^s$, ultimately reducing the loss gap $\Delta\mathcal{L}$.

2.3.4 Our StyLess Attack

Motivated by the above observations, we introduce the Style-less Perturbations (StyLess) approach, which aims to enhance adversarial transferability by jointly optimizing both the stylized and vanilla loss terms:

$$\max_{\delta} \mathbb{E}_{x_s \in \mathcal{D}} \mathcal{L}(\bar{F}_{x_s}(x + \delta), y) + \mathcal{L}(F(x + \delta), y). \quad (2.8)$$

The core idea behind constructing diverse stylized models $\bar{F}_{x_s}$ lies in generating varied style statistics (μ, σ) for Equation 2.5, which parameterize IN layers without training. To simulate multiple style variations, we adopt a strategy based on scaling and interpolation, formulated as:

$$\begin{aligned} \mu &= \beta(\lambda\mu_x + (1 - \lambda)\mu_s), \\ \sigma &= \gamma(\lambda\sigma_x + (1 - \lambda)\sigma_s), \end{aligned} \quad (2.9)$$

where μ_x and σ_x denote the mean and variance of $F_1(x)$, corresponding to the style of the image x , while μ_s and σ_s represent the style statistics derived from an arbitrary style image x_s . The interpolation coefficient λ adjusts the blend between the two styles. The vectors β and γ , both of dimension c (the number of channels in the style representation), are used to scale the synthesized style features. We present a summary of our StyLess attack in Algorithm 1.

Algorithm 1 Proposed StyLess Attack

Input: A white-box model F , a victim image x , iteration number T , maximum perturbation ϵ , input augmentation method $\phi(\cdot)$, MI decay factor η .

Input: StyLess' scaling factors β, γ and interpolation rate λ . The total number of stylized models in StyLess N .

Output: An adversarial example x_{adv}

```

1:  $x_{adv} = x, g_0 = 0, \alpha = \epsilon/2$ .
2: for  $t = 0 \rightarrow T - 1$  do:
3:   Apply data augmentation  $x_{adv} = \phi(x_{adv})$ 
4:   Compute the gradient of input  $\tilde{g}_{t+1}$  with respect to  $x_{adv}$  using  $F$ 
5:   repeat
6:     Generate a synthesized style statistic by Equation 2.9
7:     Build a stylized model  $\bar{F}_{x_s}$  by Equation 2.3 and 2.5
8:     Obtain gradient  $\tilde{g}$  of  $x_{adv}$  using  $\bar{F}_{x_s}$ 
9:     Update gradient using multiple surrogate models  $\tilde{g}_{t+1} = \tilde{g}_{t+1} + \tilde{g}$ 
10:  until  $N$  stylized models are generated
11:  Calculate gradient momentum  $g_{t+1} = \eta \cdot g_t + \tilde{g}_{t+1}/\|\tilde{g}_{t+1}\|_1$ 
12:  Update adversarial image  $x_{adv} = x_{adv} + \alpha \cdot \text{sign}(g_{t+1})$ 
13: end for
14: return  $x_{adv}$ .
```

2.4 Experiments

2.4.1 Experimental Setup

Dataset. We conduct our experiments using the ImageNet dataset [101]. Specifically, we adopt 1,000 images from the validation set of ImageNet, which is randomly sampled by VT [131]. This configuration is commonly adopted in prior research [131, 22, 80, 143, 139, 43].

Models. The effectiveness of current adversarial attacks is evaluated across various black-box DNN models, which include both standard (unsecured) and adversarially trained (secured) models. The standard models include VGG19 [112], AlexNet [66], RN50 [44], WRN101 [149], DN121 [49], IncV3 [118], MnasNet [121], MNv2 [103], SNv2 [85], and ViT [24]. All models utilize pretrained weights provided by the official PyTorch library. The secured models include IncV3_{ens3} (an ensemble of three InceptionV3 models), IncV3_{ens4} (ensemble of four InceptionV3 models), and IncResV2_{ens3} (ensemble of three IncResV2 networks). These models have undergone adversarial training and are widely adopted in adversarial robustness research [22, 139, 131, 81]. For surrogate models, we use four commonly used architectures: VGG19 [112], RN50 [44], WRN101 [149], and DN121 [49].

Implementation Details of StyLess.

I-FGSM [40] is used as the baseline attack. All attacks are non-targeted and constrained under the l_∞ norm. The perturbation magnitude is $\epsilon = 16/255$. StyLess is compared with six representative transferable attack methods: Momentum Iterative (MI) [22], DI [143], TI [23], SI [80], Admix (AI) [132], and a standard ensemble-based approach [81]. The learning rate is set to $\alpha = \epsilon/2$, and the attack iterations T are fixed at 50. For MI, the decay factor is $\mu = 1$. For StyLess, we simulate $N = 10$ stylized models per iteration. Regarding generating a stylized surrogate model for a specific input x , we randomly sample λ from the interval $[0, 0.2]$, and β, γ from $[0, 2]$. We ensure that the stylized model output $\bar{F}_{x_s}(x)$ matches either the vanilla model output $F(x)$ or the ground-truth label. The IN layer is inserted at different locations based on the network architecture: after the initial bottleneck block in RN50 and WRN101, and the first dense block in DN121.

2.4.2 Attacking Unsecured Models

We conduct a comprehensive comparison between StyLess and existing attack methods on various unsecured DNN models, using three different surrogate models. As shown in Table 2.1, StyLess serves as a powerful and general-purpose enhancement technique that can be effectively integrated with standard attack methods to significantly improve their transferability. More specifically, we evaluate StyLess in conjunction with six baseline methods: I-FGSM (I), Momentum Iterative (MI), Diverse Input (DI), Translation-Invariant (TI), Scale-Invariant (SI), and Admix (AI). In the hardest setting, attacking the black-box IncV3 using RN50 as the surrogate model, StyLess offers substantial gains in attack success rate: from 46.2% to 68.3% for I-FGSM, and from 59.2% to 78.9% for MI. StyLess also maintains strong performance across different surrogate models. For instance, in the DN121 $\Rightarrow$ ShuffleNetV2 (SNv2) setting, StyLess significantly enhances the success rate over the baseline: from 69.3% to 91.4% for I-FGSM, and from 77.4% to 95.1% for MI.

In addition to being used independently, StyLess can be incorporated into more complex, combination-based attacks. Prior research has demonstrated that fusing different attack strategies often leads to more effective and transferable adversarial examples. StyLess can further augment such combinations, providing additional performance gains. We evaluate StyLess in combination with four compound attack variants: MDI, MTDI, MTDSI, and MTDAI. In the RN50 $\Rightarrow$ IncV3 setting, StyLess improves the success rates of these methods by +8.0%, +8.2%, +3.0%, and +3.3%, respectively. We also assess performance using various surrogate architectures, including RN50, WRN101, and DN121. For example, when integrating StyLess with MTDAI, we observe consistent improvements in multiple transfer scenarios: WRN101 $\Rightarrow$ IncV3: +4.7%, WRN101 $\Rightarrow$ MNv2: +1.5%, WRN101 $\Rightarrow$ SNv2: +3.4%, DN121 $\Rightarrow$ IncV3: +4.3%, DN121 $\Rightarrow$ MNv2: +0.6%, DN121 $\Rightarrow$ SNv2: +3.8%. These results demonstrate that StyLess is both an effective and flexible framework for boosting the transferability of adversarial attacks across a wide range of models and settings.

Table 2.1: Attacking unsecured DNN models using StyLess and existing attack methods.

Source	Attack	VGG19	RN50	WRN101	DN121	IncV3	MNV2	SNv2
RN50	I / +Ours	72.8 / 88.8	100 / 100	80.8 / 97.1	83.0 / 97.8	46.2 / 68.3	77.1 / 91.9	60.6 / 77.8
	MI / +Ours	82.9 / 94.1	100 / 100	83.9 / 97.2	87.5 / 98.7	59.2 / 78.9	83.8 / 93.2	72.3 / 83.5
	MDI / +Ours	97.5 / 99.2	100 / 100	98.2 / 99.8	99.4 / 100	89.5 / 97.5	98.1 / 99.9	88.5 / 96.9
	MTDI / +Ours	98.6 / 99.7	100 / 100	99.2 / 100	99.8 / 100	90.2 / 98.4	98.7 / 100	90.8 / 98.1
	MTDSI / +Ours	98.6 / 99.2	100 / 100	99.5 / 100	99.8 / 100	96.2 / 99.2	98.7 / 99.9	96.2 / 98.2
	MTDAI / +Ours	99.3 / 99.6	100 / 100	99.8 / 100	99.9 / 100	95.5 / 98.8	99.7 / 100	95.7 / 99.0
WRN101	I / +Ours	64.8 / 79.6	88.6 / 98.7	100 / 100	77.7 / 94.2	43.2 / 66.6	68.1 / 84.7	58.1 / 73.2
	MI / +Ours	75.6 / 86.4	89.8 / 98.7	100 / 100	83.9 / 96.4	57.6 / 73.7	76.1 / 87.0	70.2 / 81.2
	MDI / +Ours	92.2 / 98.5	99.0 / 99.9	100 / 100	97.1 / 99.9	86.3 / 96.3	93.6 / 99.0	86.8 / 95.8
	MTDI / +Ours	92.5 / 98.6	99.2 / 100	100 / 100	97.9 / 99.6	89.2 / 97.9	95.4 / 99.3	88.4 / 97.0
	MTDSI / +Ours	95.9 / 98.9	99.6 / 100	100 / 100	99.7 / 100	95.8 / 99.5	97.7 / 99.9	94.5 / 98.4
	MTDAI / +Ours	96.5 / 99.2	99.9 / 99.9	100 / 100	99.4 / 100	93.7 / 98.4	98.1 / 99.6	94.6 / 98.0
DN121	I / +Ours	79.7 / 97.7	87.0 / 99.4	74.7 / 98.0	100 / 100	55.7 / 88.7	81.3 / 97.0	69.3 / 91.4
	MI / +Ours	86.9 / 99.1	89.4 / 99.7	78.2 / 98.6	100 / 100	66.0 / 95.7	86.5 / 98.7	77.4 / 95.1
	MDI / +Ours	96.7 / 99.9	98.9 / 100	95.4 / 99.8	100 / 100	90.5 / 99.0	97.5 / 100	90.9 / 98.6
	MTDI / +Ours	97.5 / 99.9	99.1 / 100	95.9 / 99.8	100 / 100	92.7 / 99.4	96.9 / 100	92.2 / 98.6
	MTDSI / +Ours	97.4 / 99.9	99.3 / 100	98.0 / 99.8	100 / 100	96.5 / 99.8	98.9 / 100	95.1 / 99.1
	MTDAI / +Ours	99.2 / 99.9	99.8 / 100	98.1 / 99.9	100 / 100	95.1 / 99.4	99.4 / 100	95.2 / 99.0

2.4.3 Attacking Secured Models

We assess the effectiveness of StyLess on three widely adopted secured models: $\text{IncV3}_{\text{ens3}}$, $\text{IncV3}_{\text{ens4}}$ and $\text{IncResV2}_{\text{ens}}$, as shown in Table 2.2. We report the attack success rates of I, MI, MDI, MTDI, and MTDSI against widely-used defended models using two surrogate models: RN50 and WRN101. Compared to the unsecured DNNs discussed earlier, these secured models exhibit significantly stronger robustness. Taking $\text{RN50} \Rightarrow \text{IncV3}_{\text{ens3}}$ transfer attack as an example, baseline methods I, MI, MDI, and MTDI obtain attack success rates of 21.6%, 31.5%, 59.6%, and 69.2%, respectively. StyLess substantially improves the success rates across all these methods. In particular, when applied to enhance MTDSI, StyLess achieves notable improvements: $91.6\% \rightarrow 97.5\%$ in $\text{WRN101} \Rightarrow \text{IncV3}_{\text{ens3}}$ attack; $88.6\% \rightarrow 96.1\%$ in $\text{WRN101} \Rightarrow \text{IncV3}_{\text{ens4}}$ attack; $83.0\% \rightarrow 92.5\%$ in $\text{WRN101} \Rightarrow \text{IncResV2}_{\text{ens}}$ attack. These results demonstrate the remarkable ability of StyLess to boost transferability against strong, adversarially trained models.

Figure 2.3 provides a comparative analysis between StyLess and LinBP. We also include combinations of LinBP with other established attack methods such as ILA, SGM, and MDI, applied to various target DNNs. StyLess consistently outperforms all these methods, achieving the highest transfer success rates across the board.

2.4.4 Evaluating with Ensemble-Based Attacks

Constructing adversarial examples against various surrogate models simultaneously has been proven effective in improving adversarial transferability in practice [81]. This ensemble-based strategy is known to be a strong attack approach, and our proposed StyLess method can further enhance its effectiveness. In our experiments, we construct an ensemble surrogate by integrating RN50, WRN101, and DN121.

StyLess is compared with MI and MTDI based on the ensemble-based attack frame-

Table 2.2: Attacking secured DNN models with StyLess and existing attack methods. RN50 or WRN101 is used as the surrogate model.

Attack	<u>RN50</u> $\Rightarrow$		
	IncV3 _{ens3}	IncV3 _{ens4}	IncResV2 _{ens}
I / +Ours	21.6 / 34.6	18.9 / 32.0	14.1 / 21.5
MI / +Ours	31.5 / 47.1	29.3 / 42.4	20.8 / 31.0
MDI / +Ours	59.6 / 78.1	53.5 / 69.8	38.3 / 57.3
MTDI / +Ours	69.2 / 89.6	63.8 / 81.8	54.6 / 72.2
MTDSI / +Ours	88.0 / 93.1	84.7 / 91.3	77.8 / 84.5
	<u>WRN101</u> $\Rightarrow$		
I / +Ours	23.4 / 40.3	20.9 / 34.6	15.5 / 25.6
MI / +Ours	35.0 / 51.2	30.4 / 46.6	24.7 / 35.9
MDI / +Ours	64.0 / 81.5	58.8 / 76.2	46.9 / 66.4
MTDI / +Ours	75.5 / 91.4	70.5 / 87.4	63.6 / 81.0
MTDSI / +Ours	91.6 / 97.5	88.6 / 96.1	83.0 / 92.5

work in Table 2.3. The power of ensemble-based attacks becomes evident when comparing these results to those in Table 2.1 and Table 2.2. For instance, in the RN50 $\Rightarrow$ IncV3 transfer setting, MI achieves a success rate of 59.2%, whereas combining RN50, WRN101, and DN121 as surrogates raises the MI success rate to 81.0%. All results are reported under a standard perturbation constraint of $\epsilon = 16$. Despite the strength of ensemble-based methods, even the MTDI attack achieves an average success rate below 90% when attacking the these secured networks: IncV3_{ens3}, IncV3_{ens4}, and IncResV2_{ens}. However, integrating StyLess with MTDI leads to substantial improvements in adversarial transferability: 92.7% $\rightarrow$ 98.5% for IncV3_{ens3}, 86.4% $\rightarrow$ 97.5% for IncV3_{ens4}, 80.7% $\rightarrow$ 94.3% for IncResV2_{ens}. These results suggest that StyLess is a new type of method, and can integrates seamlessly with ensemble-based approaches.

We also evaluate attack performance under smaller perturbation budgets ($\epsilon = 4$ and

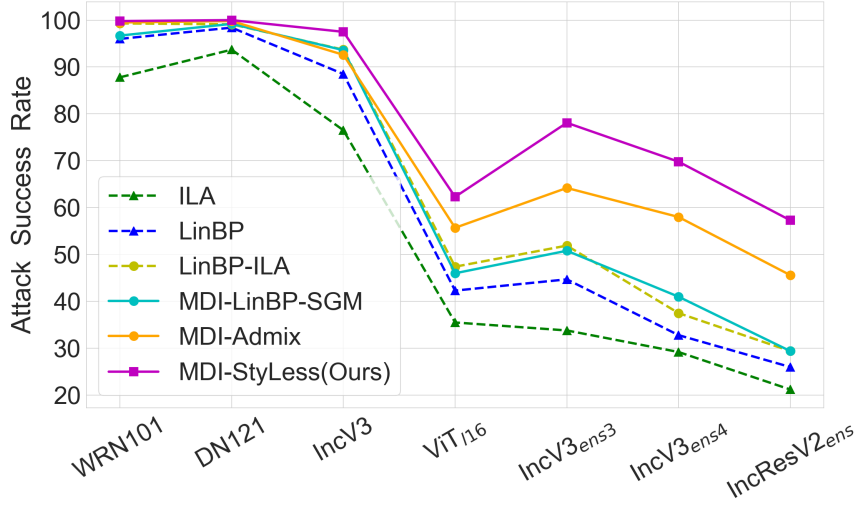


Figure 2.3: Comparing with other attack methods on various DNN models and defensive methods. RN50 is used as the surrogate model.

$\epsilon = 8$), which represent more challenging scenarios. For instance, in the ensemble-based MTDI $\Rightarrow$ IncV3_{ens3} case, reducing ϵ from 16 to 8 leads to a performance drop from 92.7% to 67.6%. StyLess successfully mitigates this degradation, boosting performance by +17.5% (from 67.6% to 85.1%). Even with $\epsilon = 4$, StyLess continues to demonstrate noticeable gains. These findings confirm that StyLess maintains effective transferability even under more restrictive perturbation constraints and against stronger, more resilient models.

2.4.5 Real-World Evaluation

We evaluate the transferability of adversarial attacks in a real-world setting using the Google Cloud Vision API as a representative example. This API provides access to a range of computer vision services such as face detection and image labeling. Its internal components, such as model architecture, training data, and defense mechanisms, are not accessible to ordinary users. In this section, we use it as a black-box model and specifically target its image labeling functionality.

ϵ	Attack	RN50+WRN101+DN121 $\Rightarrow$									
		VGG19	RN50	WRN101	DN121	IncV3	MNv2	SNv2	IncV3 _{ens3}	IncV3 _{ens4}	IncResV2 _{ens}
4	MI	56.8	96.4	94.8	97.2	35.7	62.3	48.3	18.4	15.8	9.4
	MTDI	79.2	97.9	96.6	98.8	67.5	84.5	71.6	36.5	32.1	22.3
	+StyLess	90.4	99.5	99.4	99.9	83.2	94.3	85.4	50.0	42.9	29.1
8	MI	79.2	99.6	98.7	99.4	58.3	83.3	68.0	31.4	26.8	18.2
	MTDI	95.2	99.9	99.5	99.9	89.5	97.5	90.4	67.6	61.8	46.9
	+StyLess	99.1	100	100	100	97.5	99.6	97.2	85.1	77.4	66.1
16	MI	92.3	100	100	100	81.0	93.2	85.0	53.0	46.3	35.0
	MTDI	99.2	100	100	100	97.5	100	97.9	92.7	86.4	80.7
	+StyLess	100	100	100	100	99.8	100	100	98.5	97.5	94.3

Table 2.3: Combining with model ensemble attack methods.

To interact with this service, users upload input images and receive predicted labels returned by the API. We follow the experimental setup described earlier and select 1,000 images from the ImageNet validation set as our attack targets. Since the API does not support all 1,000 ImageNet categories, the goal of the attack is to change the API’s top-1 prediction.

For this evaluation, we adopt RN50 as the surrogate model and compare the performance of the baseline attack MTDSI with our enhanced variant. Results indicate that MTDSI obtains a success rate of 75.6%. In contrast, our MTDSI-StyLess significantly improves the success rate to 85.2%, yielding a +9.6% gain over MTDSI. These findings highlight the practical threat posed by adversarial transferability in real-world scenarios. Moreover, they show the effectiveness of StyLess in enhancing transferability under real-world, restricted-access conditions.

2.4.6 Ablation Study

In this section, we conduct four ablation studies to evaluate the impact of different components in our proposed method: 1) the insertion position of the instance normalization (IN) layer; 2) the number of stylized surrogate models generated; 3) the

relationship between clean losses of stylized surrogate models and adversarial transferability; 4) the most influential statistical component among the style features.

The insertion position of the IN layer. As illustrated in Figure 2.4, we explore the impact of injecting synthesized style features at various layers of the vanilla surrogate models, which serve as the encoder F_1 in our method. We conduct experiments using two surrogate networks: WRN101 and DN121. Synthesized styles are injected at different depths of the networks, and we report the resulting attack success rates on a variety of black-box models. Our observations reveal a consistent trend: the highest success rates are typically achieved when the IN layer is placed in the early (shallow) layers of the surrogate network. For instance, when WRN101 is used as the surrogate model, injecting style features before layer 10 yields strong results. While inserting styles into mid-level layers (e.g., layers 40 to 80) can still be effective, it often leads to performance instability. Injection into the final layers generally results in degraded attack performance. A similar pattern is observed for DN121: the shallow layers prove to be the most effective insertion points for the IN layer, whereas the final layers consistently yield the weakest performance.

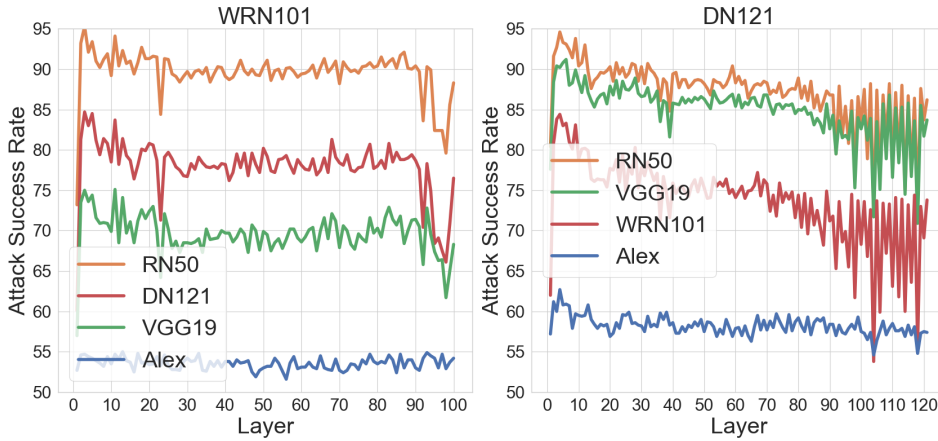


Figure 2.4: Ablation study on the insertion position of the IN layer. WRN101 or DN121 is used as the surrogate.

The number of stylized surrogate models generated. In Figure 2.5, we ex-

amine the effect of varying the number of stylized surrogates simulated per iteration, ranging from 0 to 10. To evaluate this, we combine StyLess with two baseline attack methods: MI and DI. The left subplot presents the attack success rates of the MI+StyLess combination on three unsecured models, while the right subplot illustrates the performance of DI+StyLess evaluated on adversarially trained models. If the number of stylized surrogates is set to zero, StyLess is not activated, and the performance corresponds to the vanilla baseline attack. As observed, increasing the number from 0 to 1 leads to an immediate improvement in attack success rate, indicating that StyLess begins to contribute positively. An interesting anomaly is observed in the DI+StyLess setting: for IncResV2_{ens} , when fewer than three stylized models are used, the success rate slightly decreases compared to the baseline (around 25%). However, once the number exceeds three, the attack success rate surpasses the baseline. This behavior may be attributed to the high robustness of IncResV2_{ens} , which likely requires a greater variety of synthesized style features to sufficiently diversify the surrogate representation. Overall, the results suggest that StyLess achieves stable and notable improvements when six to ten stylized models are employed per iteration.

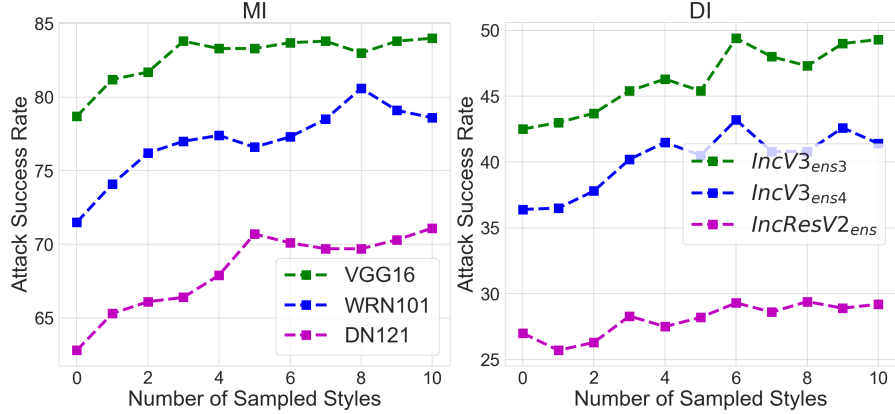


Figure 2.5: Impact of the number of stylized models, using RN50 as the surrogate.

The relationship between clean losses of stylized models and transferability. In Figure 2.6, we analyze how the strength of style injection affects the surrogate model’s loss, and consequently, the attack success rate. This analysis is conducted

using VGG19 as the surrogate model. The strength of injected style features is indicated by the number of stars, where more stars represent stronger style modification effects. In principle, the incorporation of stylized features should preserve the surrogate model’s clean loss, as elevated loss values often correlate with degraded classification accuracy. When the stylized surrogate becomes overly corrupted, the resulting gradients may be detrimental to attack optimization. Thus, it is crucial to maintain the clean loss below a certain upper bound when generating stylized variants. In the figure, the red line corresponds to scenarios where the stylized model becomes overly distorted, that is, the clean loss exceeds a critical threshold (represented by the yellow line in the right subplot). Under such conditions, the attack success rate drops sharply. This highlights the importance of preserving reasonable clean performance to ensure the effectiveness of stylized models in transfer-based attacks.

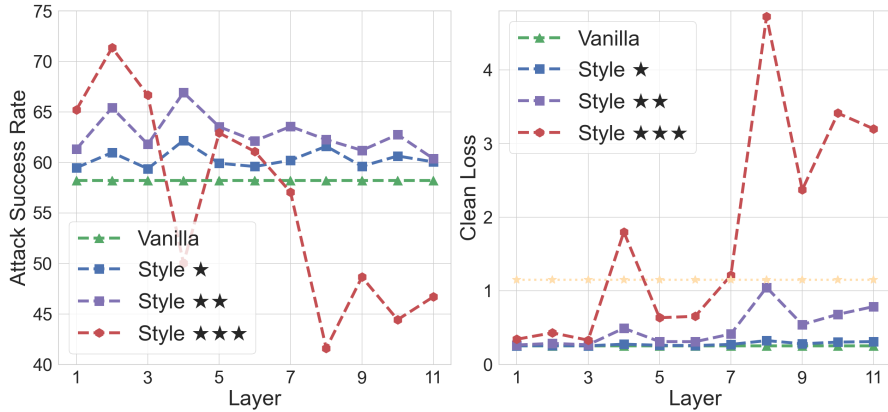


Figure 2.6: Impact of synthesized style feature variation on clean loss and StyLess success rate. Star count indicates the degree of style modification.

The most influential statistical component among the style features. In Figure 2.7, we investigate the relative impact of the mean and variance of style features on the performance of StyLess. In this setting, the interpolation factor is fixed at $\lambda = 0$. According to Equation 2.9, the parameter β controls the mean in the IN layer, while γ determines the variance. We report the average success rate across five target models: VGG19, AlexNet, RN50, WRN101, and DN121. The results in-

dicating that γ has a significantly greater influence on attack performance than β . For example, when using RN50 as the surrogate model, modifying β alone produces negligible improvement over the baseline, whereas adjusting γ alone boosts the success rate by approximately 5%. A similar trend is observed when VGG19 serves as the surrogate model. This observation aligns with the theoretical understanding of gradient backpropagation through an IN layer. The partial derivative $\frac{\partial}{\partial x} \text{IN} = \sigma$ suggests that variance (controlled by γ) contributes more critically to gradient flow, thereby exerting a stronger influence on adversarial optimization.

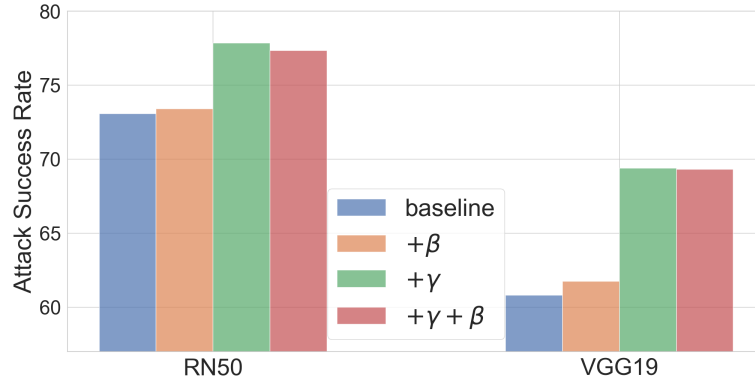


Figure 2.7: Impact of different style statistics.

2.5 Summary

In this work, we investigate the mechanism of adversarial transferability in terms of image styles. Our investigation reveals that existing attack methods tend to increasingly exploit the style characteristics of the surrogate model during their attack process, which in turn degrades their transferability. To mitigate this limitation, we introduce StyLess, a novel attack strategy that improves transferability by explicitly reducing reliance on the surrogate model’s original style features. Instead of relying solely on a vanilla surrogate model, StyLess constructs and leverages stylized

surrogate models to generate adversarial examples. Extensive experiments confirm that StyLess significantly outperforms existing transfer-based attack methods across a variety of settings. It is highly compatible with other attack techniques and can be seamlessly integrated to further enhance their effectiveness. Importantly, StyLess represents a distinct paradigm from traditional adversarial attacks, and we hope this work will inspire deeper exploration into the role of feature representations.

Chapter 3

Improving Transferable Targeted Attacks with Feature Tuning Mixup

In this chapter, we investigate the challenging problem of transferable targeted attacks, which aim to craft adversarial examples that mislead DNNs into predicting predefined target classes. Although a variety of methods have been introduced to improve the effectiveness of targeted attacks, these methods often suffer from high computational overhead and limited transferability. To address these challenges, we propose a novel method called Feature Tuning Mixup (FTM), which enhances the transferability of targeted attacks by injecting a combination of random and optimized noise directly into the feature space. FTM introduces learnable feature perturbations and utilizes an efficient stochastic update strategy to optimize them. These learnable perturbations enable the generation of stronger adversarial examples that generalize better to unseen models. Moreover, we demonstrate that attack performance can be further improved by ensembling multiple surrogate models, each perturbed with independently optimized FTM perturbations. Comprehensive experiments show that

FTM significantly outperforms existing state-of-the-art methods for targeted transfer attacks while maintaining a low computational cost.

3.1 Introduction

Deep neural networks (DNNs) have been shown to be highly susceptible to adversarial attacks [40, 67, 86], raising serious concerns regarding their security and reliability. In image classification tasks, these attacks operate by introducing imperceptible perturbations to benign inputs, thereby causing DNNs to produce incorrect predictions. Even more concerningly, prior studies [22, 143, 23] have revealed the transferability of adversarial examples: adversarial inputs crafted using one white-box model can effectively deceive other black-box models without access to their internal architectures or parameters. This cross-model generalization significantly amplifies the real-world threat posed by adversarial attacks, particularly in black-box or limited-access deployment scenarios. As a result, understanding and addressing the transferability of adversarial examples is essential for improving the robustness of DNNs and safeguarding AI systems against adversarial vulnerabilities.

Targeted transfer-based attacks aim to mislead models into predicting specific target classes, which remains a significant challenge. Due to the lack of knowledge about the target model’s training data and architecture, such attacks tend to generate targeted adversarial examples that overfit to surrogate models and fail to transfer to target models. Various transferable targeted attacks [11, 156, 137] have been proposed but still leave much room for improvement. Notably, data augmentation methods [131, 80, 132] are commonly employed for targeted attacks. But they primarily operate in the image space, while augmentation methods at the feature level remain relatively unexplored. Recently, the Clean Feature Mixup (CFM) method [12] has made important progress by using clean features from random images to disrupt adversarial examples, leading to better targeted attack success rates (Figure 3.1).

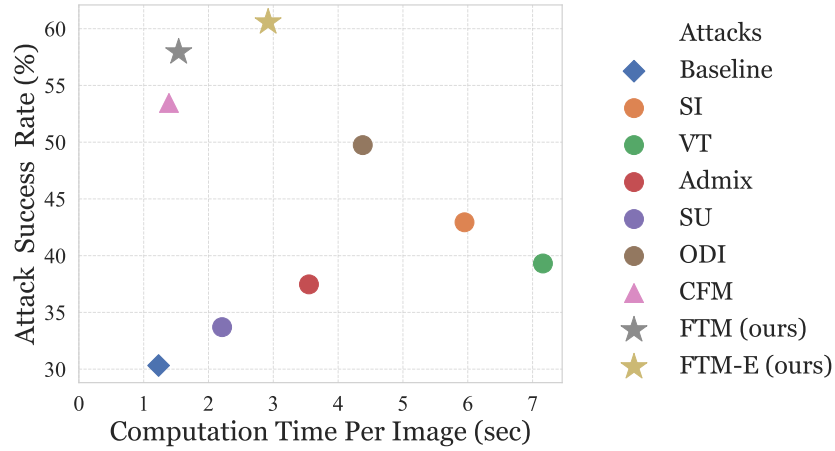


Figure 3.1: **Effectiveness and efficiency of targeted attacks.** Average attack success rates on 14 black-box models, along with the computation time required to generate an adversarial example. Our methods demonstrate superior performance with low computational cost, surpassing state-of-the-art targeted attacks.

However, CFM uses random clean features without optimizing them specifically for disrupting adversarial examples. We hypothesize that carefully optimized perturbations would be more effective in disrupting adversarial examples, potentially yielding substantial improvements in attack transferability.

In this work, we propose Feature Tuning Mixup (FTM), a novel attack method that optimizes attack-specific feature perturbations to enhance the transferability of targeted adversarial examples. Our method fine-tunes the intermediate features of surrogate models during the attack process by introducing learnable feature perturbations. These feature perturbations are optimized by mixing them with clean features from random images to counteract the adversarial effects of targeted adversarial examples. Through this mechanism, adversarial examples are gradually compelled to become more robust, allowing them to overcome these feature-level disturbances. To address the computational overhead of optimizing feature perturbations, we propose a stochastic update strategy. This strategy initializes the learnable perturbations using values from previous attack iterations and selectively optimizes only a small subset

of feature perturbations in each iteration. Ultimately, FTM significantly improves the transferability of targeted adversarial examples while maintaining computational efficiency.

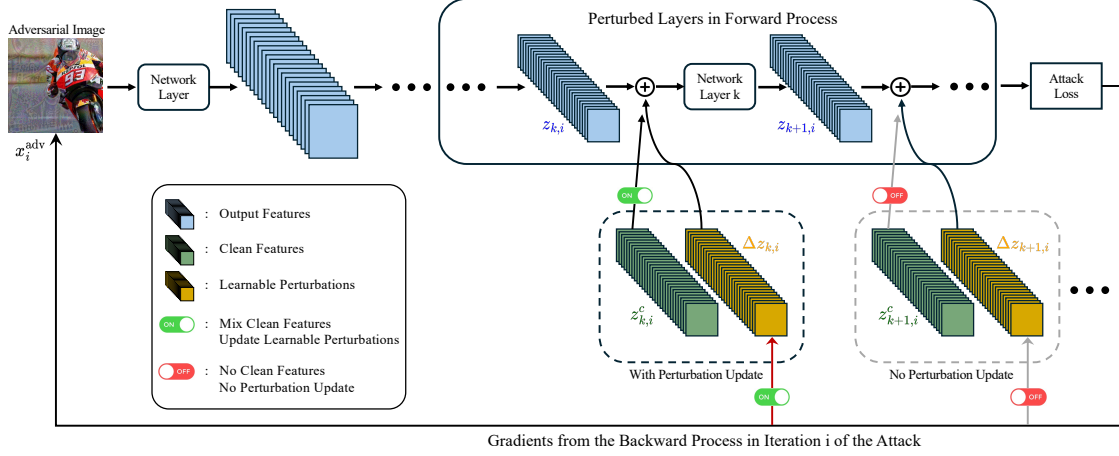


Figure 3.2: Overview of our FTM attack. In the forward pass, the learnable perturbations are added to the output features. Only a small portion of network layers are randomly selected for perturbation updates and clean feature mixup in each attack iteration. In the backward pass, the gradients of the selected perturbations and adversarial image are computed together for update.

We provide an overview of the proposed FTM in Figure 3.2. FTM first introduces learnable perturbations into the deep layers of the surrogate model. The learnable perturbations are added element-wise to the original output features before passing to the next layer. In each attack iteration, a small subset of layers is randomly selected and combined with random clean features. Finally, we compute gradients for the learnable perturbations and adversarial image using only one forward and backward pass, similar to standard attacks. We update the selected learnable perturbations and adversarial image, while unselected perturbations remain unchanged. As a result, FTM fully leverages optimized attack-specific feature perturbations, prevents overfitting to surrogate models, and significantly enhances the transferability of targeted attacks.

3.2 Literature Review

Adversarial attacks [14, 29, 54, 40, 86] have been extensively explored within the field of adversarial machine learning, including unrestricted attacks [20], physical attacks [73, 74], and black-box attacks [10, 78, 22]. Notably, black-box attacks can be categorized into query-based [10] and transfer-based attacks [22, 78]. The latter, which does not require access to the target DNN model’s outputs, parameters or architecture, is more practical and is the focus of this paper.

Non-targeted Transfer-based Attacks. Current non-targeted transfer attacks predominantly rely on I-FGSM [67]. This method iteratively calculates the gradient of the classification loss with respect to the input and updates the adversarial example in the direction that increases the loss maximally. However, it often leads to overfitting on white-box models, making the adversarial examples less effective against other black-box models. Numerous studies have explored solutions to the overfitting issue to enhance adversarial transferability. MI-FGSM [22] incorporates a momentum-based optimization method during iterative attacks. Translation-Invariant (TI-FGSM) method [23] enhances the transferability of adversarial examples by achieving translation invariance through gradient smoothing. Data augmentation techniques are widely used to mitigate overfitting. The Diverse Inputs (DI) method [143] employs random resizing and padding with a probability p to generate diverse input, while the Resized Diverse Inputs (RDI) method [160] fixes $p = 1$ and rescales DI-enhanced images back to their original size. The Scale-Invariant (SI) method [80] generates adversarial perturbations using multiple scale transformations, and Admix [132] extends SI by blending images from other labels into the input. Variance Tuning (VT) method [131] randomly selects data points around a given data point and computes their gradients to avoid overfitting. These methods operate in the image space. Feature-level approaches have also been studied to improve adversarial transferability. For instance, the Feature Importance-aware Attack (FIA) [134]

identifies and perturbs important features in the intermediate layers of a surrogate model. The Skip Gradient Method (SGM) [139] utilizes more gradients from skip connections. Ensemble-based attacks [81, 22] utilize multiple surrogate models to generate adversarial examples.

Targeted Transfer-based Attacks. Targeted attacks are more challenging than non-targeted attacks because they not only aim to cause incorrect predictions by the models but also to mislead the models into predicting a specific target class. We can categorize targeted attacks into two types based on whether they require a training dataset. Previous works have utilized training datasets to generate auxiliary modules [55, 56] for transfer-based attacks or directly train a model to generate adversarial examples [91, 148, 135]. This paper primarily focuses on transfer-based attacks that do not require a training dataset.

Targeted transfer-based attacks are widely studied in adversarial machine learning due to their practicality, as they do not require access to the target model’s architecture and parameters, nor do they need training datasets. The Activation Attack (AA) [57] proposes using the distance between adversarial examples and targeted images in feature space as a loss function, but its attack effectiveness may depend on the selection of targeted images. Po+Trip [69] employs Poincare distance to adapt the gradients’ magnitude and uses triplet loss to address insufficient inter-class distances, but it only works well on target models with similar architectures. Logit Attack (Logit) [156] discovers that a simple logit loss can generate better transferable adversarial examples than CE loss, significantly outperforming previous methods. Self-Universality (SU) method [137] finds that highly universal adversarial perturbations tend to be more transferable for targeted attacks and proposes optimizing the perturbations in different local regions of the image. The Object-based Diverse Input (ODI) method [11] is an efficient data augmentation in image space for targeted attacks using random 3D object and rendering.

Recently, the Clean Feature Mixup (CFM) method [12] proposes a feature-level aug-

mentation which achieves state-of-the-art performance in targeted transfer-based attacks. It stores clean features as random feature perturbations, which mitigates the overfitting to surrogate models. However, the proposed clean features in CFM lack specific optimization for targeted attacks, leaving significant room for improvement. We develop the Feature Tuning Mixup (FTM) method, which effectively and efficiently optimizes attack-specific feature perturbations and significantly improves transferable targeted attacks.

Differences from intermediate-level attacks. This paper investigates feature-level augmentation for targeted attacks, which differs from existing intermediate-level attacks [51, 71, 134, 151]. Our FTM is based on the transfer-based attack framework [67] with logit loss [156], which is currently the most effective framework for targeted transfer-based attacks. In contrast, intermediate-level attacks use loss functions based on a specific intermediate layer, which is hard to apply to targeted attacks. For instance, FIA [134] and NAA [151] require identifying important features based on the true label, while targeted attacks focus on the target label instead of the true label. ILA [51] requires an existing adversarial example as guidance and cannot function as a standalone targeted attack. Therefore, we do not compare FTM with intermediate-level attacks in experiments.

Comparison with adversarial learning methods. Our method learns feature perturbations in an adversarial manner without requiring additional training datasets. While some existing attacks [140, 97] also employ adversarial strategies, they operate in the image space and typically require access to training datasets. For instance, ATTA [140] improves transferability by training an adversarial transformation network to destroy adversarial examples. However, this approach demands substantial computational resources and training data. Like previous work [137, 156, 11, 12], we study targeted transfer attacks without extra data and do not compare with methods that need training data. RAP [97] utilizes adversarial learning to identify worst-case perturbations in the image space, aiming to locate regions with flat loss landscapes.

However, this process requires numerous additional backpropagation steps, leading to prohibitively high computational costs. In contrast, we propose a novel approach that optimizes feature perturbations directly in the multi-layer feature space, an unexplored direction in previous research. Through this innovation, our method achieves superior practicality, effectiveness, and efficiency compared to existing methods.

3.3 Methodology

3.3.1 Preliminaries

Problem formulation. In the setting of targeted transfer-based attacks, the adversary is assumed to have access to a white-box surrogate model F , which, in this work, is a pre-trained image classifier. We further assume that the attacker does not possess any additional training data. Given a benign input image x , a surrogate model F , and a target label y_t that differs from the ground-truth label y of x , the objective is to generate an adversarial example x^{adv} that is misclassified as y_t by F . The perturbation applied to x should be imperceptible, and following prior work, we enforce an ℓ_∞ -norm constraint on the perturbation magnitude. The optimization objective can thus be defined as follows:

$$\arg \min_{x^{\text{adv}}} \mathcal{L}(F(x^{\text{adv}}), y_t), \quad s.t. \quad \|x - x^{\text{adv}}\|_\infty \leq \epsilon, \quad (3.1)$$

where $\mathcal{L}$ is the adversarial loss, e.g., the cross-entropy loss or logit-based loss, and ϵ is the perturbation budget.

MI-FGSM. To solve the problem above, existing methods usually adopt the iterative

gradient-based technique, e.g., the MI-FGSM [22], formulated as follows:

$$g_{i+1} = \nabla_x \mathcal{L}(F(x_i^{\text{adv}}), y_t), \quad (3.2)$$

$$\tilde{g}_{i+1} = \mu \cdot g_i + g_{i+1} / \|g_{i+1}\|_1, \quad (3.3)$$

$$x_{i+1}^{\text{adv}} = x_i^{\text{adv}} - \eta \cdot \text{sign}(\tilde{g}_{i+1}), \quad (3.4)$$

$$x_{i+1}^{\text{adv}} = \text{Clip}_{x,\epsilon}(x_{i+1}^{\text{adv}}), \quad (3.5)$$

where g_i is the gradient of the loss function with respect to the input image x in the i -th iteration. μ is the momentum decay factor, and η is the step size. x_{i+1}^{adv} is clipped within ϵ bound. Initially, we set $x_0^{\text{adv}} = x$ and $g_0 = 0$.

CFM. To improve the transferability of targeted attacks, the Clean Feature Mixup (CFM) method [12] leverages stored clean features to augment adversarial examples, achieving state-of-the-art performance among existing targeted transfer-based attacks. Specifically, it selects outputs from certain network layers as perturbation targets and uses output features from different benign images as feature perturbations. To integrate with MI-FGSM, the feature augmentation in CFM can be formulated as follows:

$$z'_{k,i} = (1 - \alpha_{k,i}) \odot z_{k,i} + \alpha_{k,i} \odot z_{k,i}^c, \quad (3.6)$$

where $\odot$ denotes element-wise multiplication, $z_{k,i}$ is the input feature to the k -th layer in the i -th attack iteration of MI-FGSM, $z_{k,i}^c$ is a random stored clean feature, and $\alpha_{k,i}$ is a vector representing channel-wise mixing ratios. The network layer k is randomly selected from pre-defined candidate layers with probability p .

3.3.2 Analysis of CFM

CFM is a powerful *feature-level augmentation* method for targeted attacks. To understand the effectiveness of CFM, we formulate a generalized optimization step for

MI-FGSM based on feature-level augmentation as follows:

$$g'_{i+1} = \nabla_x \mathcal{L} \left(F(x_i^{\text{adv}}; \Delta z_i), y_t \right), \quad (3.7)$$

$$\Delta z_i = \{\Delta z_{k,i}, k = 1, \dots, n\}, \quad (3.8)$$

where $F(x_i^{\text{adv}}; \Delta z_i)$ denotes the surrogate model F with the perturbation Δz_i added to the network outputs in the i -th attack iteration. Equation (3.7) improves upon Equation (3.2) by incorporating feature-level perturbations, which is the key idea of CFM. Notably, the perturbation Δz_i represents the collection of perturbations for *all* network layers. We use $\Delta z_{k,i}$ to denote the perturbation for the input of the k -th layer, where n is the maximum layer index of F . Initially, we set $\Delta z_i = 0$. For any layer d that does not require perturbation, we simply set $\Delta z_{d,i} \equiv 0$.

CFM uses clean features to construct Δz_i , and it claims that the rationale is that clean features can suppress the adversarial effect of x_i^{adv} [12]. We formulate this idea in the following equation:

$$\mathcal{L} \left(F(x_i^{\text{adv}}; \Delta z_i), y_t \right) > \mathcal{L} \left(F(x_i^{\text{adv}}), y_t \right), \quad (3.9)$$

where Δz_i is supposed to appropriately disrupt adversarial loss in Equation (3.1) for obtaining robust x^{adv} .

However, clean features are attack-agnostic in nature, as they are not explicitly optimized for the objective in Equation (3.9). Although CFM demonstrates promising improvements, we argue that there remains unexplored potential in feature-level augmentation for targeted attacks. We believe that integrating attack-specific optimization with attack-agnostic clean features can create a more powerful feature-level augmentation strategy.

3.3.3 Feature Tuning Mixup (FTM)

We propose Feature Tuning Mixup (FTM), a novel method that uses attack-specific feature perturbations to enhance feature-level augmentation methods. We formu-

late the optimization problem for learning our feature perturbations and adversarial examples in each attack iteration as follows:

$$\min_{x_i^{\text{adv}}} \max_{\Delta z_i} \mathcal{L} \left(F(x_i^{\text{adv}}; \Delta z_i), y_t \right). \quad (3.10)$$

However, the optimization problem above is challenging to solve from two perspectives: effectiveness and efficiency. Regarding efficiency, traditional adversarial training [86] often requires multiple steps to optimize Δz_i in each attack iteration, resulting in substantial additional computational cost. As for effectiveness, since Δz_i involves a joint problem across multiple iterations and network layers, inappropriate perturbations can damage the attack effect, which will be shown in our experiments.

Momentum-based stochastic update. We propose a novel technique, momentum-based stochastic update, to optimize Δz_i . We employ one-step optimization with using perturbations in previous iterations as momentum, i.e., Δz_i will be initialized as Δz_{i-1} in Equation (3.10). Momentum can effectively improve the optimization effectiveness without increasing additional optimization steps. In addition, as we only require one-step optimization, the gradient for Δz_i can be jointly computed with the gradient for x_i^{adv} , which can further speed up the optimization process. To further improve the optimization efficiency, we use a stochastic update strategy, where only the randomly selected layers are updated in each attack iteration. For each selected layer k , it has probability p to be updated in the i -th attack iteration, otherwise, we set $\Delta z_{k,i} = \Delta z_{k,i-1}$. We illustrate the detailed optimization process for FTM in the following.

The forward process for FTM in the i -th attack iteration is formulated as follows:

$$\bar{z}_{k,i} = z_{k,i} + \beta \|z_{k,i}\| \cdot \frac{\Delta z_{k,i}}{\|\Delta z_{k,i}\| + \bar{\epsilon}}, \quad (3.11)$$

$$z'_{k,i} = \begin{cases} (1 - \alpha_{k,i}) \odot \bar{z}_{k,i} + \alpha_{k,i} \odot z_{k,i}^c, & \tau_k < p, \\ \bar{z}_{k,i}, & \text{else,} \end{cases} \quad (3.12)$$

where $\beta\|z_{k,i}\|$ is the custom perturbation budget for our feature perturbation $\Delta z_{k,i}$. β is a scaling factor, and $\bar{\epsilon}$ is a small constant. We mix the output feature $z_{k,i}$ and our attack-specific perturbation $\Delta z_{k,i}$ to obtain $\bar{z}_{k,i}$. Then, we use clean features to augment $\bar{z}_{k,i}$ to get $z'_{k,i}$, which is the final perturbed feature that will be fed into the k -th network layer. These perturbed features are computed layer by layer during the forward process. $z_{k,i}^c$ is a random clean feature, and $\alpha_{k,i}$ is a vector representing channel-wise mixing ratio. We randomly sample $\alpha_{k,i}$ from $\mathcal{U}(0, \alpha_{max})$. In each iteration, we randomly sample τ_k from $\mathcal{U}(0, 1)$ to determine whether to update the k -th perturbation $\Delta z_{k,i}$ during the i -th attack iteration. p is the probability threshold.

The backward process for FTM in the i -th attack iteration is formulated as follows:

$$\Delta z'_i = \{(\Delta z_{k,i}, \tau_k) \mid k = 1, \dots, n\}, \quad (3.13)$$

$$g'_{i+1}, g^{\Delta z'} = \nabla_{x_i^{\text{adv}}, \Delta z'_i} \mathcal{L}(F(x_i^{\text{adv}}; \Delta z'_i), y_t), \quad (3.14)$$

$$\Delta z_{k,i+1} = \begin{cases} \Delta z_{k,i} + g_k^{\Delta z'}, & \tau_k < p, \\ \Delta z_{k,i}, & \text{otherwise,} \end{cases} \quad (3.15)$$

where $\Delta z'_i$ represents a set containing both perturbations $\Delta z_{k,i}$ and their corresponding random variables τ_k from the forward process, which determines whether each perturbation needs to be updated. $g_k^{\Delta z'}$ denotes the gradient with respect to the k -th perturbation $\Delta z_{k,i}$. The threshold p remains consistent with that used in the forward process. We summarize our FTM attack in Algorithm 2.

Efficiency analysis. FTM does not require complex operations or additional forward and backward passes. Therefore, FTM is theoretically efficient on both CNNs and ViTs. In contrast, SI, VT, Admix, and SU require additional forward and backward passes, while ODI requires a computationally expensive input transformation.

Ensemble of surrogate variants (FTM-E). Observing that FTM can efficiently perturb the surrogate model, we propose to further enhance the attack effectiveness through an ensemble strategy. We create multiple copies of the surrogate model and independently apply FTM to each copy, then ensemble them as a new surrogate

Algorithm 2 Feature Tuning Mixup (FTM) Algorithm

Input: Surrogate model F , clean image x , target label y_t , iteration number T , perturbation budget ϵ , input transformation $\phi(\cdot)$ (RDI, etc.), step size η , decay factor μ

Input: Mixing range α_{max} , probability p , mixing ratio β

Output: An adversarial example x^{adv}

- 1: $x_0^{\text{adv}} = x, g_0 = 0$.
 - 2: Initialize perturbation $\Delta z_0 = 0$
 - 3: **for** $i = 0 \rightarrow T - 1$ **do**:
 - 4: Extend $F(x_i^{\text{adv}})$ to $F(\phi(x_i^{\text{adv}}))$
 - 5: Select layers to update and obtain $\Delta z'_i$ from Δz_i
 - 6: Forward and backward pass to get g'_{i+1} and $g^{\Delta z'}$ using Equations (3.12) and (3.14)
 - 7: Obtain Δz_{i+1} using Equation (3.15)
 - 8: Calculate $\tilde{g}_{i+1} = \mu \cdot g_i + g'_{i+1} / \|g'_{i+1}\|_1$
 - 9: Update example $x_{i+1}^{\text{adv}} = x_i^{\text{adv}} - \eta \cdot \text{sign}(\tilde{g}_{i+1})$
 - 10: Clip x_{i+1}^{adv} to be ℓ_∞ -bounded by ϵ
 - 11: **end for**
 - 12: $x^{\text{adv}} = x_T^{\text{adv}}$
 - 13: **return** x^{adv}
-

model. We empirically show that utilizing an ensemble of two perturbed surrogate models can substantially enhance the attack performance while maintaining reasonable computational costs.

3.4 Experiments

3.4.1 Experimental Setup

Datasets. In line with prior research [11, 12, 137, 156], we conduct our experiments using the ImageNet-compatible dataset released as part of the NIPS 2017 Adversarial Challenge. This dataset consists of 1,000 images, each annotated with both a ground-truth label and a predefined target label, making it well-suited for evaluating targeted attack performance.

General settings. We adopt experimental configurations consistent with recent literature [156, 11, 12]. Specifically, we use an ℓ_∞ -norm setting with $\epsilon = 16/255$. We set decay $\mu = 1.0$ and step size $\eta = 2/255$. We perform 300 iterations ($T = 300$) for iterative targeted attacks as per [12]. For optimization of adversarial examples, all methods employ the simple logit loss function [156].

Models. We evaluate our method using 15 pre-trained DNNs, consisting of 10 CNN-based models and 5 transformer-based models. For CNN-based architectures, we employ VGG-16 [112], RN-18 [44], RN-50 [44], DN-121 [49], Xcep [17], MB-v2 [103], EF-B0 [122], IR-v2 [117], Inc-v3 [118], and Inc-v4 [117]. For Transformer-based architectures, we utilize ViT [24], LeViT [41], ConViT [21], Twins [18], and PiT [45].

Baseline attacks. We mainly compare our FTM against current competitive methods, including input-level methods DI [143], RDI [160], TI [23], SI [80], VT [131], Admix [132], and ODI [11]. ODI is especially designed for targeted attacks. In addition, we compare with SU [137] and CFM [12]. We primarily reference CFM, the current state-of-the-art method for targeted transfer-based attacks that requires no additional training dataset. To avoid redundancy, the ‘MI-TI’ prefix is omitted in our references, as these techniques are applied to all attack methods by default. RDI is the baseline method in most cases. For DI, RDI, TI, SI, Admix, and ODI, we follow the configurations specified in CFM. For Admix, we also follow CFM to configure the weight to $w = 0.2$ and use $N = 3$ mixed images (i.e., $m_2 = 3$ in [132]). Given Admix’s computational intensity, we set the number of scale images in Admix to 1 (i.e.,

$m_1 = 1$ in [132]) unless otherwise specified. We report the results for ODI instead of RDI-ODI, as ODI performs similarly to RDI-ODI [11, 12]. For SU and CFM, we use the original settings from their papers.

Settings for our FTM. We set the mixing ratio β to 0.01. For FTM-E, we set the number of perturbed models to 2 for all experiments due to the computational cost. Following CFM, we avoid perturbing shallow layers because their low-level features vary significantly with input transforms and may cause excessive disturbance during attacks, as explained in CFM’s paper. Specifically, we perturb layers where the feature map size is no greater than $\frac{1}{16}$ of the image size. For a pure ViT, we select the outputs of all fully-connected layers. We sample $\alpha_{k,i}$ from $\mathcal{U}(0, 0.75)$, i.e., $\alpha_{max} = 0.75$. The probability p is set to 0.1.

3.4.2 Success Rates of Targeted Attacks

Table 3.1 shows the transfer success rates of various attack methods on ten black-box models. When using Inc-v3 as the source model, RDI-FTM-E achieves an average success rate of 51.2%, surpassing the previous best method CFM by 12.0%. The improvement is particularly notable for challenging target models like IR-v2 and Inc-v4, where RDI-FTM-E increases the success rates by 16.2% and 14.4%, respectively. When using RN-50 as the source model, both RDI-FTM and RDI-FTM-E achieve higher success rates than CFM. The results show that the effective feature-level augmentation employed by FTM helps generate transferable adversarial examples.

Table 3.2 shows the transfer success rates of various attack methods on five transformer-based models. Our methods also achieve significant improvements over existing methods. For example, when RN-50 is the surrogate, RDI-FTM and RDI-FTM-E improve the targeted attack success rates of CFM by 4.9% and 8.0%, respectively. The results suggest that our method can effectively improve the transferability of adversarial examples across different architectures.

Source	Attack	Time (s)	VGG-16	RN-18	RN-50	DN-121	Xcep	MB-v2	EF-B0	IR-v2	Inc-v3	Inc-v4	Avg.
RN-50	DI	1.37	63.2	56.4	98.9	74.7	5.5	27.8	29.0	5.6	9.0	9.4	37.9
	RDI	1.23	66.3	70.7	98.2	81.9	13.2	46.4	47.2	17.1	29.9	23.3	49.4
	RDI-SI	5.95	70.5	79.8	<u>98.8</u>	88.9	29.5	56.2	66.2	37.9	56.4	43.6	62.8
	RDI-VT	7.16	68.8	78.7	98.2	82.5	27.9	54.5	56.1	32.8	45.8	37.9	58.3
	RDI-Admix	3.55	74.2	80.7	98.7	86.8	20.9	59.4	56.1	26.7	42.7	34.1	58.0
	RDI-SU	2.21	67.9	74.7	98.3	83.3	24.9	56.3	50.2	20.9	32.1	26.2	53.5
	ODI	4.38	78.3	77.1	97.6	87.0	43.8	67.3	70.0	49.5	65.9	55.4	69.2
	RDI-CFM	1.39	84.7	<u>88.4</u>	98.4	<u>90.3</u>	51.1	81.5	78.8	48.0	65.5	59.3	74.6
	RDI-FTM	1.54	<u>86.3</u>	87.5	97.9	89.7	<u>56.1</u>	<u>83.3</u>	<u>81.2</u>	<u>54.7</u>	<u>70.8</u>	<u>66.6</u>	<u>77.4</u>
	RDI-FTM-E	2.92	88.1	88.6	98.3	91.4	59.4	85.4	84.3	56.9	73.4	69.3	79.5
Inc-v3	DI	2.01	2.9	2.4	3.4	5.0	1.9	1.8	3.7	3.0	99.2	4.2	12.8
	RDI	1.76	3.5	3.8	4.0	7.0	3.1	3.0	5.9	6.3	98.7	7.1	14.2
	RDI-SI	8.11	4.0	5.2	5.7	11.0	6.3	4.6	8.2	11.6	<u>98.8</u>	12.1	16.8
	RDI-VT	10.50	5.9	8.9	9.4	13.2	7.4	5.9	9.8	12.3	98.7	14.7	18.6
	RDI-Admix	4.92	6.3	6.5	8.8	12.8	6.0	6.1	10.9	12.2	98.7	13.6	18.2
	RDI-SU	2.36	7.8	11.0	11.4	12.7	9.4	8.2	6.3	8.3	98.7	10.9	18.5
	ODI	6.42	14.3	14.9	16.7	32.3	20.3	13.7	25.3	26.4	95.9	31.6	29.1
	RDI-CFM	2.13	22.9	26.8	26.2	39.1	34.1	27.1	38.6	36.2	95.9	44.8	39.2
	RDI-FTM	2.35	<u>25.8</u>	<u>30.9</u>	<u>32.1</u>	<u>41.7</u>	<u>36.3</u>	<u>30.7</u>	<u>41.4</u>	<u>42.9</u>	94.1	<u>48.1</u>	<u>42.4</u>
	RDI-FTM-E	4.37	34.2	39.3	40.6	54.4	47.0	37.1	51.6	52.4	96.6	59.2	51.2

Table 3.1: Success rates (%) of targeted attacks evaluated across ten target models. The best-performing results are highlighted in bold, while the second-best are indicated with underlining. Time (s) represents the average computational time required to generate an adversarial example for a single image.

We report the targeted attack success rates when using DN-121 or LeViT as the surrogate model in Table 3.3. The results demonstrate that, across all black-box attack scenarios, our methods consistently outperform existing approaches, regardless of whether DN-121 or LeViT is used as the surrogate model.

We show that FTM remains effective and efficient when using ViT as the surrogate model. Unlike other models in Section 3.4.1, ViT’s feature map size does not decrease with depth. As analyzed in Section 3.3.3, FTM is theoretically efficient and adaptable to various architectures, including ViT. Tables 3.4 and 3.5 confirm that FTM maintains low computational complexity when using ViT as the surrogate model. Our

Source	Attack	Time (s)	ViT	LeViT	ConViT	Twins	PiT	Avg.
RN-50	DI	1.37	0.2	3.5	0.4	1.5	1.7	1.5
	RDI	1.23	0.7	13.2	1.7	6.1	7.0	5.7
	RDI-SI	5.95	2.9	29.4	6.3	15.5	17.9	14.4
	RDI-VT	7.16	2.9	28.1	5.2	15.0	14.0	13.0
	RDI-Admix	3.55	1.3	22.5	2.5	8.5	8.4	8.6
	RDI-SU	2.21	0.8	16.9	2.3	6.9	7.8	6.9
	ODI	4.38	5.1	37.0	10.7	20.1	29.1	20.4
	RDI-CFM	1.39	4.3	46.1	8.9	25.2	24.7	21.8
	RDI-FTM	1.54	<u>5.9</u>	<u>52.9</u>	<u>10.8</u>	<u>32.4</u>	<u>31.5</u>	<u>26.7</u>
	RDI-FTM-E	2.92	6.8	58.6	13.6	35.2	34.9	29.8
Inc-v3	DI	2.01	0.1	0.3	0.0	0.0	0.1	0.1
	RDI	1.76	0.2	1.8	0.2	0.4	0.7	0.7
	RDI-SI	8.11	0.3	4.1	0.9	0.7	3.2	1.8
	RDI-VT	10.5	0.4	5.2	0.8	1.6	1.8	2.0
	RDI-Admix	4.92	0.1	4.1	0.6	1.4	1.4	1.5
	RDI-SU	2.36	0.2	2.0	0.3	1.3	1.0	1.0
	ODI	6.42	0.8	12.4	1.7	3.5	6.7	5.0
	RDI-CFM	2.13	2.1	21.9	3.2	6.1	11.6	8.9
	RDI-FTM	2.35	<u>2.4</u>	<u>25.0</u>	<u>4.5</u>	<u>10.0</u>	<u>15.3</u>	<u>11.5</u>
	RDI-FTM-E	4.37	3.8	32.7	6.8	12.7	20.4	15.3

Table 3.2: Success rates (%) of targeted attacks evaluated across five target models. The best-performing results are highlighted in bold, while the second-best are indicated with underlining. Time (s) represents the average computational time required to generate an adversarial example for a single image.

RDI-FTM-E-SI $_{m_1=2}$ generates adversarial examples in just 3.82 seconds on average while significantly outperforming existing attacks. In RDI-FTM-E-SI $_{m_1=2}$, the two copies of the surrogate model in FTM-E use scaled inputs (1 and 0.5, respectively), preserving efficiency. The parameter m_1 in RDI-FTM-E-SI $_{m_1=2}$ is set to 2 to correspond with the two copies of the surrogate model used in FTM-E. Overall, our results validate FTM’s effectiveness and efficiency across different architectures.

Attack	DN-121 $\Rightarrow$											
	VGG-16	RN-50	Inc-v3	DN-121*	IR-v2	Inc-v4	Xcep	ViT	LeViT	ConViT	Twins	PiT
DI	37.1	44.4	7.1	98.7	4.3	8.3	5.2	0.2	3.0	0.4	1.0	1.1
RDI	42.1	55.7	20.8	98.5	12.8	18.8	10.1	0.8	8.5	1.3	3.7	4.5
RDI-Admix	53.2	67.6	31.1	98.3	20.1	26.5	17.8	1.0	14.7	1.7	6.8	7.4
RDI-Admix ₅	49.6	65.3	41.0	<u>98.6</u>	28.9	34.3	21.6	2.4	21.8	3.4	10.5	14.2
RDI-SI	45.4	60.1	34.3	<u>98.6</u>	22.0	25.8	16.1	2.0	16.1	2.4	8.2	11.7
RDI-VT	47.7	62.1	31.5	<u>98.6</u>	25.4	27.2	20.3	2.2	19.2	3.5	8.3	11.7
RDI-ODI	64.2	71.7	52.8	98.0	39.8	45.9	31.4	3.3	26.9	7.4	14.7	21.9
RDI-CFM	76.2	83.9	56.1	97.8	43.6	53.8	41.1	3.6	32.8	6.4	17.3	21.1
RDI-FTM	<u>77.3</u>	<u>85.6</u>	<u>62.8</u>	98.2	<u>46.9</u>	<u>58.4</u>	<u>46.0</u>	<u>3.7</u>	<u>40.0</u>	<u>8.2</u>	<u>21.6</u>	<u>26.5</u>
RDI-FTM-E	79.5	87.6	64.6	98.0	48.7	60.1	47.6	4.2	43.4	8.9	24.4	26.7

Attack	LeViT $\Rightarrow$											
	VGG-16	RN-50	Inc-v3	DN-121	IR-v2	Inc-v4	Xcep	ViT	LeViT*	ConViT	Twins	PiT
DI	1.6	2.5	4.2	2.7	1.8	2.6	2.3	0.6	100	4.2	9.3	10.3
RDI	3.0	3.5	5.4	4.1	3.6	4.6	2.3	1.1	100	7.9	13.8	22.1
RDI-Admix	6.5	8.3	9.9	8.8	5.8	7.2	5.5	3.3	100	10.7	23.3	30.9
RDI-Admix ₅	5.3	8.1	13.9	12.4	9.9	8.4	7.2	8.0	99.9	20.6	30.0	47.2
RDI-SI	3.4	6.3	10.6	10.0	6.4	5.4	5.1	4.3	100	18.1	24.1	38.4
RDI-VT	5.3	7.2	12.0	10.1	8.3	8.8	8.9	6.3	99.9	18.7	27.2	40.5
RDI-ODI	21.0	25.0	40.9	38.3	25.7	31.2	26.3	15.3	98.7	34.1	43.8	66.1
RDI-CFM	27.3	30.3	39.8	39.0	23.6	30.1	27.2	18.4	100	45.5	63.8	75.7
RDI-FTM	<u>41.3</u>	<u>41.9</u>	<u>56.9</u>	<u>54.2</u>	<u>39.8</u>	<u>46.5</u>	<u>42.2</u>	<u>31.1</u>	99.9	<u>63.2</u>	<u>77.5</u>	<u>86.7</u>
RDI-FTM-E	50.1	52.4	62.8	63.1	48.4	53.2	49.2	40.3	99.9	72.2	86.2	93.0

Table 3.3: Targeted attack success rates (%) using DN-121 or LeViT as surrogate model. All methods are combined with MI-TI. The best results are shown in bold, and the second best results are underlined. The surrogate models are marked with * in the first column.

3.4.3 Combination with Existing Techniques

Table 3.6 presents transfer success rates of our method combined with ensemble-based method [81], SI, and Admix. Using the ensemble of RN-50 and Inc-v3 as the surrogate model, our RDI-SI-FTM-E achieves 74.5% average success rate, surpassing existing methods. FTM notably enhances performance on transformer-based models. For

3.4. Experiments

Source	Attack	Time (s)	VGG-16	RN-18	RN-50	DN-121	Xcep	MB-v2	EF-B0	IR-v2	Inc-v3	Inc-v4	Avg.
ViT	RDI	1.76	2.4	2.4	3.1	5.2	3.5	3.6	10.6	4.4	5.3	4.3	4.5
	RDI-VT	10.20	2.5	3.6	4.1	7.1	5.9	4.1	13.4	7.4	5.5	5.4	5.9
	RDI-Admix	5.11	5.4	5.4	6.9	11.5	6.7	6.9	18.7	8.1	10.1	8.0	8.8
	RDI-SI	8.38	4.2	8.2	9.5	17.6	10.2	9.1	24.7	13.3	18.0	12.1	12.7
	RDI-ODI	4.53	10.2	12.7	14.3	22.1	18.6	12.1	32.2	<u>22.5</u>	22.3	20.7	18.8
	RDI-CFM	1.80	10.8	14.2	14.9	21.0	14.9	15.0	31.9	15.0	18.9	17.1	17.4
	RDI-FTM	1.92	11.6	15.1	14.6	22.1	16.7	18.6	31.8	15.8	20.5	18.0	18.5
	RDI-FTM-E	3.81	<u>13.4</u>	<u>18.1</u>	<u>18.3</u>	<u>26.5</u>	<u>19.6</u>	<u>19.5</u>	<u>37.2</u>	19.8	<u>23.0</u>	<u>21.3</u>	<u>21.7</u>
	RDI-FTM-E-SI _{m1=2}	3.82	24.0	29.1	30.0	41.5	28.9	31.3	54.7	30.9	39.3	31.8	34.2

Table 3.4: Success rates (%) of targeted attacks evaluated across ten target models, with ViT as the surrogate model. All methods are combined with MI-TI. The best results are shown in bold, and the second best results are underlined. Time (s) denotes the average computation time required to attack a single image. RDI-FTM-E-SI_{m1=2} denotes using scaled input for RDI-FTM-E.

Source	Attack	Time (s)	ViT	LeViT	ConViT	Twins	PiT	Avg.
ViT	RDI	1.76	99.5	24.3	23.3	11.7	26.0	37.0
	RDI-VT	10.20	98.3	32.0	29.7	15.3	35.7	42.2
	RDI-Admix	5.11	<u>99.1</u>	39.0	34.6	18.5	40.5	46.3
	RDI-SI	8.38	98.4	<u>58.4</u>	<u>69.2</u>	<u>38.8</u>	<u>63.8</u>	<u>65.7</u>
	RDI-ODI	4.53	87.6	45.9	42.1	29.5	49.3	50.9
	RDI-CFM	1.80	96.5	47.2	50.1	27.4	48.8	54.0
	RDI-FTM	1.92	94.3	50.5	53.3	31.0	52.6	56.3
	RDI-FTM-E	3.81	96.0	54.9	55.6	36.3	55.8	59.7
	RDI-FTM-E-SI _{m1=2}	3.82	96.3	71.2	74.8	54.7	74.6	74.3

Table 3.5: Success rates (%) of targeted attacks evaluated across five transformer-based DNNs, with ViT as the surrogate model. All methods are combined with MI-TI. The best results are shown in bold, and the second best are underlined. Time (s) denotes the average time required to attack a single image. RDI-FTM-E-SI_{m1=2} denotes using scaled input for RDI-FTM-E.

instance, RDI-SI-FTM achieves 64.3% on PiT, which is 5.0% higher than RDI-SI-CFM (59.3%) with comparable computational cost. These results demonstrate that FTM is a generic technique that combines effectively with existing attack methods.

Attack	RN-50 + Inc-v3 $\Rightarrow$													
	Time (s)	VGG-16	RN-50*	Inc-v3*	DN-121	IR-v2	Inc-v4	Xcep	ViT	LeViT	ConViT	Twins	PiT	Avg.
RDI	3.26	66.9	98.4	98.7	83.1	35.6	42.5	25.4	0.8	22.4	3.2	9.8	10.8	41.5
ODI	7.17	70.9	94.0	96.1	83.4	65.4	69.9	57.6	9.1	52.9	18.6	28.6	42.8	57.4
RDI-CFM	4.39	86.1	<u>98.3</u>	97.0	92.1	71.7	79.9	72.0	10.3	66.5	20.3	41.9	45.8	65.2
RDI-ODI-CFM	7.73	63.9	88.6	81.3	74.1	54.3	56.0	54.3	12.6	49.9	20.9	29.6	41.0	52.2
RDI-Admix-CFM	13.38	86.2	98.1	<u>97.7</u>	92.7	75.1	81.6	76.1	14.0	73.1	24.8	50.7	53.9	68.7
RDI-SI-CFM	22.13	87.5	98.1	97.3	92.7	78.0	81.6	78.0	20.8	76.3	32.9	55.0	59.3	71.5
RDI-FTM	4.47	87.5	97.8	96.7	92.4	75.4	81.7	74.0	14.5	71.0	25.2	49.3	54.4	68.3
RDI-FTM-E	8.81	88.2	98.1	97.1	<u>93.4</u>	76.7	83.4	77.1	18.4	74.9	30.1	53.4	58.0	70.7
RDI-Admix-FTM	14.10	86.0	96.7	96.5	92.2	78.3	83.8	77.3	20.1	75.8	29.9	55.7	58.5	70.9
RDI-SI-FTM	22.66	87.5	96.8	96.2	93.3	<u>79.0</u>	<u>83.6</u>	79.1	<u>25.7</u>	<u>77.7</u>	<u>37.5</u>	<u>59.9</u>	<u>64.3</u>	<u>73.4</u>
RDI-SI-FTM-E	43.43	<u>87.8</u>	97.3	96.7	93.9	79.1	83.5	<u>78.2</u>	27.3	79.2	42.9	62.9	65.6	74.5

Table 3.6: Targeted attack success rates (%) of method combinations. The surrogate model used is an ensemble of RN-50 and Inc-v3. The best results are indicated in bold, while the second-best are underlined. Time (s) refers to the average time required to generate a targeted adversarial example for a single image.

We report the targeted attack success rates using ensemble-based surrogate models in Table 3.7. We use two settings for the ensemble-based surrogate models: RN-50 + Inc-v3 and RN-50 + LeViT. The results show that our methods can be combined with current ensemble-based methods to further improve the attack success rates.

3.4.4 Attacking Multimodal LLMs

FTM can be used to attack multimodal LLMs. We use four commercial LLMs for evaluation, including Qwen2-VL, Llama-3.2, Claude-3.5, and GPT-4o. We randomly select 100 images from the ImageNet-compatible dataset and generate targeted adversarial examples on ViT using RDI-FTM-E. For each generated targeted adversarial example, we use the prompt “Is this image a photo of {target label}? Yes or No?” to obtain the predictions of the LLMs. Table 3.8 shows that our method achieves an average attack success rate of 40.5%.

Attack	RN-50 + Inc-v3 $\Rightarrow$											
	VGG-16	RN-50*	Inc-v3*	DN-121	IR-v2	Inc-v4	Xcep	ViT	LeViT	ConViT	Twins	PiT
DI	63.5	99.0	99.2	77.0	16.6	24.0	12.5	0.2	7.3	0.7	3.1	2.6
RDI	66.9	<u>98.4</u>	98.7	83.1	35.6	42.5	25.4	0.8	22.4	3.2	9.8	10.8
RDI-SI	70.1	98.0	98.9	89.3	53.3	57.3	40.0	4.5	40.7	8.8	21.4	27.4
RDI-VT	66.4	<u>98.4</u>	98.8	83.1	51.6	56.6	40.7	5.0	39.6	8.4	21.6	22.7
RDI-Admix	72.9	<u>98.4</u>	<u>99.1</u>	88.0	46.3	58.3	36.5	1.8	33.4	4.5	15.4	17.1
RDI-ODI	70.9	94.0	96.1	83.4	65.4	69.9	57.6	9.1	52.9	18.6	28.6	42.8
RDI-CFM	86.1	98.3	97.0	92.1	71.7	79.9	72.0	10.3	66.5	20.3	41.9	45.8
RDI-FTM	<u>87.5</u>	97.8	96.7	<u>92.4</u>	<u>75.4</u>	<u>81.7</u>	<u>74.0</u>	<u>14.5</u>	<u>71.0</u>	<u>25.2</u>	<u>49.3</u>	<u>54.4</u>
RDI-FTM-E	88.2	98.1	97.1	93.4	76.7	83.4	77.1	18.4	74.9	30.1	53.4	58.0

Attack	RN-50 + LeViT $\Rightarrow$											
	VGG-16	RN-50*	Inc-v3	DN-121	IR-v2	Inc-v4	Xcep	ViT	LeViT*	ConViT	Twins	PiT
DI	71.3	99.2	29.5	80.3	16.2	23.7	15.4	1.2	100	6.7	18.2	20.8
RDI	75.3	98.7	55.7	87.8	35.1	41.9	29.6	4.2	99.4	14.6	34.3	41.1
RDI-SI	78.1	<u>98.9</u>	74.0	92.8	53.1	57.4	45.5	10.6	<u>99.9</u>	30.5	52.2	67.1
RDI-VT	77.4	<u>98.9</u>	66.4	88.1	50.7	56.0	48.4	14.6	99.1	30.3	53.9	60.7
RDI-Admix	82.3	98.8	64.8	91.2	43.6	53.9	41.0	7.6	<u>99.9</u>	17.4	43.9	48.8
RDI-ODI	81.9	96.6	81.6	89.6	69.9	73.4	67.9	27.8	97.4	52.7	66.5	81.0
RDI-CFM	88.5	98.8	84.1	92.1	69.5	78.7	72.1	26.4	99.8	51.6	77.5	81.3
RDI-FTM	<u>89.2</u>	98.4	<u>86.4</u>	<u>93.0</u>	<u>74.2</u>	<u>81.0</u>	<u>77.1</u>	<u>41.0</u>	99.5	<u>66.1</u>	<u>85.5</u>	<u>89.2</u>
RDI-FTM-E	90.0	98.1	86.7	93.9	76.7	82.4	80.1	51.5	99.7	72.6	87.9	90.8

Table 3.7: Targeted attack success rates (%) using ensemble-based surrogate models. All methods are combined with MI-TI. The best results are shown in bold, and the second best results are underlined. The surrogate models are marked with * in the first column.

3.4.5 Further Analysis

Figure 3.3 shows the success rates of targeted attacks using different numbers of iterations. We evaluate our methods under different numbers of iterations using two experimental settings: RN-50 $\rightarrow$ Inc-v4 and Inc-v3 $\rightarrow$ IR-v2. The results show that FTM and FTM-E achieve stable performance across different iteration counts. For the RN-50 $\rightarrow$ Inc-v4 setting, SI exhibits superior performance at the beginning of the

Response	Qwen2-VL	Llama-3.2	Claude-3.5	GPT-4o	Avg
Total	100	100	100	100	100 %
Refuse to Answer	0	10	0	0	2.50 %
Uncertain	1	5	1	0	1.75 %
Attack Failed	52	42	54	73	55.25 %
Attack Succeeded	47	43	45	27	40.50 %

Table 3.8: Evaluation on multimodal LLMs. The targeted adversarial examples are generated on ViT using RDI-FTM-E.

optimization process. However, as the number of iterations increases, FTM surpasses existing approaches. In the Inc-v3 $\rightarrow$ IR-v2 setting, FTM-E significantly outperforms other methods across the entire range of iterations.

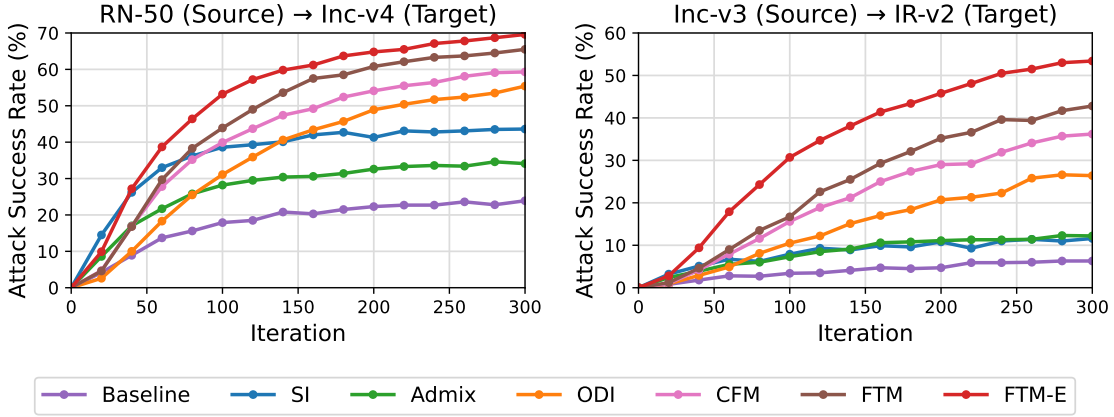


Figure 3.3: Success rates of targeted attacks under different iteration counts.

Computational overhead. We analyze the computational efficiency of different methods and report their costs in Table 3.2. We use an NVIDIA 3090 GPU to compute the time per image. The results demonstrate that our approach introduces reasonable computational overhead compared to CFM. Specifically, with RN-50 as the source model, RDI-FTM requires an average of 1.54 seconds per image, representing only a 10.8% increase over CFM’s 1.39 seconds. In contrast, alternative methods like SI, VT, and ODI impose substantially higher computational burdens. For instance,

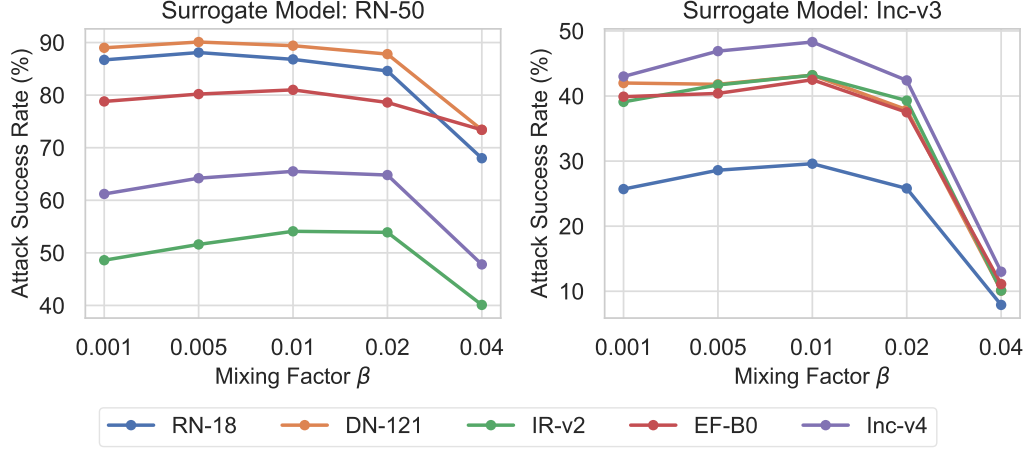


Figure 3.4: Attack success rates with different mixing ratio β . Each color represents the attack effectiveness on a specific target model.

VT requires 7.16 seconds per image, and SI requires 5.95 seconds per image. We also include relevant analysis in Figure 3.1. The result shows that FTM achieves strong attack performance while maintaining low computational cost.

3.4.6 Ablation Study

For all ablation experiments, we use RN-50 to generate adversarial examples and evaluate the average success rates on all models from Section 3.4.1, unless stated otherwise.

Stability with respect to β . Figure 3.4 shows the transfer success rates of FTM across different mixing ratios β . The success rates on five target models are plotted in distinct colors. When β exceeds 0.04, the success rates decrease substantially across all target models. However, for β values below 0.02, we observe stable attack performance across all models. The results suggest that FTM maintains robust performance within a reasonable range of β values, showing its stability with respect to β .

Compare FTM and CFM using ensemble strategy. Figure 3.5 shows the attack

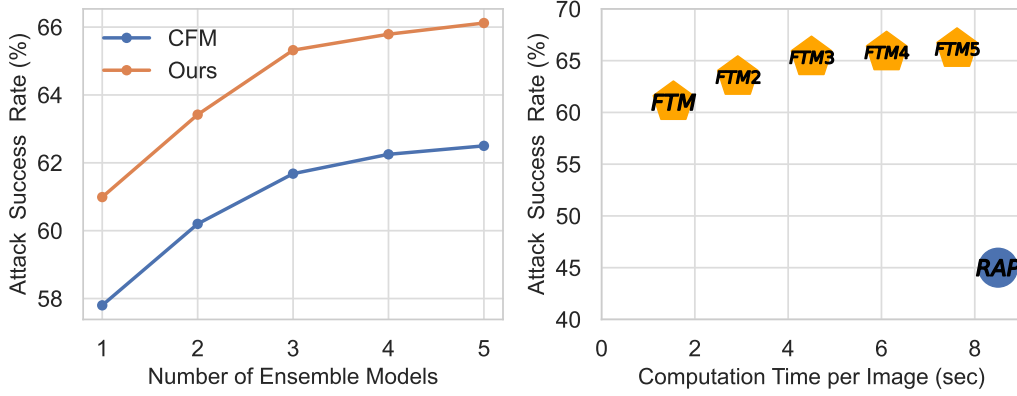


Figure 3.5: Attack performance and computational costs with different numbers of ensemble models. Left: Our FTM vs. CFM using ensemble strategy. Right: Computational costs of our FTM and RAP. -ens2 means using 2 perturbed models, and so on.

performance of FTM and CFM using different numbers of perturbed models. On the left, our FTM consistently outperforms CFM across all ensemble sizes. Notably, FTM with only 2 perturbed models achieves better attack performance than an ensemble of 5 CFM-perturbed models, underscoring FTM’s superior effectiveness. It is also worth mentioning that single FTM and CFM have comparable time complexity (Table 3.2). On the right, we compare our FTM with RAP [97], and the results show that FTM significantly outperforms RAP in both effectiveness and computational efficiency.

Analysis of probability p and ratio α_{max} . Figure 3.6 presents an analysis of FTM’s attack success rates under varying values of p and α_{max} . The left plot shows that FTM achieves optimal performance with $p = 0.1$ and $\alpha_{max} = 0.75$. FTM maintains competitive performance under various parameter configurations, such as $p, \alpha_{max} = 0.15, 0.5$ or $0.05, 1.0$, indicating its parameter stability. The right plot illustrates the impact of clean features on FTM. With clean features enabled, FTM’s performance declines significantly as p increases from 0.1 to 1.0, while FTM without clean features shows stable performance across varying p values. This suggests that although clean features are crucial to FTM’s performance, they also add instability

to the attack process. Additionally, setting $p = 1.0$ (where all layers are updated) causes excessive perturbations, making the attack overly aggressive and reducing its effectiveness.

The drop of the blue line in Figure 3.6 occurs because the clean accuracy of the perturbed surrogate model becomes extremely low (3.8%) at $p = 1.0$ when using clean feature mixup. Without clean feature mixup (orange line), the accuracy remains high at 65.3%. The low accuracy makes it impossible to generate effective adversarial examples, causing the attack success rate of the blue line to drop to near 0.

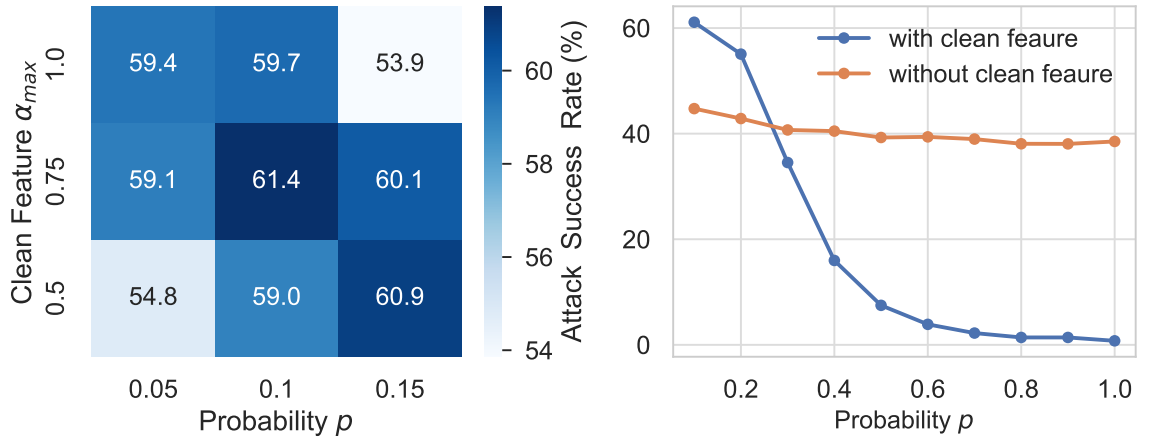


Figure 3.6: Left: Attack success rates vs. p and α_{max} . Right: Attack success rates vs. p with/without clean features; The blue line is the success rates when the clean features from CFM are integrated with our FTM, while the orange line represents only using our learnable perturbations.

Table 3.9 shows the clean accuracy of the perturbed surrogate model with varying values of p . The results demonstrate that increasing p leads to extremely low clean accuracy when using clean feature mixup in FTM, which explains the rapid decline of the blue line in Figure 3.6. In contrast, without clean feature mixup, the clean accuracy remains stable, maintaining 65.3% even at $p = 1.0$. Consequently, the corresponding orange line in Figure 3.6 exhibits relatively stable behavior. Notably, as shown in the left plot of Figure 3.6 our FTM performs stably around the optimal

hyperparameter p and α_{max} .

Ablation	Parameter p									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
w/ clean feature mixup	51.9	28.3	14.0	10.8	8.2	6.0	5.4	4.9	4.3	3.8
w/o clean feature mixup	79.8	73.3	69.7	68.6	67.0	66.4	66.0	65.8	65.4	65.3

Table 3.9: Accuracy (%) of clean images on the perturbed surrogate model with varying parameter p .

3.4.7 Visualization

Figure 3.7 and Figure 3.8 present visualizations of adversarial examples crafted by different targeted transfer-based attack methods.

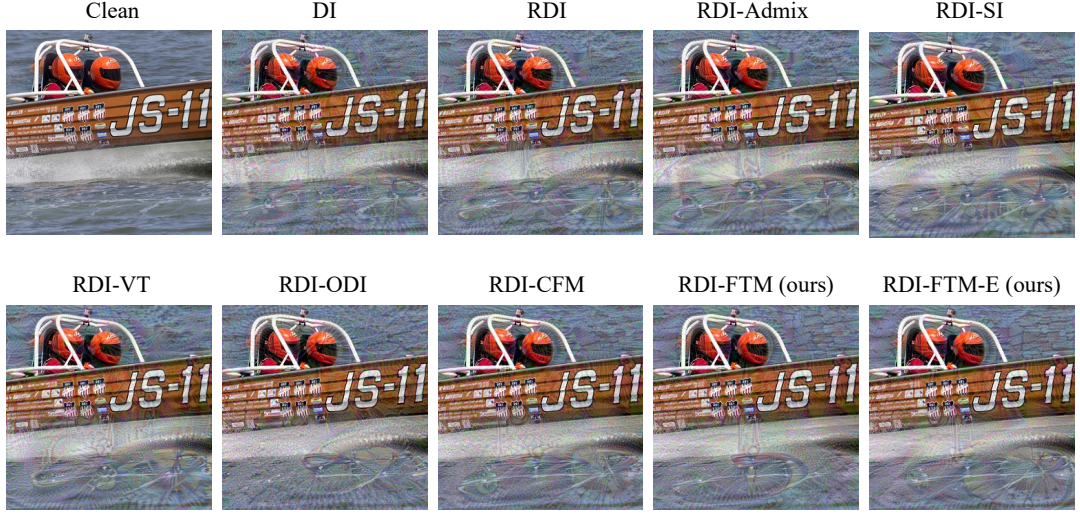
3.4.8 Discussion of Targeted Attacks under Defenses

We evaluate our methods, FTM and FTM-E, against strong defenses using adversarially trained networks. This is a realistic and challenging scenario, as adversarial training is a standard defense strategy.

We adopt three adversarially trained models that represent state-of-the-art robustness. Targeted transfer attacks are tested with two surrogate ensembles: RN-50 + LeViT and RN-50 + Inc-v3, as shown in Table 3.10 and Table 3.11. This setup evaluates whether our methods remain effective under both architectural variety and adversarial training.

Across all defenses, FTM and FTM-E consistently outperform existing targeted attack methods. Specifically, with the RN-50 + LeViT surrogate, FTM-E achieves the highest average success rate of 13.73%, surpassing strong baselines such as ODI and CFM. Similar trends are observed with the RN-50 + Inc-v3 surrogate, where

True Label: speedboat
Target Label: unicycle, monocycle



True Label: soccer ball
Target Label: potter's wheel

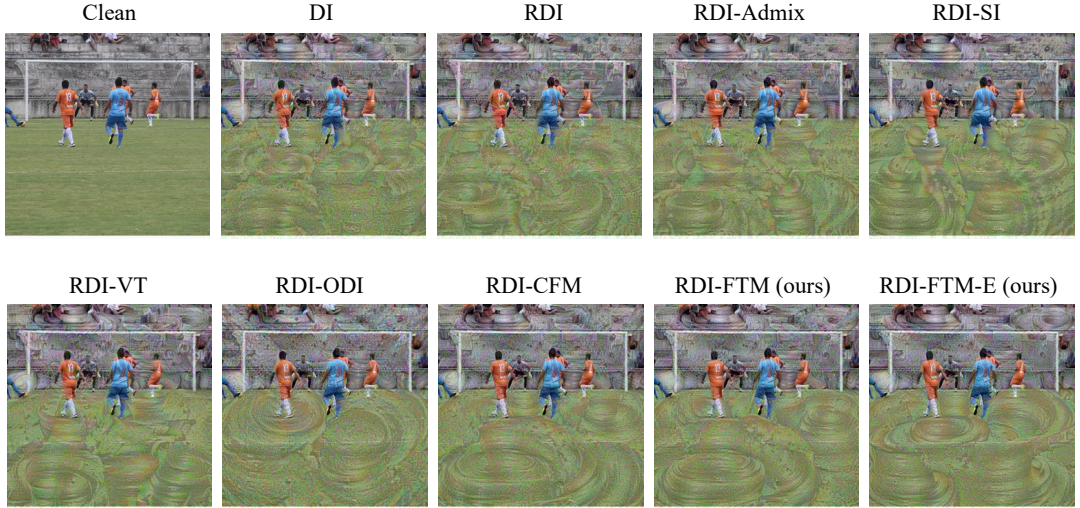
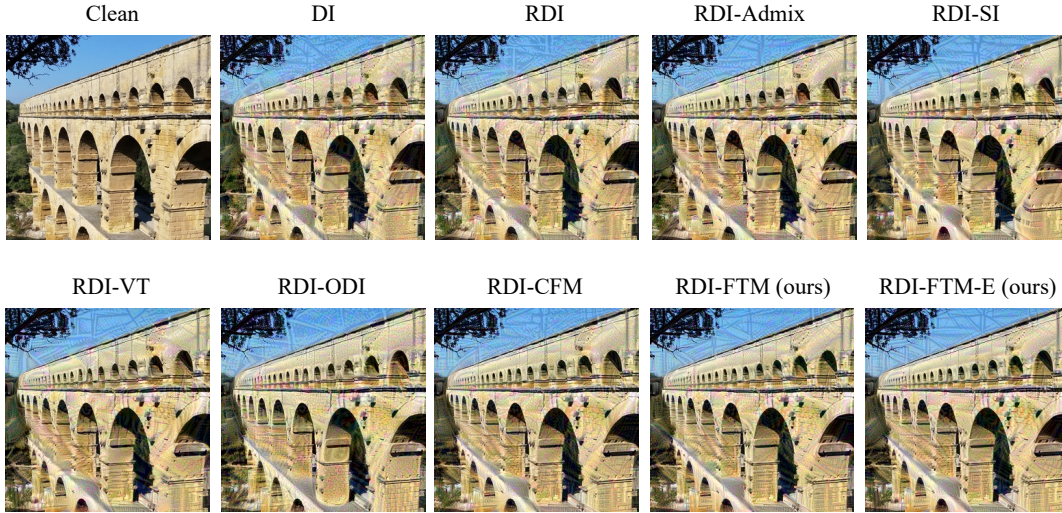


Figure 3.7: Visualization of targeted adversarial examples generated by different attack methods. The surrogate model used for attack generation is an ensemble of RN-50 and Inc-v3.

True Label: viaduct
Target Label: bullet train



True Label: lakeside, lakeshore
Target Label: dhole, Cuon alpinus

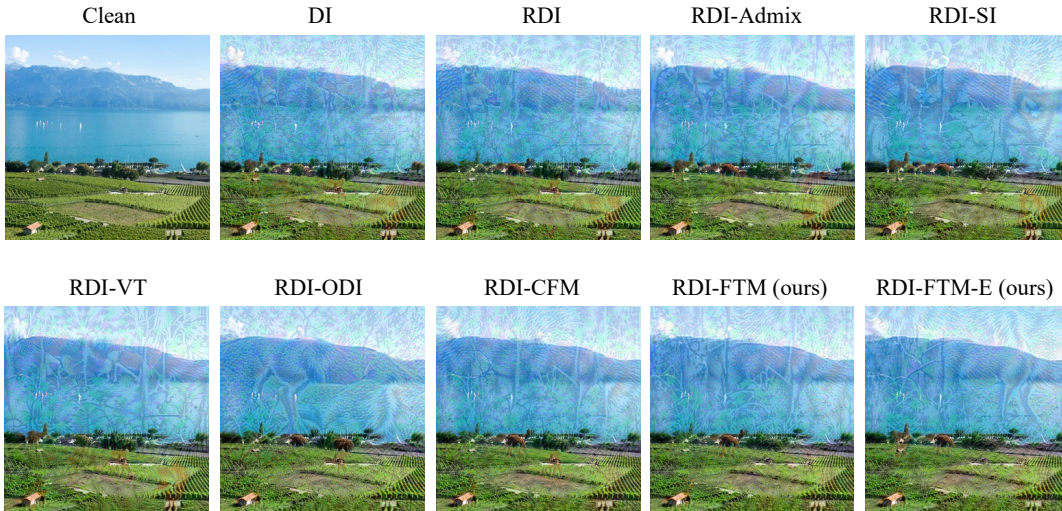


Figure 3.8: Visualization of targeted adversarial examples generated by different attack methods. The surrogate model used for attack generation is an ensemble of RN-50 and LeViT.

FTM-E again leads. These results demonstrate that our FTM method enhances the transferability of targeted attacks even against adversarially trained models.

However, the absolute attack success rates remain relatively low. This highlights a key limitation: targeted attacks struggle against effective defenses. Even the best methods achieve only moderate success, indicating that adversarial training substantially reduces the transferability of targeted attacks. Enhancing the transferability of targeted attacks remains an open problem for future research.

Attack	Inc-v3 _{ens3}	Inc-v3 _{ens4}	IR-v2 _{ens}	Average
RDI	0.2	0.2	0.1	0.17
RDI-SI	5.9	2.8	1.8	3.50
RDI-VT	8.7	4.4	2.7	5.27
RDI-Admix	0.5	0.2	0.2	0.30
ODI	14.8	6.3	<u>4.2</u>	8.43
RDI-CFM	7.5	3.4	2.4	4.43
RDI-FTM	<u>16.0</u>	<u>7.1</u>	3.8	<u>8.97</u>
RDI-FTM-E	22.0	12.3	6.9	13.73

Table 3.10: Success rates (%) of targeted attacks evaluated across three adversarially trained defenses. The surrogate model is the ensemble of RN-50 and LeViT. The best-performing results are highlighted in bold, while the second-best are indicated with underlining.

3.5 Summary

In this work, we address the challenging task of improving adversarial transferability in targeted attacks. Our analysis shows that feature-level augmentation presents a promising direction for enhancing transferability in both an effective and computationally efficient manner. To this end, we propose Feature Tuning Mixup (FTM), a novel attack method that blends attack-specific feature perturbations with clean

Attack	Inc-v3 _{ens3}	Inc-v3 _{ens4}	IR-v2 _{ens}	Average
RDI	0.2	0.0	0.0	0.07
RDI-SI	7.8	4.4	2.7	4.97
RDI-VT	7.2	2.3	1.1	3.53
RDI-Admix	0.8	0.1	0.2	0.37
ODI	11.7	5.2	<u>3.5</u>	6.80
RDI-CFM	7.9	2.6	1.7	4.07
RDI-FTM	<u>12.8</u>	<u>5.3</u>	3.2	<u>7.10</u>
RDI-FTM-E	17.1	6.3	3.6	9.00

Table 3.11: Success rates (%) of targeted attacks evaluated across three adversarially trained defenses. The surrogate model is the ensemble of RN-50 and Inc-v3. The best-performing results are highlighted in bold, while the second-best are indicated with underlining.

features within the feature space. To efficiently optimize the learnable perturbations, we introduce a momentum-based stochastic update strategy. Extensive experiments demonstrate that FTM achieves notable improvements over existing state-of-the-art methods, while incurring low computational overhead.

Chapter 4

Soft Dropout Adaptation for Transferable and Efficient Attacks

In this chapter, we explore a general strategy for adapting the input both within the content and beyond the boundary, aimed at improving adversarial transferability. While input transformations are widely used to improve adversarial transferability, existing methods often rely on a limited set of fixed, hand-crafted transformations, lacking adaptive and content-aware diversity. Inspired by the role of dropout in promoting generalization, we hypothesize that introducing learnable, structured randomness into the input space can similarly boost adversarial transferability. Our analysis shows that a differentiable, patch-wise soft mask outperforms simple random dropout in achieving this goal. Building on this insight, we introduce Soft Dropout Adaptation (SODA), a method that improves adversarial transferability by jointly adapting the internal image content and the surrounding spatial context. Specifically, SODA employs two complementary, co-adapted components: an internal, content-modulating soft mask and a surrounding, context-expanding visual prompt. Extensive experiments demonstrate that SODA outperforms state-of-the-art transfer-based attacks, achieving superior results on standard, adversarially trained, and defended models.

Notably, SODA requires no additional training data and incurs minimal computational overhead.

4.1 Introduction

Deep neural networks (DNNs) [66] have achieved remarkable success across a wide range of vision tasks, including image classification [44, 118], object detection [38, 99], and semantic segmentation [83, 100]. However, they are vulnerable to adversarial examples [40, 67], which are carefully crafted inputs that induce incorrect predictions. A particularly concerning property of adversarial examples is their adversarial transferability [143, 22]: adversarial examples generated for a white-box surrogate model can often mislead other unseen models, even when their model architectures and parameters differ. This transferability poses serious security risks, particularly in real-world applications where attackers lack access to the target model.

Recent transfer-based attacks enhance adversarial transferability using gradient smoothing [131, 22], feature-level manipulation [134, 51], and input-space transformations [143, 80]. Among these, input transformation-based attacks [132, 84, 23] are particularly effective, as they introduce *input diversity* during the attack process. The core idea is to perform data augmentation on the intermediate adversarial image in each attack iteration, thereby mitigating overfitting to the white-box surrogate model. Recently, SIA [133] uses block-wise shuffling, and BSR [129] extends it with block-wise rotation, which are simple spatial transformations that yield strong adversarial transferability.

Transfer-based attacks often suffer from overfitting to the white-box surrogate model, substantially limiting their transferability to black-box models. This occurs because the optimization process exploits features and gradients that are highly specific to the surrogate. This challenge is analogous to model overfitting in training DNNs, where models learn to rely on spurious correlations in the data. Dropout [46] helps

prevent overfitting by randomly deactivating neurons during model training. Inspired by this principle, we hypothesize that introducing structured, learnable randomness to the input space can similarly mitigate overfitting in the adversarial attacks. By dynamically masking regions of the input, we can prevent the attack from relying on any single, surrogate-specific artifact. Our empirical analysis leads to two key observations. First, random binary masks (i.e., hard dropout) fail to enhance adversarial transferability. In contrast, soft masks with continuous values between 0 and 1 yield better transferability. Second, patch-based masking significantly outperforms pixel-wise masking. These findings motivate us to introduce a patch-wise learnable soft mask. It acts as a differentiable adaptive dropout in the pixel space to guide the spatial transformation of the input image during the attack process.

Recognizing that the spatial scope of input diversification could be extended beyond the core image content, we introduce a complementary, learnable context-expanding border prompt that is applied to the image boundary, inspired by visual prompt tuning [59]. This design deliberately preserves the image region for the soft mask while utilizing the under-exploited border area. The visual prompt operates exclusively outside the soft mask region and can be optimized jointly with the mask in an end-to-end manner, requiring no additional data and incurring negligible overhead.

To this end, we propose Soft Dropout Adaptation (SODA), a unified framework that enhances transferability by dynamically co-adapting a patch-wise soft mask (for content modulation) and a border visual prompt (for context expansion) during the attack process. Specifically, SODA learns a soft mask and a visual prompt for each intermediate adversarial image. As shown in Figure 4.1, the mask modifies the image content while the prompt appears as a border. Both components are jointly optimized during each attack iteration. This dynamic, test-time adaptation ensures that they synergistically diversify the input specific to the intermediate adversarial image, introducing minimal computational overhead. Extensive experiments demonstrate that SODA consistently outperforms existing transfer-based attacks across various model

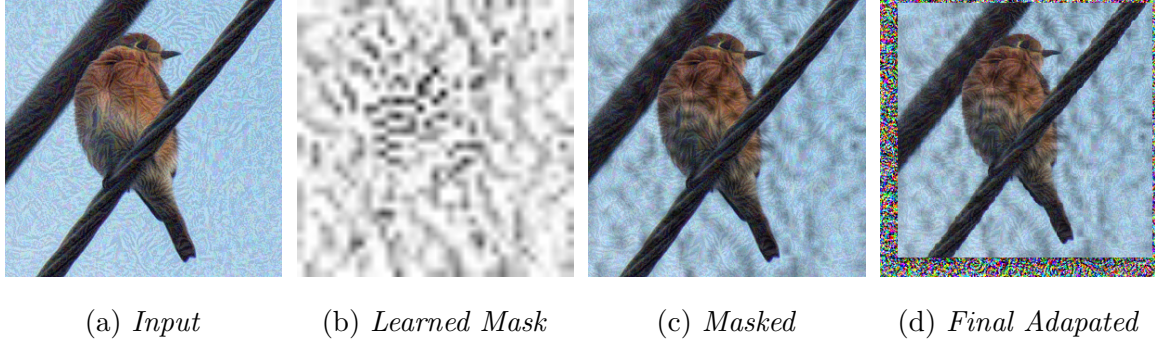


Figure 4.1: Overview of the proposed SODA. In each attack iteration, a soft mask and a visual prompt are jointly learned to modify the intermediate adversarial input. (a) Intermediate adversarial image; (b) Corresponding learned soft mask; (c) Input image after applying the soft mask; (d) Input image after applying the soft mask and the border prompt.

architectures and defense settings, achieving state-of-the-art transferability without extra data, additional models, or significant overhead.

4.2 Literature Review

Adversarial examples reveal a core vulnerability of DNNs: small, imperceptible changes to the input can cause misclassification [119, 40, 20, 96]. This vulnerability poses significant security risks, as these attacks can be precisely targeted [77] or even deployed in the physical world [29, 74]. Attack methods can be categorized by the attacker’s knowledge of the victim model. In white-box scenarios, where the attacker has full access to the victim model, powerful gradient-based attacks are commonly used. However, in more realistic black-box scenarios, where the victim model’s architecture and parameters are unknown, attackers must resort to other strategies. These include query-based attacks [10, 53], which are often costly, and transfer-based attacks [139, 43, 78], which generate an adversarial example on a surrogate model

and transfer it to the black-box victim.

Transfer-based Attacks. Transfer-based attacks often extend the Iterative Fast Gradient Sign Method (I-FGSM) [67, 40]. While I-FGSM is effective in white-box settings, its transferability to black-box models is limited due to overfitting to the surrogate model. To mitigate this, MI-FGSM [22] introduces momentum to stabilize updates, yet a significant performance gap between white-box and black-box attacks remains. Studying adversarial transferability is a key research area for evaluating and enhancing the robustness of DNNs.

Input transformation-based methods improve adversarial transferability by introducing input diversity during attacks. A common framework for transformation-based attacks generates multiple transformed copies of the input and averages their gradients from the surrogate model. DI [143] uses random resizing and padding; TI [23] uses spatial translation of pixels; and SI [80] augments inputs with pixel value scaling. Admix [132] mixes other images into the input, while SSA [84] perturbs the frequency domain of input. More recent methods such as SIA [133] and BSR [129] employ block-wise transformations: SIA shuffles and transforms blocks independently, and BSR further incorporates random block rotations.

Beyond input transformation, a range of techniques have been introduced to enhance transferability, including ensemble-based models and feature-level manipulations. Ensemble-based methods [22, 81] generate adversarial perturbations using multiple surrogate models. GN [76] simulates diverse surrogate models by perturbing skip connections and dropout layers. However, these methods do not leverage input diversity, and simulating multiple models introduces considerable computational overhead. Feature-level attacks [157, 139, 43, 134, 151] have also been proposed to improve transferability. TAP [157] improves transferability by maximizing feature-space distance. SGM [139] increases the contribution of skip connections during backpropagation. LinBP [43] alters gradient flow by skipping activation layers during backpropagation. FIA [134] and NAA [151] enhance transferability by identifying

and disrupting critical features in the feature space.

Defense Methods. Adversarial Training (AT) [86] improves robustness by incorporating adversarial examples during training. Ensemble AT [124] extends AT using multiple surrogate models for better robustness. Fast AT [138] reduces training cost by using single-step attacks. Random Smoothing (RS) [19] is a representative defense that provides provable robustness. Purification-based methods are also explored to recover clean images from adversarial examples. HGD [79] trains a denoising network with high-level feature guidance. NRP [90] uses self-supervised learning to train a denoising model. DiffPure [93] employs pre-trained diffusion models to remove adversarial perturbations via the diffusion process. These defense methods are model-agnostic and can be applied to various DNNs.

Dropout. Dropout methods [46, 128, 4, 34] are a family of regularization techniques used in training DNNs. Traditional binary dropout [46, 115] randomly deactivates neurons during training to reduce overfitting and improve generalization. While traditionally fixed, the dropout rate can be treated as a learnable parameter [64]. A random continuous dropout strategy [108] uses continuous masks sampled from uniform or Gaussian distributions. However, most existing dropout variants are primarily applied during model training and do not explore their effects in the pixel space or under adversarial transfer settings.

Visual Prompt. Visual Prompt Tuning (VPT) [59] introduces learnable patch tokens that are inserted into the image space or intermediate layers of pre-trained models. Bahng et al. [6] propose pixel-level perturbations to reprogram frozen models, inspired by adversarial reprogramming [28]. Other works integrate visual prompts with explicit annotations, such as red circles [109] or inpainting pairs [7], to guide model behavior. However, these methods require training datasets for prompt optimization, which limits their applicability to zero-shot settings. Test-time Prompt Tuning (TPT) [110] addresses this issue by optimizing prompts at inference time using multiple augmented views of a single test image. DiffTPT [32] extends TPT by

leveraging diffusion models to synthesize diverse augmented images. However, these methods are not designed for adversarial transferability, and generating multiple augmented images is computationally expensive.

4.3 Methodology

4.3.1 Preliminaries

Problem Formulation. Given a white-box surrogate model f_θ and a victim image x , the objective of a non-targeted adversarial attack is to find an adversarial example x^{adv} that can deceive f_θ :

$$\max_{x^{adv}} \mathcal{L}(T(x^{adv}), y; \theta) \quad \text{s.t.} \quad \|x^{adv} - x\|_\infty \leq \epsilon, \quad (4.1)$$

where $\mathcal{L}$ measures the distance between the predicted label of $T(x^{adv})$ and the ground-truth label y (i.e., cross-entropy), $T(\cdot)$ denotes input transformation, and ϵ is the maximum perturbation size. The attackers are assumed to possess a white-box surrogate model f_θ , but they cannot access the target black-box models. In this work, we assume that the attackers have no additional training data, which is a common assumption in transfer-based attacks.

Iterative Fast Gradient Sign Method (I-FGSM). Based on FGSM [40], which uses the gradient sign of input images to generate adversarial examples, I-FGSM [67] proposes an iterative version of FGSM, formulated as:

$$g_{t+1} = \nabla_x \mathcal{L}(T(x_t^{adv}), y; \theta), \quad (4.2)$$

$$x_{t+1}^{adv} = \text{Clip}_x^\epsilon \{x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1})\}, \quad (4.3)$$

where the $\text{Clip}_x^\epsilon(\cdot)$ function restricts the perturbation within the ϵ -ball (i.e., maximum perturbation budget), α is the step size, and x_t^{adv} is the intermediate adversarial image at the t^{th} iteration.

Momentum I-FGSM (MI-FGSM) [22] proposes using momentum to stabilize the update directions of I-FGSM and improve the transferability of adversarial examples:

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x \mathcal{L}(T(x_t^{adv}), y; \theta)}{\|\nabla_x \mathcal{L}(T(x_t^{adv}), y; \theta)\|}, \quad (4.4)$$

where g_t is the accumulated gradient at the t^{th} iteration, which is an extension of the gradient in Equation 4.2; μ is the momentum decay factor.

4.3.2 Motivation

Transfer-based adversarial attacks often suffer from overfitting to the surrogate model, which limits their transferability. Although input transformations can mitigate overfitting, existing methods often lack adaptive and content-aware diversity. Our core insight is that overcoming such overfitting requires diversifying the inputs along two complementary aspects: modulating the internal image content and expanding the surrounding spatial context.

For internal content modulation, we draw inspiration from dropout. Our empirical analysis reveals that a naive application, such as random binary dropout, is ineffective. We discovered two crucial insights: first, a soft mask with continuous values significantly outperforms hard masks; second, patch-based masking is far more effective than randomly dropping pixels (see Table 4.7, Figure 4.3). This motivated the first component of our method: a differentiable, patch-wise soft mask that adaptively modulates image content during the attack.

However, modifying the internal content alone is insufficient, as DNNs are also sensitive to an object’s broader context and framing. To exploit this, we introduce a mechanism to expand the input diversity beyond the original image boundaries. We propose our second component: a lightweight, learnable border prompt. This prompt surrounds the image, allowing the attack to manipulate the spatial context in a way that minimally interferes with the core object but can influence a model’s perception.

Integrating these two components into a unified framework leads to significantly improved transferability. We jointly optimize both the internal modulation mask and the external spatial prompt. Our pixel-space adaptation encourages the discovery of more fundamental and generalizable vulnerabilities, substantially improving transferability.

4.3.3 Proposed Soft Dropout Adaptation

We propose the Soft Dropout Adaptation (SODA) method, which integrates a learnable soft mask and a test-time adaptive visual prompt to improve adversarial transferability. We first introduce the soft mask and then the visual prompt.

Soft Mask. A key idea of the dropout mechanism is to randomly perturb the activations of neurons during training. Similarly, we learn to better perturb the input space with a soft mask. Thus, we propose learning a mask for each intermediate adversarial image in an adversarial manner, formulated as:

$$\max_{x^{adv}} \min_m \mathcal{L}(T(\text{Up}(m) \cdot x^{adv}), y; \theta), \quad (4.5)$$

where x^{adv} is the adversarial image or an intermediate adversarial image in any attack iteration, and m is a patch-wise soft mask with a smaller size than x^{adv} ; that is, each element in the mask corresponds to a patch in x^{adv} . $\text{Up}(\cdot)$ is the upsampling operation to match the size of x^{adv} , $T(\cdot)$ is the input transformation function, y is the ground-truth label of x , θ represents the parameters of the surrogate model, and $\mathcal{L}$ is the loss function that aligns with Equation 4.1. Notably, the generated x^{adv} satisfies $\|x^{adv} - x\|_\infty \leq \epsilon$, and the values of mask m are in $[0, 1]$.

Visual Prompt. To further enrich input diversity while preserving the effect of the above soft mask, we introduce a border-style visual prompt, which consists of lightweight, learnable parameters added around the image boundaries. This border prompt p can be learned for each input image x at test time without additional

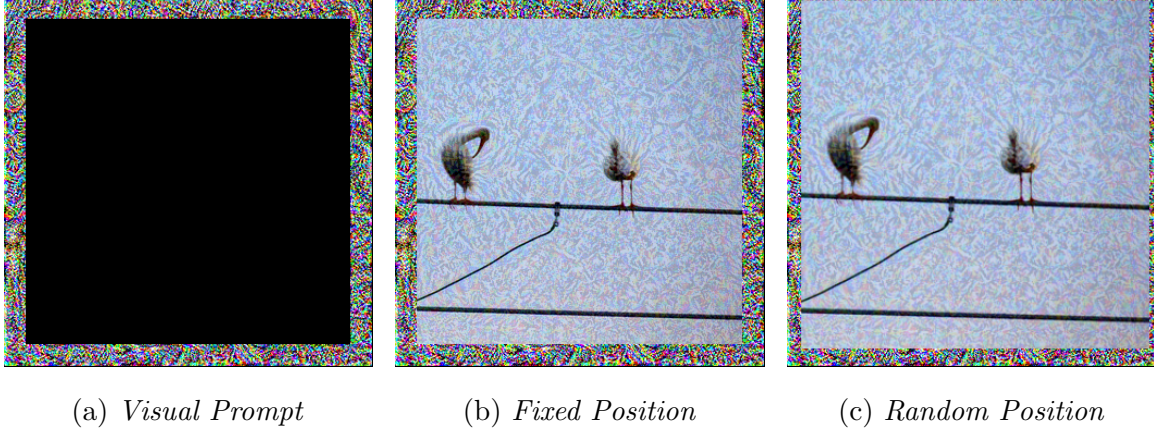


Figure 4.2: Illustration of visual prompt tuning. (a) Traditional learned prompt in pixel space; (b) Border prompt applied at a fixed position; (c) Border prompt applied at a random position.

data. The visual prompt and soft mask are jointly optimized during the attack. We formulate the final objective function of SODA as follows:

$$\max_{x^{adv}} \min_{m, p} \mathcal{L}(T((\text{Up}(m) \cdot x^{adv}) \oplus p), y; \theta), \quad (4.6)$$

$$\text{s.t.} \quad \|x^{adv} - x\|_{\infty} \leq \epsilon, \quad (4.7)$$

$$m \in [0, 1]^{1 \times H_m \times W_m}, \quad (4.8)$$

$$p \in [0, 1]^{3 \times H_p \times W_p}, \quad (4.9)$$

where $\oplus$ denotes the spatial concatenation of prompt p around the image x^{adv} . In addition, given that $x \in [0, 1]^{3 \times H \times W}$, the soft mask m is upsampled to $H \times W$ and multiplied by each channel of x^{adv} . In the implementation, we define a larger image canvas as the prompt p whose spatial dimensions exceed those of the input x^{adv} . During each attack iteration, we randomly select a location on p and paste the resized x^{adv} onto it, following a cut-out-style strategy. Thus, only the regions of p outside the pasted x^{adv} are treated as learnable and are updated during that attack iteration. An example of this process is provided in Figure 4.2. Specifically, Figure 4.2c shows our visual prompt applied at a random position, where x^{adv} is slightly enlarged to fit

the random position.

SODA algorithm. We approximately solve the optimization problem in Equation 4.6 by computing gradients with respect to the adversarial image x^{adv} , the soft mask m , and the visual prompt p , and updating them jointly in each backpropagation step. We formulate the optimization process as follows:

$$g^x, g^m, g^p = \nabla_{x^{\text{adv}}, m, p} \mathcal{L}(T((\text{Up}(m) \cdot x^{\text{adv}}) \oplus p), y; \theta), \quad (4.10)$$

$$x^{\text{adv}} \leftarrow x^{\text{adv}} + \alpha \cdot \text{sign}(g^x), \quad (4.11)$$

$$m \leftarrow m - \beta \cdot \text{sign}(g^m), \quad (4.12)$$

$$p \leftarrow p - \gamma \cdot \text{sign}(g^p), \quad (4.13)$$

where α , β , and γ are the learning rates for the adversarial image, soft mask, and visual prompt, respectively. For updating the adversarial image x^{adv} , we can further integrate the momentum strategy in Equation 4.4 or other existing methods to stabilize the gradient.

We summarize the SODA algorithm in Algorithm 3. Following prior work, during each attack iteration, we randomly transform the input into N transformed images, then use the average gradient to update the adversarial image, soft mask, and visual prompt. For the input transformation function $T(\cdot)$, we randomly select a known function from an input transformation pool $\{T_i\}$, such as block-wise shuffling and block-wise rotation. SODA updates the adversarial image, soft mask, and visual prompt jointly without requiring additional forward or backward passes. Therefore, SODA is theoretically computationally efficient, which will be verified in the experimental section.

4.4 Experiments

4.4.1 Experimental Setup

Datasets. Following previous work [22, 133, 129], we evaluate the proposed SODA method using the ImageNet-compatible dataset from the NIPS 2017 adversarial challenge. It consists of one thousand images from the ILSVRC 2012 dataset [101] specifically curated for adversarial attack evaluation. Notably, these images cover almost all categories in the ImageNet dataset, and most DNNs can accurately classify them.

General Settings. We assess the transferability of adversarial examples under a non-targeted attack setting. The generated adversarial examples are constrained by the ℓ_∞ norm, with a perturbation limit of $\epsilon = 16$. We set the momentum decay factor μ to 1.0 and the learning rate $\alpha = 2$. The attack iteration is $K = 10$. All evaluated methods adopt the cross-entropy loss $\mathcal{L}$ as the optimization objective for generating adversarial examples.

Models. We evaluate our method on eight pre-trained DNN models, comprising four CNN-based models and four transformer-based models. The CNN-based models include RN-50 [44], DN-121 [49], MB-v2 [103], and Inc-v3 [118]. For transformer-based architectures, we employ Vision Transformer (ViT) [24], LeViT [41], ConViT [21], and Pooling-based ViT (PiT) [45]. We also use three secured DNNs: Inc-v3 ens3 (an ensemble of three Inception-v3 networks), Inc-v3 ens4 (an ensemble of four Inception-v3 networks), and IR-v2 ens (an ensemble of three IncRes-v2 networks). Adversarially trained models like these have been frequently used in earlier research [22, 23, 132].

Attacks. We comprehensively evaluate SODA against current transfer-based attack techniques: DI [143], TI [23], SI [80], Admix [132], SSA [84], SIA [133], and BSR [129]. Among these, SIA and BSR represent the current state-of-the-art in transfer-based attacks. We incorporate MI [22] as a fundamental baseline component across all attack methods. For clarity in presentation, we omit the ‘MI’ prefix when referring

to these methods. All comparative experiments strictly follow the original parameter settings proposed in their respective papers. Specifically: For DI, we set the transformation probability to 0.7. For TI, we employ a 5×5 convolution kernel. SI utilizes 5 scaled copies of the input image. In the case of Admix, we configure its weight as 0.2 then generate 3 mixed images (corresponding to $m_2 = 3$ in [132]). For SSA, we set the tuning factor $\rho = 0.5$ and std σ of noise as $16/255$. For SIA, we use 3 shuffled blocks and 10 random transformations. For BSR, we use 2 shuffled blocks and the maximum rotation angle of 24 degrees. We use $N = 20$ transformed copies for SSA, SIA, and BSR. We also evaluate the attack performance in the ensemble framework [81].

Implementation Details of SODA. For a fair comparison, we set the number of transformed images $N = 20$ and the attack iteration $K = 10$, strictly following the settings of SIA and BSR. We set the learning rates $\alpha = 2, \beta = 0.2, \gamma = 0.1$. The patch size of the soft mask is $s = 8$. Specifically, given an input image x with size $3 \times 299 \times 299$, the width and height of the soft mask are $H_m = W_m = \lfloor 299/8 \rfloor = 37$, i.e., $m \in [0, 1]^{1 \times 37 \times 37}$. The canvas size of the visual prompt is $H_p = W_p = 340$. For the random position strategy, we randomly enlarge the input image without exceeding the canvas size and then randomly paste it onto the canvas. The input transformation pool $\{T_i\}$ includes two types of transformations: block-wise shuffle and block-wise rotation. Their parameters are set according to SIA and BSR, respectively. For comparing the computational overhead, we fix the batch size at 20 for all methods and report the average time per image.

4.4.2 Attacking Unsecured Models

Comparison with existing methods. We compare SODA with existing transfer-based attack methods in Table 4.1, including DI, TI, SI, Admix, SSA, SIA and BSR. We use various surrogate models including RN-50, DN-121, Inc-v3 and MB-v2. The

generated adversarial examples are evaluated on eight target models with different architectures. SODA outperforms existing methods in all cases. For example, using DN-121 as the surrogate model, SODA outperforms Admix, SSA, SIA and BSR by 14.8%, 6.7%, 3.0% and 3.9%, respectively; using Inc-v3 as the surrogate model, SODA outperforms Admix, SSA, SIA and BSR by 19.1%, 14.0%, 4.0% and 7.3%, respectively. Among the eight target models, ViT and ConViT are the most challenging models to attack, and SODA achieves the highest attack success rate on these two models. The results show that SODA is effective in improving adversarial transferability.

Combination with DI-TI. It is a common practice to combine DI and TI to evaluate the generality of the attack. As shown in Table 4.2, our DI-TI-SODA achieves the best average success rate across different settings. When using RN-50 as the surrogate model, DI-TI-SODA’s average success rate outperforms DI-TI-SSA, DI-TI-SIA, and DI-TI-BSR by 8.5%, 2.1%, and 2.9%, respectively. Similar superiority is observed when using DN-121, Inc-v3, and MB-v2 as the surrogate models. These results demonstrate that SODA is effective in improving adversarial transferability when combined with DI and TI.

Combination with ensemble-based method. The ensemble-based attack framework leverages an ensemble of multiple models as a surrogate model to generate adversarial examples. It is a widely used technique to boost the adversarial transferability of existing attacks. In Table 4.3, we use an ensemble of four models, including RN-50, DN-121, Inc-v3, and MB-v2, as the surrogate model. The results show that SODA achieves the highest average success rate, proving its effectiveness in improving adversarial transferability.

Source	Attack	RN-50	DN-121	Inc-v3	MB-v2	ViT	LeViT	ConViT	PiT	Average
RN-50	DI	100	97.0	76.1	95.3	29.7	32.4	19.8	29.8	60.0
	TI	100	90.5	59.8	87.2	31.6	29.0	18.9	25.7	55.3
	SI	100	95.9	71.3	90.7	27.3	24.5	16.2	23.4	56.2
	Admix	100	97.7	80.2	95.0	29.6	32.4	18.4	27.7	60.1
	SSA	100	98.2	88.3	98.4	42.3	54.1	29.8	45.3	69.5
	SIA	100	99.9	96.2	99.7	45.6	68.9	39.0	59.0	76.0
	BSR	100	99.9	95.5	99.6	48.8	71.6	40.6	60.7	77.1
	SODA	100	99.9	97.2	99.7	49.7	77.3	44.6	65.7	79.3
DN-121	DI	96.7	100	77.5	93.2	32.2	38.5	23.2	34.3	61.9
	TI	90.5	100	62.9	86.4	33.9	35.1	22.2	33.3	58.0
	SI	94.1	100	75.6	89.1	33.7	36.7	26.2	35.3	61.3
	Admix	96.8	100	84.5	94.5	38.9	47.4	28.8	41.9	66.6
	SSA	98.6	100	92.0	96.7	47.0	64.4	39.7	58.9	74.7
	SIA	99.7	100	96.6	99.2	50.5	73.1	42.9	65.5	78.4
	BSR	99.5	100	95.0	98.9	49.6	71.8	41.7	63.7	77.5
	SODA	99.6	100	96.9	99.4	56.2	80.8	48.0	70.5	81.4
Inc-v3	DI	74.0	74.7	99.6	79.5	29.5	30.8	20.8	29.3	54.8
	TI	59.3	61.7	100	67.8	30.1	25.7	19.1	26.8	48.8
	SI	69.5	71.7	100	76.3	31.9	31.0	23.3	33.4	54.6
	Admix	78.5	81.9	100	81.7	36.6	40.7	28.4	39.5	60.9
	SSA	82.1	85.9	99.7	85.1	40.6	51.3	33.2	49.7	66.0
	SIA	94.8	96.7	100	96.4	47.6	67.7	42.7	62.4	76.0
	BSR	91.8	95.8	100	94.0	44.8	58.8	38.8	57.4	72.7
	SODA	96.7	97.8	100	96.9	55.0	74.5	48.8	70.2	80.0
MB-v2	DI	89.8	88.3	70.0	100	28.2	34.0	17.9	30.1	57.3
	TI	78.4	75.8	53.1	100	28.5	29.0	18.8	24.5	51.0
	SI	83.2	84.6	64.0	100	27.2	28.8	18.1	26.6	54.1
	Admix	88.6	90.3	71.8	100	29.8	33.6	18.4	29.6	57.8
	SSA	91.4	89.3	72.0	100	33.3	41.3	21.8	34.3	60.4
	SIA	99.2	98.8	90.2	100	41.1	65.7	32.6	53.0	72.6
	BSR	99.1	99.3	95.2	100	47.2	73.6	38.7	62.5	77.0
	SODA	99.4	99.2	95.2	100	49.6	76.5	43.2	64.8	78.5

Table 4.1: Attack success rates (%) on eight target models. The best results are shown in bold.

Source	Attack	RN-50	DN-121	Inc-v3	MB-v2	ViT	LeViT	ConViT	PiT	Average
RN-50	DI-TI	100	98.2	82.0	96.8	38.7	46.1	27.2	38.4	65.9
	DI-TI-SI	100	99.4	93.7	97.9	48.0	62.4	37.3	53.0	74.0
	DI-TI-Admix	100	99.7	96.3	98.6	54.3	72.1	41.6	61.0	78.0
	DI-TI-SSA	100	99.1	91.7	98.3	55.2	62.3	40.5	57.4	75.6
	DI-TI-SIA	100	99.9	97.0	99.7	59.2	81.2	49.1	69.7	82.0
	DI-TI-BSR	100	99.9	96.3	99.8	58.8	78.3	48.2	68.2	81.2
	DI-TI-SODA	100	99.9	97.5	99.6	64.3	84.6	53.8	72.7	84.1
DN-121	DI-TI	96.4	100	81.3	94.4	39.5	47.7	28.2	43.2	66.3
	DI-TI-SI	98.1	100	90.5	96.8	50.9	64.8	40.4	59.8	75.2
	DI-TI-Admix	99.1	100	94.6	98.3	59.2	71.1	46.9	66.1	79.4
	DI-TI-SSA	98.7	100	93.7	97.2	57.0	73.4	48.7	69.7	79.8
	DI-TI-SIA	99.6	100	97.2	99.1	60.3	81.3	51.7	73.6	82.9
	DI-TI-BSR	99.5	100	95.2	98.5	56.5	75.0	46.0	68.4	79.9
	DI-TI-SODA	99.7	100	96.8	99.4	64.5	82.7	54.6	76.5	84.3
Inc-v3	DI-TI	73.8	78.1	99.5	79.3	34.7	38.8	25.8	35.8	58.2
	DI-TI-SI	83.3	88.5	100	86.1	42.2	50.6	36.7	48.7	67.0
	DI-TI-Admix	88.5	92.8	100	90.5	50.3	60.7	42.7	60.8	73.3
	DI-TI-SSA	81.7	87.5	99.8	85.7	47.3	58.5	41.3	55.6	69.7
	DI-TI-SIA	94.1	96.7	100	95.2	54.2	75.3	51.6	69.6	79.6
	DI-TI-BSR	90.4	95.3	100	93.4	47.6	62.9	41.1	61.0	74.0
	DI-TI-SODA	96.3	97.9	100	96.9	57.5	78.4	53.6	73.2	81.7
MB-v2	DI-TI	92.9	93.3	76.9	100	36.5	48.8	24.6	38.7	64.0
	DI-TI-SI	96.7	98.1	91.7	100	48.5	67.4	35.3	57.1	74.3
	DI-TI-Admix	98.6	99.0	93.8	100	53.7	73.8	38.5	61.4	77.3
	DI-TI-SSA	95.2	94.9	83.4	100	46.8	58.2	32.5	52.0	70.4
	DI-TI-SIA	99.2	99.2	94.2	100	53.6	78.2	43.6	64.1	79.0
	DI-TI-BSR	99.0	99.2	95.3	100	56.8	80.4	44.8	67.9	80.4
	DI-TI-SODA	99.5	99.6	96.2	100	59.0	84.2	49.5	72.7	82.6

Table 4.2: Attack success rate (%) of combining DI-TI with existing methods against eight target models. The best results are shown in bold.

4.4.3 Attacking Secured Models

We evaluate the effectiveness of SODA on recent representative defense methods, including HGD, AT, RS, NRP and DiffPure. The evaluation is conducted under a black-box setting, in which the attacker has no access to the defense details. We em-

Attack	RN-50*	DN-121*	Inc-v3*	MB-v2*	ViT	LeViT	ConViT	PiT	Average
DI	100*	100*	99.6*	100*	41.5	71.6	40.6	62.1	76.9
TI	100*	100*	99.7*	100*	42.0	59.6	38.1	53.4	74.1
SI	100*	100*	100*	100*	46.6	65.4	41.0	60.1	76.6
Admix	100*	100*	99.9*	100*	49.5	71.0	43.7	63.3	78.4
SSA	100*	100*	99.9*	100*	65.2	89.0	63.7	82.9	87.6
SIA	100*	100*	100*	100*	68.7	94.3	69.6	89.1	90.2
BSR	100*	100*	99.9*	100*	68.3	93.6	67.8	85.6	89.4
SODA	100*	100*	99.9*	100*	73.5	95.5	72.2	90.8	91.5

Table 4.3: Attack success rate (%) using an ensemble of models as the surrogate model against eight target models. * indicates the surrogate models. The best results are shown in bold.

ploy three surrogate settings: RN-50, DN-121, and an ensemble of these two models. Since HGD, NRP and DiffPure are plug-and-play purification methods, they can be applied to protect any given DNN model. We select two target models as protected models, ViT and ConViT, as they are the most challenging models to attack according to Table 4.1. The experimental results are reported in Table 4.4. The first row shows the prediction error rates of the target defense methods without any attack. Our SODA achieves remarkable attack performance across all settings, achieving the highest average success rate. The results demonstrate that SODA is an effective method for attacking secured models, highlighting the urgent need to develop more robust defense methods.

We also evaluate the effectiveness of SODA on three commonly used ensemble-based adversarially trained models, including Inc-V3_{ens3}, Inc-V3_{ens4} and IR-V2_{ens}. The experimental results are reported in Table 4.5, which shows that SODA also outperforms existing methods in this case. For example, using RN-50 as the surrogate model, SODA’s average success rate outperforms Admix, SSA, SIA and BSR by 18.9%, 7.6%, 6.4% and 2.4%, respectively.

Source	Attack	HGD	AT	RS	Protected Model: ViT			Protected Model: ConViT			Average
					HGD	NRP	DiffPure	HGD	NRP	DiffPure	
–	–	0.7	36.7	19.9	11.9	18.7	15.3	5.1	10.3	7.6	14.0
RN-50	DI	22.8	38.7	28.0	18.9	28.8	20.0	12.2	17.3	14.9	22.4
	TI	25.0	38.7	30.8	23.4	30.1	21.1	14.1	17.9	16.4	24.2
	SI	21.1	38.3	27.8	19.1	29.1	18.1	11.5	17.3	12.9	21.7
	Admix	26.1	38.4	28.8	21.1	29.6	17.7	12.8	18.2	14.9	23.1
	SSA	33.9	41.6	37.7	23.8	37.4	24.1	15.3	25.6	19.8	28.8
	SIA	63.5	41.0	37.2	29.6	36.9	24.5	25.1	25.6	19.9	33.7
	BSR	63.7	40.9	36.0	34.0	37.3	26.0	29.1	27.9	22.9	35.3
	SODA	66.0	41.8	39.2	35.3	39.3	27.8	31.6	29.8	23.9	37.2
DN-121	DI	20.5	38.5	28.0	21.1	31.5	20.4	13.2	19.4	14.7	23.0
	TI	24.0	38.6	29.8	25.6	30.4	22.7	16.4	18.7	16.5	24.7
	SI	21.5	39.4	30.1	21.9	31.3	19.2	14.7	20.6	15.2	23.8
	Admix	29.4	39.7	31.7	24.4	33.0	21.3	17.8	19.9	16.2	25.9
	SSA	43.4	41.2	37.8	28.3	37.9	27.1	21.2	26.7	21.7	31.7
	SIA	55.9	40.7	36.0	31.5	37.1	26.3	26.2	26.1	22.5	33.6
	BSR	58.9	40.5	35.9	35.3	36.7	26.7	31.9	25.5	22.8	34.9
	SODA	61.3	41.9	39.7	36.8	39.2	29.5	32.7	29.0	23.7	37.1
Ensemble	DI	38.5	40.4	32.9	23.7	34.6	22.9	19.6	24.8	19.3	28.5
	TI	52.9	40.7	35.8	30.8	35.0	26.1	27.1	26.3	23.1	33.1
	SI	48.7	41.6	35.9	26.4	35.7	23.2	21.4	26.6	20.9	31.2
	Admix	50.2	42.1	37.4	28.0	37.8	25.9	22.5	26.4	21.0	32.4
	SSA	67.1	45.3	48.9	35.7	47.1	38.0	34.1	43.5	33.2	43.7
	SIA	82.1	45.1	44.5	41.4	44.6	33.3	43.3	39.2	32.0	45.1
	BSR	85.7	44.0	44.0	49.9	44.6	34.6	51.0	38.5	31.5	47.1
	SODA	87.5	45.5	49.1	50.2	48.1	38.3	52.4	45.1	34.8	50.1

Table 4.4: Success rates (%) against various defense methods. The best results are shown in bold.

4.4.4 Efficiency and Complexity Analysis

SODA introduces only a small number of additional parameters that require gradient computation. Specifically, the soft mask requires $\lfloor 299/8 \rfloor^2 = 37 \times 37 = 1369$ learnable parameters, which is less than 1% compared to the original image size of $3 \times 299 \times 299 = 268,203$. The visual prompt introduces $3 \times 340 \times 340 - 3 \times 299 \times 299 = 78,597$ additional parameters. Notably, the parameters involved in gradient compu-

tation during the backward pass of DNNs are dominated by the model weights, which typically exceed one million parameters. For example, the number of parameters in Inc-v3 is 23.8M, and the added parameters of SODA constitute only 0.3% of the model parameters. Thus, the parameters introduced by our soft mask and visual prompt are negligible in comparison, resulting in minimal theoretical computational overhead.

Table 4.6 reports the average runtime for generating an adversarial image using SODA on an NVIDIA RTX 3090 GPU. For fair comparison, we set the batch size to 20, attack iteration to 10, and the number of transformed images to 20 for all methods. The results demonstrate that SODA is highly efficient, and the overhead introduced by the soft mask and visual prompt is marginal.

4.4.5 Ablation Study

Soft mask vs. hard mask: ablation on the type of mask. Table 4.7 shows the impact of different types of masks on the attack success rate. All experiments are conducted without the visual prompt to clearly show the effect of the mask. We run each experiment 5 times and report the average results. For the random soft mask, we use a Gaussian distribution to generate the mask, with mean μ and standard deviation σ . We clamp the mask to be in the range of $[0, 1]$. For the random hard mask, we use a random binary mask with a drop rate ranging from 0.1% to 40%, which represents the pixel-space level of dropout. The results show that the random hard mask does not help improve the attack success rate. Additionally, the performance of the random mask drops significantly when the drop rate exceeds 1%. Thus, the random hard mask, i.e., the spatial version of dropout, is ineffective in improving adversarial transferability. For the random soft mask, the performance is also sensitive to the mean μ and standard deviation σ . When $\mu = 1.00$ and σ is small, the improvement is negligible. Therefore, the results validate that existing masking

methods are not suitable for improving adversarial transferability. One advantage of the soft mask is that it is differentiable, which allows us to optimize the mask using gradient descent. Notably, our *learnable* soft mask is more effective than the *random* one, and the term ‘soft’ in ‘soft mask’ is meaningful for the gradient-based optimization in SODA.

Patch-based mask vs. pixel-based mask. Figure 4.3 presents the effect of the mask’s patch size on the attack success rate. We aim to explore whether a patch-based mask is more effective than a pixel-based one. When the patch size is set to 1, the mask becomes pixel-based, meaning that each element of the mask corresponds to a single pixel of the target input image. In contrast, our SODA framework adopts a patch size of 8, where each element in the mask represents an 8×8 patch of the input. This experiment empirically validates the motivation for using a patch-based design. As shown in the figure, using a patch size larger than 1 significantly improves the attack success rate, indicating that operating at the patch level helps capture more structured and spatially coherent features. The patch-based strategy consistently outperforms its pixel-level counterpart, demonstrating its superiority in optimization and transferability. Moreover, increasing the patch size beyond 1 results in relatively stable performance, with the best results observed around patch size 8. These findings justify our choice of patch-based masking as a more effective and robust design choice than pixel-level masking.

Ablation on the components of SODA. We study the impact of the soft mask, the visual prompt, and the placement strategy of the visual prompt on the attack success rate. The random position strategy involves randomly selecting a location within the visual prompt canvas to place the target input image. As shown in Table 4.8, employing the soft mask alone improves the attack success rate by 2.3% (from 76.8% to 79.1%). In contrast, using only the visual prompt with random placement yields a 77.9% success rate, improving the baseline by only 1.1%. These results indicate that the soft mask is more effective than the visual prompt. We further analyze the specific

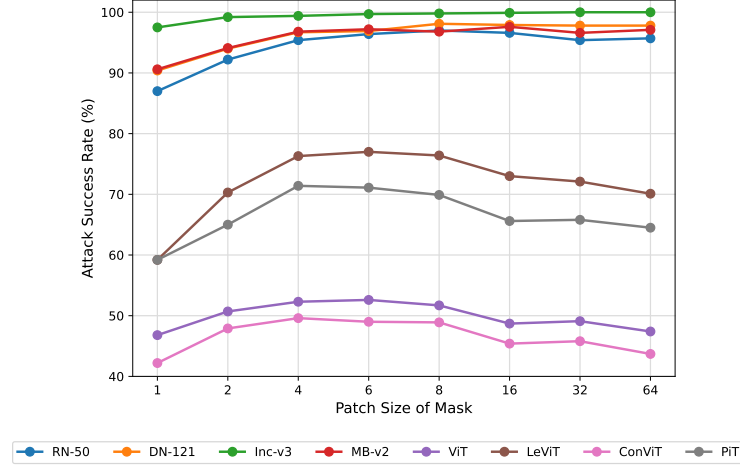


Figure 4.3: Patch-based mask vs. pixel-based mask. Impact of different patch sizes of our mask on the success rate of SODA. When the patch size is 1, the mask becomes pixel-based.

impact of random placement of the visual prompt. Incorporating random placement offers negligible improvements to the attack performance of either the soft mask or the baseline. For example, applying random placement to the soft mask increases the success rate by only 0.1% (from 79.1% to 79.2%), while the baseline sees a 0.2% increase (from 76.8% to 77.0%). Adding the visual prompt to the soft mask improves the success rate by 0.8% (from 79.2% to 80.0%), suggesting that the visual prompt is more effective than random placement. In summary, each component of SODA contributes to the overall success rate, with the learnable soft mask being the most effective.

Parameter study on the visual prompt. Table 4.9 presents the effect of the canvas size ($H_p \times W_p$) of the visual prompt on the attack success rate. In this analysis, we fix $H_p = W_p$ and set them larger than the target input image, which has a shape of $3 \times 299 \times 299$. Therefore, we vary the canvas size from 310 to 400 and report the average attack success rate across eight target models. The results indicate that the attack success rate is relatively insensitive to the canvas size of the visual prompt.

The highest success rate is observed when the canvas size is around 340.

Parameter study on the number of attack iterations. Figure 4.4 illustrates the effect of the number of attack iterations on the attack success rate. We adopt Inc-v3 as the surrogate model and vary the number of attack iterations from 2 to 30. We report the attack success rates across eight target models. The results demonstrate that the attack success rate increases consistently with the number of iterations. To ensure fair comparison with existing methods, we follow their setting by fixing the number of attack iterations to 10.

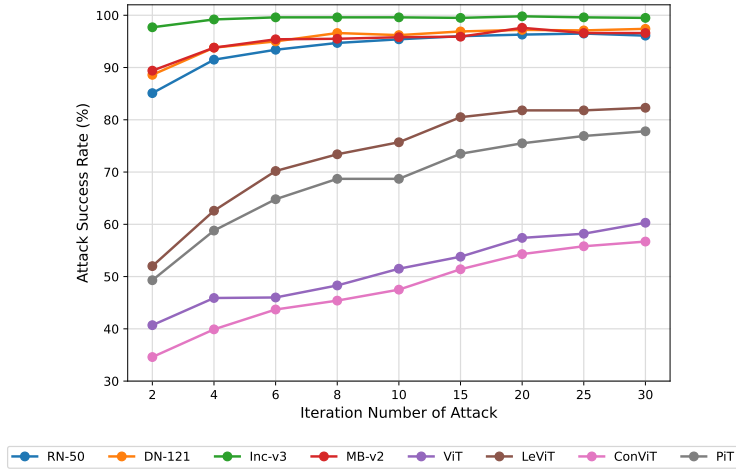


Figure 4.4: Impact of different numbers of attack iterations on the attack success rate of our SODA.

Parameter study on the learning rate of the mask optimization. Figure 4.5 illustrates the effect of the learning rate used in mask optimization on the attack success rate. Note that the number of optimization steps for the mask is aligned with the attack iterations, i.e., both are set to 10. Empirically, the learning rate for mask optimization tends to perform better when it is larger than $1/10 = 0.1$. We vary the learning rate from 0.05 to 1.0 and report the attack success rates across eight target models. The results indicate that the highest success rates are obtained when the learning rate is in the range of $[0.2, 0.4]$. Setting the learning rate to 1.0 results in

learning a hard binary mask. Based on these findings, we set the learning rate to 0.2 in our experiments.

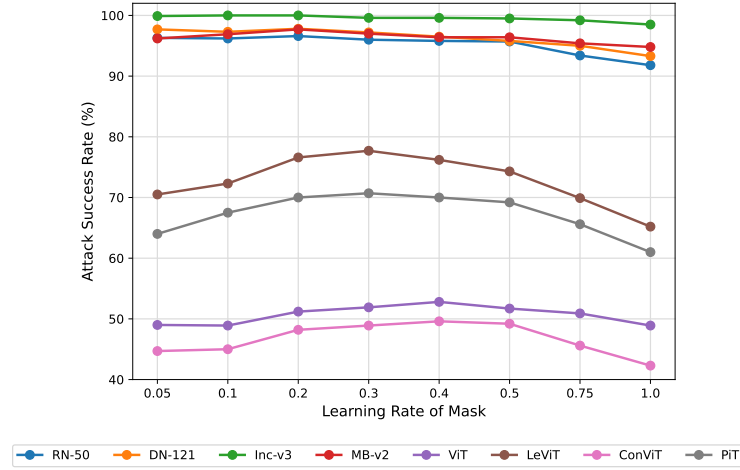


Figure 4.5: Impact of different learning rates of our mask optimization on the attack success rate of our SODA.

Figure 4.6 shows the adversarial examples generated by our SODA, where Inc-v3 is used as the surrogate model.

4.4.6 Visualization of SODA’s Augmented Images

Traditional iterative attacks progressively perturb a benign image, producing a series of intermediate adversarial images over multiple optimization steps. As illustrated in Figure 4.7, SODA integrates into this process by dynamically generating a soft mask and a visual prompt at each iteration, tailored to the current intermediate adversarial image. These components are applied to yield an augmented intermediate adversarial input, which is then used for loss computation and gradient updates.

Although the perturbations in intermediate adversarial examples are typically distributed across the entire image, some high-frequency patterns appear more visually salient near object contours. The learned soft masks adaptively suppress portions of these patterns, making them less conspicuous while maintaining the structural



Figure 4.6: Adversarial examples generated by SODA.

appearance of the underlying image. Interestingly, the soft masks tend to focus on regions surrounding key visual elements, such as objects and edges, indicating a form of structure-aware behavior, even though no explicit semantic guidance is provided. This suggests that such soft masking helps prevent overfitting to a limited set of vulnerable regions, potentially enhancing adversarial transferability across models.

4.5 Summary

In this work, we addressed the overfitting to the surrogate model in transfer-based adversarial attacks. We introduced Soft Dropout Adaptation (SODA), a novel framework that enhances transferability by diversifying inputs along two complementary aspects: modulating the internal image content and expanding the surrounding spa-



(a) *Benign Image* (b) *Adv. Input* (c) *Learned Mask* (d) *Masked* (e) *Final*

Figure 4.7: Visualization of the transformation process in our SODA. (a) The benign image; (b) the intermediate adversarial image during the iterative attack; (c) the learned soft mask corresponding to (b); (d) the masked intermediate adversarial image after applying the soft mask; (e) the final augmented input with both the soft mask and the visual prompt.

tial context. SODA implements this principle by jointly learning an internal soft mask for content modulation and a surrounding border prompt for context expansion. This adaptation is performed dynamically during the attack without additional data, allowing for practical deployment with minimal overhead. Extensive experiments demonstrate that SODA consistently outperforms state-of-the-art attacks across a wide range of standard, adversarially trained, and defended models.

Algorithm 3 Soft Dropout Adaptation (SODA)

Input: Surrogate model f_θ , loss function $\mathcal{L}$, benign image x with ground-truth label y , perturbation budget ϵ , momentum decay factor μ , attack iteration K , number of transformed images N , input transformation pool $\{T_i\}$

Input: Learning rate α, β, γ

Input: Patch size of soft mask s

Input: Canvas size of visual prompt H_p, W_p

Output: An adversarial example x^{adv}

- 1: Initialize adversarial image $x_0^{adv} = x$
- 2: Initialize momentum $g_0 = 0$
- 3: Obtain mask size $H_m = W_m = \lfloor \text{size}(x)/s \rfloor$
- 4: Initialize soft mask $m_0 = 1$
- 5: Initialize visual prompt $p_0 = 0$
- 6: **for** $t = 0 \rightarrow K - 1$ **do**
- 7: $\bar{g}^x = 0$ ▷ Gradients of N transformed images
- 8: $\bar{g}^m = 0$
- 9: $\bar{g}^p = 0$
- 10: **for** $j = 0$ **to** $N - 1$ **do**
- 11: Randomly select a function $T(\cdot)$ from $\{T_i\}$
- 12: Obtain gradients g^x, g^m, g^p using Equation 4.10
- 13: $\bar{g}^x = \bar{g}^x + g^x$
- 14: $\bar{g}^m = \bar{g}^m + g^m$
- 15: $\bar{g}^p = \bar{g}^p + g^p$
- 16: **end for**
- 17: Calculate momentum $g_{t+1} = \mu \cdot g_t + \bar{g}^x / \|\bar{g}^x\|_1$
- 18: Update example $x_{t+1}^{adv} = \text{Clip}_x^\epsilon(x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1}))$
- 19: Update soft mask $m_{t+1} = \text{Clip}(m_t - \beta \cdot \text{sign}(\bar{g}^m), 0, 1)$
- 20: Update visual prompt $p_{t+1} = \text{Clip}(p_t - \gamma \cdot \text{sign}(\bar{g}^p), 0, 1)$
- 21: **end for**
- 22: **return** x_K^{adv} .

Source	Attack	Inc-v3 _{ens3}	Inc-v3 _{ens4}	IR-v2 _{ens}	Avg
RN-50	DI	27.3	24.3	12.0	21.2
	TI	25.6	25.4	14.3	21.8
	SI	29.9	26.6	12.3	22.9
	Admix	31.0	28.9	14.2	24.7
	SSA	44.1	39.9	23.9	36.0
	SIA	46.0	42.3	23.2	37.2
	BSR	50.4	45.3	27.9	41.2
	SODA	53.2	48.3	29.4	43.6
DN-121	DI	27.8	26.1	13.6	22.5
	TI	29.1	26.4	14.9	23.5
	SI	36.3	33.1	16.6	28.7
	Admix	40.7	38.0	18.8	32.5
	SSA	56.5	47.8	30.3	44.9
	SIA	47.3	43.6	24.2	38.4
	BSR	50.7	46.5	28.0	41.7
	SODA	56.8	50.8	30.4	46.0
Ens	DI	44.5	39.7	21.2	35.1
	TI	46.0	41.1	25.9	37.7
	SI	55.7	52.6	29.1	45.8
	Admix	61.6	55.0	32.3	49.6
	SSA	78.3	72.6	51.8	67.6
	SIA	77.2	69.0	47.0	64.4
	BSR	77.0	70.0	48.8	65.3
	SODA	79.7	73.5	52.7	68.6

Table 4.5: Attack success rates (%) against three ensemble-based adversarially trained defenses. The best results are shown in bold.

Method	Time (s)
SSA	0.986
SIA	0.823
BSR	0.729
SODA w/o mask and prompt	0.774
SODA	0.797

Table 4.6: Average time for generating an adversarial image.

Ablation	RN-50	DN-121	Inc-v3	MB-v2	ViT	LeViT	ConViT	PiT	Average
Our Soft Mask	96.5	97.8	100	96.9	50.3	74.2	47.9	69.1	79.1
w/o Mask	96.3	98.0	100	96.6	48.0	69.5	43.6	62.7	76.8
w/ Random Soft Mask ($\mu = 1.00, \sigma = 1.00$)	78.6	78.3	92.4	89.2	34.7	36.6	21.5	32.5	58.0
w/ Random Soft Mask ($\mu = 1.00, \sigma = 0.10$)	96.2	97.6	100	96.2	48.3	70.0	43.7	66.0	77.3
w/ Random Soft Mask ($\mu = 1.00, \sigma = 0.01$)	96.0	97.9	100	96.6	47.6	69.6	43.7	65.6	77.1
w/ Random Soft Mask ($\mu = 0.75, \sigma = 1.00$)	70.5	68.6	84.7	84.5	29.8	30.6	19.1	26.8	51.8
w/ Random Soft Mask ($\mu = 0.50, \sigma = 1.00$)	61.3	58.4	73.3	77.8	28.3	23.9	15.4	23.7	45.3
w/ Random Soft Mask ($\mu = 0.25, \sigma = 1.00$)	53.8	53.0	61.8	72.1	28.1	19.6	12.3	21.0	40.2
w/ Random Hard Mask (drop 0.1%)	95.0	96.8	99.9	96.6	46.8	70.4	44.8	65.3	76.9
w/ Random Hard Mask (drop 1%)	94.7	95.8	99.9	96.4	46.8	67.6	40.0	58.8	75.0
w/ Random Hard Mask (drop 10%)	74.0	72.6	95.7	84.1	34.5	39.0	23.6	35.4	57.3
w/ Random Hard Mask (drop 20%)	50.2	51.9	77.7	65.5	28.9	23.3	16.5	22.9	42.1
w/ Random Hard Mask (drop 40%)	31.9	31.0	39.8	45.5	23.2	13.1	9.7	14.7	26.1

Table 4.7: Soft mask vs. hard mask: ablation study on the type of mask. The surrogate model is Inc-v3. μ and σ are the mean and standard deviation of the Gaussian distribution. The best results are shown in bold.

Soft Mask	Visual Prompt	Random Position	RN-50	DN-121	Inc-v3	MB-v2	ViT	LeViT	ConViT	PiT	Average
✓	✓	✓	96.7	97.8	100	96.9	55.0	74.5	48.8	70.2	80.0
✓		✓	95.4	97.3	99.6	96.1	53.5	73.7	48.6	69.3	79.2
✓			96.5	97.8	100	96.9	50.3	74.2	47.9	69.1	79.1
	✓	✓	93.8	96.3	99.9	96.2	52.9	70.7	45.7	67.9	77.9
		✓	94.1	95.9	99.8	95.7	51.0	69.9	44.6	64.6	77.0
			96.3	98.0	100	96.6	48.0	69.5	43.6	62.7	76.8

Table 4.8: Ablation study on the components of SODA. The surrogate model is Inc-v3. The best results are shown in bold.

Ablation	Canvas Size of Visual Prompt									
	310	320	330	340	350	360	370	380	390	400
w/ soft mask	79.74	79.79	79.90	79.99	79.73	79.26	78.95	78.57	78.28	77.85
w/o soft mask	77.26	77.85	77.91	77.89	77.81	77.80	77.42	77.05	76.94	76.35

Table 4.9: Ablation study on the canvas size of the visual prompt in our SODA. The surrogate model is Inc-v3.

Chapter 5

Perspective-Invariant Attack with Enhanced Adversarial Transferability

In this chapter, we investigate the limitations of existing transfer attacks based on input transformations. While input transformation techniques are commonly employed to enhance adversarial transferability, existing methods rely on fixed perspective transformations, limiting adversarial transferability. To this end, we propose a novel perspective-invariant attack (PIA) to generate highly transferable adversarial examples by making them invariant to perspective transformations. Our PIA introduces random perspective transformations to the input images before feeding them into surrogate models, significantly increasing the diversity of transformed images. Specifically, we use stochastic perspective transformation and other transformation techniques to form a transformation pool. Then, we randomly sample a transformation function from the pool to transform the input image in each attack iteration. Through extensive evaluation involving various DNN models, defensive mechanisms, and large language models (LLMs), we demonstrate that PIA significantly outper-

forms existing transfer-based attacks in different scenarios. Notably, with ResNet-50 as the source model, PIA achieves average performance gains of 13.9% on three widely used adversarially trained DNNs compared to state-of-the-art methods.

5.1 Introduction

Over the past decade, deep neural networks (DNNs)[66, 44, 118] have demonstrated outstanding performance and have been extensively applied in high-stakes fields such as autonomous driving[37], medical image analysis[100], and face recognition[105]. However, despite their success, DNNs are known to be vulnerable to adversarial examples [40, 67, 29, 86], which are carefully designed inputs containing subtle perturbations that can lead the model to produce incorrect predictions. More concerning, these adversarial examples exhibit cross-model transferability [143, 22, 132], where adversarial examples generated against one surrogate model can successfully fool other unknown target models without requiring access to target models’ parameters or architectures. This transferability property makes adversarial examples particularly threatening, as attackers can leverage surrogate models to launch practical black-box attacks against deployed DNN systems.

Adversarial attacks can be classified into white-box and black-box categories depending on the attacker’s level of access to the victim model [40, 88, 14, 10, 22, 143, 132]. In white-box scenarios, the adversary has full access to the model’s parameters and architecture. In contrast, black-box attacks assume no such access, making them inherently more difficult to execute. A common strategy in black-box settings leverages the transferability of adversarial examples: an effect where adversarial inputs generated from a surrogate model can also mislead unseen target models [22, 143, 132]. These so-called transfer-based attacks operate without interacting with the target model, but their effectiveness is often limited by overfitting to the surrogate model. When there is a significant architectural difference between the surrogate and target

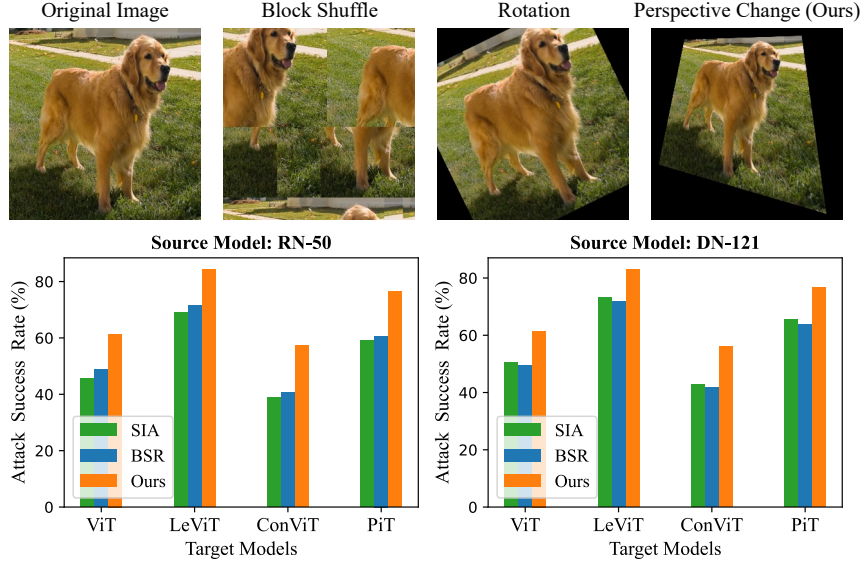


Figure 5.1: Our method significantly outperforms state-of-the-art attacks including SIA [133] and BSR [129]. SIA uses block-wise shuffling, while BSR uses block-wise shuffling combined with slight rotations, both of which are fixed perspective transformations. We consider altering the perspective of input images to increase the diversity of transformed images.

models, the success rate of transfer-based attacks tends to decline sharply.

Transfer-based attacks can be categorized into input transformation, model enhancement, and optimization improvements. Notably, input transformation-based techniques have been widely adopted in various attack methods. Representative input transformation-based attacks include DI [143], TI [23], SI [80], Admix [132], and SSA [84]. Recently, SIA [133] and BSR [129] have achieved state-of-the-art performance by introducing block-wise shuffling and block-wise shuffling combined with slight rotations, respectively. The idea behind input transformation-based attacks is that effective transformations can increase the diversity of transformed images [133], thereby mitigating overfitting to the surrogate model and improving the transferability of adversarial examples. However, existing input transformation-based attacks have not considered perspective changes in input images, which may limit the trans-

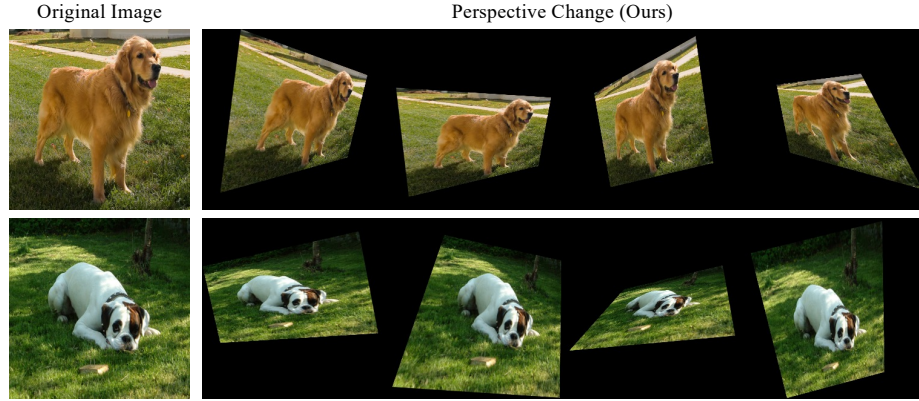


Figure 5.2: Visualization of perspective transformations applied to input images using our PIA method.

ferability of adversarial examples. Figure 5.1 shows that SIA and BSR use fixed perspective transformations, such as block shuffling and slight rotations, while our method can simulate diverse perspectives of input images. We believe that appropriately transforming the perspectives of input images can further increase input diversity and consequently enhance the transferability of adversarial examples.

In this work, we propose a novel perspective-invariant attack (PIA) to increase adversarial transferability by making adversarial examples invariant to perspective transformations. Specifically, we use perspective transformations along with other transformation methods to form a transformation pool. In each attack iteration, we randomly select a transformation from the pool to augment the input image. Figure 5.2 demonstrates the perspective changes of input images in our attack. We randomly modify the positions of the original image’s four corner points to simulate diverse perspectives of input images. This transformation increases the diversity of input images while preserving the semantics of the original images. Unlike existing input transformation-based attacks, which typically preserve parallel structures in the original image, perspective transformations maintain only straight-line properties. Theoretically, perspective transformation increases the degrees of freedom in input generation, enabling the creation of more diverse adversarial examples. We con-

duct extensive experiments in different scenarios, including target DNN models with different architectures, combinations with existing attack techniques, advanced defensive mechanisms, and multimodal large language models (LLMs). Empirical results demonstrate that PIA significantly improves adversarial transferability in different scenarios while maintaining computational efficiency.

5.2 Literature Review

Although DNNs have made significant progress in tasks such as image classification, segmentation, and detection, adversarial attacks [40, 119] have revealed their vulnerability: adding small perturbations to the input can cause DNNs to make incorrect predictions. To investigate the vulnerability of DNNs, researchers have proposed various adversarial attack methods aimed at generating adversarial examples that can mislead DNNs in different scenarios, including targeted attacks [156, 77], generative attacks [96, 89], physical-world attacks [29, 74], query-based attacks [10, 53], and transfer-based attacks [22, 143]. Among various attack strategies, transfer-based attacks involve crafting adversarial examples using a white-box surrogate model and then applying them to fool black-box models. Such attacks present substantial security risks for DNNs deployed in real-world systems, as they require no access to the internal details of the target model. This paper focuses on transfer-based attacks [132, 139, 43, 78, 51, 20], particularly on improving adversarial transferability when there is a significant architectural gap between the surrogate and target models.

Transfer-based Adversarial Attacks. Transfer-based adversarial attacks commonly build upon the Iterative Fast Gradient Sign Method (I-FGSM) [67], an extension of the Fast Gradient Sign Method (FGSM) [40]. While I-FGSM shows strong performance in white-box settings with knowledge of target models, it performs poorly in black-box scenarios where target models are unknown. To address this limitation, the Momentum-based I-FGSM (MI-FGSM) [22] incorporates momentum terms to

stabilize gradient updates. Nevertheless, a considerable performance gap persists between white-box and black-box attack scenarios, highlighting the need for more powerful transfer-based methods.

Input transformation-based attacks are a class of methods that aim to generate diverse input images to enhance adversarial transferability. Diverse Input (DI) [143] employs random resizing and padding operations to generate diverse input images. Translation-Invariant (TI) [23] focuses on the translation property of input images and introduces a convolutional kernel to process gradients. Scale-Invariant (SI) [80] leverages the loss preservation property of pixel value scaling and utilizes multiple scaled images for data augmentation. The Admix method [132] combines small portions of other images to create multiple transformed images. Spectrum Simulation Attack (SSA) [84] perturbs the frequency domain of input images using Gaussian noise and random masking. More recently, Structure Invariant Attack (SIA) [133] and Block Shuffle and Rotation (BSR) [129] have achieved state-of-the-art performance through input transformation. Specifically, SIA introduces block-wise shuffling combined with random input transformations on each block, while BSR extends the block shuffling concept by incorporating slight rotations of individual blocks. Learn to Transform (L2T) [158] proposes a learning-based framework to search for the optimal transformation for each input image.

In addition to input transformation-based attacks, researchers have proposed various other transfer-based attacks. Ensemble-based methods [22, 81] leverage multiple surrogate models simultaneously to generate more transferable adversarial examples. Ghost networks [76] simulate various surrogate networks using perturbed skip connections and dropout layers. Transferable Adversarial Perturbation (TAP) [157] enhances transferability by maximizing the feature-space distance between benign and adversarial examples. Skip Gradient Method (SGM) [139] analyzes ResNet-based architectures and incorporates more gradients from skip connections. LinBP [43] introduces skipping of activation layers during the backpropagation process. Feature

Importance Aware (FIA) attack [134] focuses on corrupting critical features at the feature level, while Neuron Attribution-based Attack (NAA) [151] proposes a feature-level attack by utilizing advanced neuron attribution.

Defenses against Adversarial Examples. The High-Level Representation Guided Denoiser (HGD) [79] proposes a denoising network trained to restore clean images from perturbed ones. Adversarial Training (AT) [86] enhances the adversarial robustness of DNNs by incorporating adversarial examples into the training process. Building upon AT, Ensemble Adversarial Training [124] leverages multiple surrogate models to achieve better attack performance and robustness. To address the computational cost of adversarial training, Fast Adversarial Training [138] demonstrates that using single-step attacks can provide reasonable robustness with significantly reduced training time. Random Smoothing (RS) [19] represents a major advance in certified defenses by providing provable robustness guarantees against adversarial perturbations. The Neural Representation Purifier (NRP) [90] employs self-supervised learning techniques to train a denoising network that can purify adversarial examples. More recently, Diffusion-based Purification (DiffPure) [93] leverages pre-trained diffusion models to effectively remove adversarial perturbations through the diffusion process.

5.3 Methodology

5.3.1 Problem Formulation

For transfer-based attacks, the attacker is assumed to have a white-box surrogate model F , which is a pre-trained image classifier in this paper. The attacker is assumed to have no access to additional training datasets. Given a benign image x , a surrogate model F , and a ground-truth label y , the goal of the attack is to generate an adversarial example x^{adv} that is misclassified by F . The difference between x and

x^{adv} should be subtle, and following previous work, we use the ℓ_∞ -norm constraint. The objective of the attack can be formulated as:

$$\arg \min_{x^{\text{adv}}} \mathcal{L}(F(x^{\text{adv}}), y), \quad s.t. \quad \|x - x^{\text{adv}}\|_\infty \leq \epsilon, \quad (5.1)$$

where $\mathcal{L}$ is the adversarial loss function and ϵ is the perturbation budget.

5.3.2 Preliminary

MI-FGSM. To solve the problem above, existing methods usually adopt the iterative gradient-based technique, e.g., the MI-FGSM [22], formulated as follows:

$$g_{i+1} = \nabla_x \mathcal{L}(F(x_i^{\text{adv}}), y), \quad (5.2)$$

$$\tilde{g}_{i+1} = \mu \cdot g_i + g_{i+1} / \|g_{i+1}\|_1, \quad (5.3)$$

$$x_{i+1}^{\text{adv}} = x_i^{\text{adv}} - \eta \cdot \text{sign}(\tilde{g}_{i+1}), \quad (5.4)$$

$$x_{i+1}^{\text{adv}} = \text{Clip}_{x, \epsilon}(x_{i+1}^{\text{adv}}), \quad (5.5)$$

where g_i is the gradient of the loss function with respect to the input image x in the i -th iteration. μ is the momentum decay factor, and η is the step size. x_{i+1}^{adv} is clipped within ϵ bound. Initially, we set $x_0^{\text{adv}} = x$ and $g_0 = 0$.

Input Transformation-based Attacks. To enhance the transferability of adversarial examples, recent studies have proposed various input transformation-based attacks. Notably, it has been shown that improving the *diversity* of transformed images can significantly improve the transferability of adversarial examples [143, 133]. These attacks use multiple copies of the input image to improve adversarial transferability, as shown in Figure 5.3. We formulate the general idea of input transformation-based attacks based on Equation 5.2 as follows:

$$g_{i+1} = \frac{1}{N} \sum_{j=1}^N \nabla_x \mathcal{L}(F(\phi_j(x_i^{\text{adv}})), y), \quad (5.6)$$

where $\phi_j(\cdot)$ is the j -th used input transformation function; N is the number of copies of the input image; x_i^{adv} is the intermediate adversarial example in the i -th iteration.

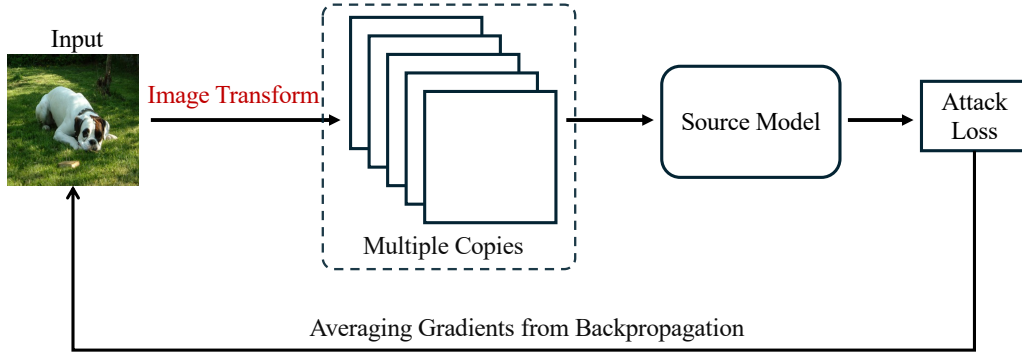


Figure 5.3: Overview of transformation-based attacks. Multiple transformed copies of the input image are used to calculate gradients for enhancing adversarial transferability.



Figure 5.4: Visualization of advanced input transformations for generating adversarial examples, including SIA and BSR.

Recently, SIA [133] and BSR [129] have achieved state-of-the-art performance in transfer-based attacks. To increase the diversity of transformed images, SIA introduces block-wise shuffling combined with random input transformations on each block. BSR also adopts the block-shuffling concept but incorporates slight rotations of individual blocks. We visualize the transformed images of SIA and BSR in Figure 5.4. However, we observe that both SIA and BSR employ fixed perspectives for the transformed images, which limits the diversity of transformed images and consequently restricts their attack transferability. As shown in Figures 5.1 and 5.2, our method enhances the diversity of transformed images by altering their perspectives,

thereby significantly improving the transferability.

5.3.3 Proposed Method

In this section, we first introduce perspective transformation and then present our perspective-invariant attack (PIA).

Perspective Transformation. This transformation, also known as the projective transformation, can be represented as a 3×3 matrix $[h_{ij}]_{3 \times 3}$ [120]:

$$\begin{bmatrix} \hat{u} \\ \hat{v} \\ \hat{w} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad (5.7)$$

$$\begin{aligned} u' &= \frac{\hat{u}}{\hat{w}} = \frac{h_{11}u + h_{12}v + h_{13}}{h_{31}u + h_{32}v + 1}, \\ v' &= \frac{\hat{v}}{\hat{w}} = \frac{h_{21}u + h_{22}v + h_{23}}{h_{31}u + h_{32}v + 1}, \end{aligned} \quad (5.8)$$

where (u, v) are the coordinates of the original image, (u', v') are the coordinates of the transformed image, and h_{ij} are the elements of the transformation matrix. To address the issues of holes and pixel overlapping in the transformed image, we can employ inverse mapping to establish a correspondence between the transformed and original images. Specifically, for each pixel coordinate (u', v') in the transformed image, we compute its corresponding position (u, v) in the original image by applying the inverse transformation matrix. Since the computed coordinates (u, v) may not be integer coordinates, we can utilize interpolation techniques (e.g., bilinear interpolation) to estimate the pixel value at (u, v) .

The commonly used transformations such as scaling, rotation, and translation can be represented as special cases of the projective transformation. For example, $\begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}$ defines scaling and rotation, while parameters h_{13} and h_{23} represent translation. Existing transformation-based attacks have proved that scaling [143], translation [23],

and rotation [129] are effective for enhancing the transferability of adversarial examples. However, the perspective transformation still remains under-explored in the context of adversarial attacks. Specifically, the parameters h_{31} and h_{32} control the perspective of the transformed image, which is not considered in existing input transformation-based attacks.

Perspective-Invariant Attack. Based on Equation 5.7, the degree of freedom (DOF) of the perspective transformation is eight. Thus, given four pairs of corresponding points in a 2D image, we can solve for the transformation matrix. Specifically, we randomly sample four positions in the input image and use them as the four corners of the transformed image. We formulate the four randomized corner positions as follows:

$$\begin{aligned} u'_{i,j} &= u_{i,j} \pm \text{int} \left(\xi_{i,j}^h \cdot (\delta_{max} \cdot \frac{h}{2}) \right), \\ v'_{i,j} &= v_{i,j} \pm \text{int} \left(\xi_{i,j}^w \cdot (\delta_{max} \cdot \frac{w}{2}) \right), \end{aligned} \quad (5.9)$$

$$\text{where } i \in \{0, h-1\}, j \in \{0, w-1\},$$

where $\xi_{i,j}^h$ and $\xi_{i,j}^w$ are randomly sampled from the uniform distribution $U(-1, 1)$; h and w are the height and width of the input image, respectively; δ_{max} is the maximum distortion budget; $\delta_{max} \cdot \frac{h}{2}$ and $\delta_{max} \cdot \frac{w}{2}$ are the maximum displacements of the corner positions; $(u_{i,j}, v_{i,j})$ represents a corner position of the input image; $u'_{i,j}$ and $v'_{i,j}$ are the coordinates of the transformed corner position; $\pm$ are used to force $(u'_{i,j}, v'_{i,j})$ to be integers within the range of $[0, h-1]$ and $[0, w-1]$, respectively. Note that for each occurrence of $\pm$, only one sign is selected based on the corner position. For instance, at the top-left corner $(0, 0)$, only ‘+’ is used, whereas at the bottom-right corner $(h-1, w-1)$, only ‘-’ is applied.

We can formulate the perspective transformation function $\Phi(\cdot)$ as follows:

$$\Phi(x; [h_{ij}]) = \Phi(x; \delta_{max}), \quad (5.10)$$

where $[h_{ij}]$ is the perspective transformation matrix, and δ_{max} is the distortion budget. Specifically, given an image x , we can randomly sample four corner positions based

on Equation 5.9. Then, we use these four corner positions to obtain eight equations based on Equation 5.8. By solving these eight equations, we can obtain the perspective transformation matrix $[h_{ij}]$.

We propose a perspective-invariant attack (PIA) based on the above perspective transformation function $\Phi(\cdot)$. We formulate our PIA based on MI-FGSM, and updates Equation 5.2 as follows:

$$g_{i+1} = \frac{1}{N} \sum_{j=1, \forall \phi_j \in \mathcal{P}}^N \nabla_x \mathcal{L}(F(\phi_j(x_i^{\text{adv}})), y), \quad (5.11)$$

where $\mathcal{P}$ is the set of transformation functions, including our perspective transformation $\Phi(\cdot)$ and other existing transformation methods.

PIA Algorithm. We summarize the PIA algorithm in Algorithm 4. In each iteration, we randomly transform the input using a set of transformation functions $\mathcal{P}$, including our perspective transformation $\Phi(\cdot)$, which has been overlooked in existing attack methods. We sum the gradients of the transformed images to obtain the final gradient g' . Since g' will be normalized when calculating the momentum g_{i+1} , we can omit the averaging operation. Finally, we update the intermediate adversarial example x_i^{adv} using the sign of the momentum, and clip it to be ℓ_∞ -bounded by ϵ .

5.4 Experiments

5.4.1 Experimental Setup

Datasets. We evaluate our method on two widely-used benchmark datasets. Following previous work [22, 133, 129], the primary dataset is the ImageNet-compatible dataset from the NIPS 2017 adversarial challenge. It consists of one thousand images from the ILSVRC 2012 dataset [101] specifically curated for adversarial attack evaluation. Additionally, we conduct experiments on CIFAR-10 [65], where we construct a

test set of 1,000 images by randomly sampling 100 images from each of the 10 classes. Unless stated otherwise, the ImageNet-compatible dataset is used for evaluation.

General Settings. We assess the transferability of adversarial examples under a non-targeted attack setting. The attacks are constrained by the ℓ_∞ norm, with a perturbation limit of $\epsilon = 16$. We set the momentum decay parameter μ to 1.0 and use a step size of $\eta = 2$. Each iterative attack is executed for 10 iterations ($T = 10$). All evaluated methods adopt the negative cross-entropy loss as the optimization objective for generating adversarial examples.

Models. For the ImageNet-compatible dataset, we evaluate our method on eight pre-trained DNN models, comprising four CNN-based models and four transformer-based architectures. The CNN-based models include RN-50 [44], DN-121 [49], MB-v2 [103], and Inc-v3 [118]. For transformer-based architectures, we employ Vision Transformer (ViT) [24], LeViT [41], ConViT [21], and Pooling-based ViT (PiT) [45]. We also use three secured DNNs: Inc-v3 ens3 (an ensemble of three Inception-v3 networks), Inc-v3 ens4 (an ensemble of four Inception-v3 networks), and IncRes-v2 ens3 (an ensemble of three IncRes-v2 networks). Adversarially trained models like these have been frequently used in earlier research [22, 23, 132]. For the CIFAR-10 dataset, we primarily evaluate our method on Inc-v3, MB-v2, and RN-20. Given that CIFAR-10’s small image dimensions are not well-suited for vision transformers, we only use CNN-based architectures. Additionally, we incorporate two adversarially trained models for CIFAR-10: RN-20-ens-gal [62] and RN-20-ens-dv [147].

Attacks. We comprehensively evaluate our method against existing transfer-based attack techniques: DI [143], TI [23], SI [80], Admix [132], SSA [84], SIA [133], and BSR [129]. Among these, SIA and BSR represent the current state-of-the-art in transfer-based attacks. We incorporate MI [22] as a fundamental baseline component across all attack methods. For clarity in presentation, we omit the ‘MI’ prefix when referring to these methods. All comparative experiments strictly follow the original parameter settings proposed in their respective papers. Specifically: For DI, we set

the transformation probability to 0.7. For TI, we employ a 5×5 convolution kernel. The MI decay factor μ is set to 1.0. SI utilizes 5 scaled copies of the input image. In the case of Admix, we configure its weight as 0.2 then generate 3 mixed images (corresponding to $m_2 = 3$ in [132]). For SSA, we set the tuning factor $\rho = 0.5$ and std σ of noise as $16/255$. For SIA, we use 3 shuffled blocks and 10 random transformations. For BSR, we use 2 shuffled blocks and the maximum rotation angle of 24 degrees. We use $N = 20$ transformed copies for SSA, SIA, and BSR. We also evaluate the attack performance in the ensemble framework [81] and the L2T framework [158].

Implementation Details of PIA. We set $N = 20$ and $\delta_{max} = 0.6$ for our perspective transformation, which is implemented using PyTorch. Our transformation pool $\mathcal{P}$ includes three types of transformations: block-wise shuffle, block-wise rotation, and perspective transformation. For block-wise shuffle, we use 3 shuffled blocks, following the settings of SIA. For block-wise rotation, we use 2 shuffled blocks and a maximum rotation angle of 24 degrees, following the settings of BSR.

5.4.2 Adversarial Transferability of PIA

In this section, we evaluate the transferability of our PIA in three different settings: attacking a single surrogate model, combining with DI-TI-FGSM, and attacking an ensemble of models. These are three standard settings for evaluating the transferability of adversarial examples.

Attacking a Single Surrogate Model. Table 5.1 shows the attack success rates of our PIA against eight target models on the ImageNet-compatible dataset. We use different surrogate models, including RN-50, DN-121, Inc-v3, and MB-v2. We compare PIA with DI, TI, SI, Admix, SSA, SIA, and BSR. The framework of all methods is MI-FGSM. Our PIA achieves the highest attack success rate among all methods, indicating that our method is effective for attacking a single surrogate model. For example, when using RN-50 as the surrogate model, our PIA achieves an average

Source	Attack	RN-50	DN-121	Inc-v3	MB-v2	ViT	LeViT	ConViT	PiT	Average
RN-50	DI	100	97.0	76.1	95.3	29.7	32.4	19.8	29.8	60.0
	TI	100	90.5	59.8	87.2	31.6	29.0	18.9	25.7	55.3
	SI	100	95.9	71.3	90.7	27.3	24.5	16.2	23.4	56.2
	Admix	100	97.7	80.2	95.0	29.6	32.4	18.4	27.7	60.1
	SSA	100	98.2	88.3	98.4	42.3	54.1	29.8	45.3	69.5
	SIA	100	99.9	96.2	99.7	45.6	68.9	39.0	59.0	76.0
	BSR	100	99.9	95.5	99.6	48.8	71.6	40.6	60.7	77.1
	PIA	100	99.9	98.3	99.8	61.4	84.2	57.3	76.6	84.7
DN-121	DI	96.7	100	77.5	93.2	32.2	38.5	23.2	34.3	61.9
	TI	90.5	100	62.9	86.4	33.9	35.1	22.2	33.3	58.0
	SI	94.1	100	75.6	89.1	33.7	36.7	26.2	35.3	61.3
	Admix	96.8	100	84.5	94.5	38.9	47.4	28.8	41.9	66.6
	SSA	98.6	100	92.0	96.7	47.0	64.4	39.7	58.9	74.7
	SIA	99.7	100	96.6	99.2	50.5	73.1	42.9	65.5	78.4
	BSR	99.5	100	95.0	98.9	49.6	71.8	41.7	63.7	77.5
	PIA	99.7	100	98.1	99.3	61.5	82.9	56.1	76.8	84.3
Inc-v3	DI	74.0	74.7	99.6	79.5	29.5	30.8	20.8	29.3	54.8
	TI	59.3	61.7	100	67.8	30.1	25.7	19.1	26.8	48.8
	SI	69.5	71.7	100	76.3	31.9	31.0	23.3	33.4	54.6
	Admix	78.5	81.9	100	81.7	36.6	40.7	28.4	39.5	60.9
	SSA	82.1	85.9	99.7	85.1	40.6	51.3	33.2	49.7	66.0
	SIA	94.8	96.7	100	96.4	47.6	67.7	42.7	62.4	76.0
	BSR	91.8	95.8	100	94.0	44.8	58.8	38.8	57.4	72.7
	PIA	95.4	97.5	100	96.8	53.6	73.7	50.0	68.3	79.4
MB-v2	DI	89.8	88.3	70.0	100	28.2	34.0	17.9	30.1	57.3
	TI	78.4	75.8	53.1	100	28.5	29.0	18.8	24.5	51.0
	SI	83.2	84.6	64.0	100	27.2	28.8	18.1	26.6	54.1
	Admix	88.6	90.3	71.8	100	29.8	33.6	18.4	29.6	57.8
	SSA	91.4	89.3	72.0	100	33.3	41.3	21.8	34.3	60.4
	SIA	99.2	98.8	90.2	100	41.1	65.7	32.6	53.0	72.6
	BSR	99.1	99.3	95.2	100	47.2	73.6	38.7	62.5	77.0
	PIA	99.3	99.4	96.6	100	55.6	81.9	48.1	71.0	81.5

Table 5.1: Attack performance on eight target models. All methods are based on MI. The best results are shown in bold.

attack success rate of 84.7% against all eight target models, which is 7.6% higher than the next best method, BSR. When using DN-121 as the surrogate model, our

PIA achieves an average attack success rate of 84.3% against all eight target models, which is 5.9% higher than the next best method, SIA.

Combining with DI-TI-FGSM. Table 5.2 presents the attack performance of combining DI-TI-FGSM with various existing attack strategies, evaluated on eight target models using the ImageNet-compatible dataset. Our proposed method, DI-TI-PIA, consistently achieves the highest success rate among all compared methods, demonstrating its generality with other attack techniques for boosting overall effectiveness. For instance, when using RN-50 as the surrogate model, DI-TI-PIA obtains an average attack success rate of 87.8% across all eight target models, outperforming the next best method, DI-TI-SIA, by 5.8%. Similarly, with DN-121 as the surrogate, DI-TI-PIA achieves an average success rate of 86.5%, exceeding DI-TI-SIA by 3.6%.

Attacking Ensemble-based Models. Table 5.3 reports the performance of our PIA within an ensemble-based attack framework [81]. The ensemble surrogate consists of four models: RN-50, DN-121, Inc-v3, and MB-v2. Experimental results indicate that PIA achieves the best overall transferability among all evaluated methods. For example, PIA obtains an average success rate of 93.9% across the eight target models, surpassing the second-best approach, SIA, by 3.7%.

5.4.3 Attacking Defense Methods

Attacking Adversarially Trained Models. Table 5.4 shows the attack success rates of our PIA against three ensemble-based adversarially trained defenses on the ImageNet-Compatible dataset. These three secured models are widely used to evaluate the transferability of adversarial examples, including Inc-v3_{ens3}, Inc-v3_{ens4}, and IR-v2_{ens}. The results show that PIA significantly improves the attack success rates of existing attack methods. PIA achieves average attack success rates of 55.1%, 56.5%, 80.7% when using RN-50, DN-121, and the ensemble of RN-50 and DN-121 as the surrogate models, respectively. These results improve upon the attack

Source	Attack	RN-50	DN-121	Inc-v3	MB-v2	ViT	LeViT	ConViT	PiT	Average
RN-50	DI-TI	100	98.2	82.0	96.8	38.7	46.1	27.2	38.4	65.9
	DI-TI-SI	100	99.4	93.7	97.9	48.0	62.4	37.3	53.0	74.0
	DI-TI-Admix	100	99.7	96.3	98.6	54.3	72.1	41.6	61.0	78.0
	DI-TI-SSA	100	99.1	91.7	98.3	55.2	62.3	40.5	57.4	75.6
	DI-TI-SIA	100	99.9	97.0	99.7	59.2	81.2	49.1	69.7	82.0
	DI-TI-BSR	100	99.9	96.3	99.8	58.8	78.3	48.2	68.2	81.2
	DI-TI-PIA	99.9	99.9	98.6	99.8	71.2	88.1	65.2	80.0	87.8
DN-121	DI-TI	96.4	100	81.3	94.4	39.5	47.7	28.2	43.2	66.3
	DI-TI-SI	98.1	100	90.5	96.8	50.9	64.8	40.4	59.8	75.2
	DI-TI-Admix	99.1	100	94.6	98.3	59.2	71.1	46.9	66.1	79.4
	DI-TI-SSA	98.7	100	93.7	97.2	57.0	73.4	48.7	69.7	79.8
	DI-TI-SIA	99.6	100	97.2	99.1	60.3	81.3	51.7	73.6	82.9
	DI-TI-BSR	99.5	100	95.2	98.5	56.5	75.0	46.0	68.4	79.9
	DI-TI-PIA	99.7	100	97.9	99.2	68.8	85.1	61.1	79.8	86.5
Inc-v3	DI-TI	73.8	78.1	99.5	79.3	34.7	38.8	25.8	35.8	58.2
	DI-TI-SI	83.3	88.5	100	86.1	42.2	50.6	36.7	48.7	67.0
	DI-TI-Admix	88.5	92.8	100	90.5	50.3	60.7	42.7	60.8	73.3
	DI-TI-SSA	81.7	87.5	99.8	85.7	47.3	58.5	41.3	55.6	69.7
	DI-TI-SIA	94.1	96.7	100	95.2	54.2	75.3	51.6	69.6	79.6
	DI-TI-BSR	90.4	95.3	100	93.4	47.6	62.9	41.1	61.0	74.0
	DI-TI-PIA	94.3	96.8	100	96.1	59.3	75.4	53.0	72.4	80.9
MB-v2	DI-TI	92.9	93.3	76.9	100	36.5	48.8	24.6	38.7	64.0
	DI-TI-SI	96.7	98.1	91.7	100	48.5	67.4	35.3	57.1	74.3
	DI-TI-Admix	98.6	99.0	93.8	100	53.7	73.8	38.5	61.4	77.3
	DI-TI-SSA	95.2	94.9	83.4	100	46.8	58.2	32.5	52.0	70.4
	DI-TI-SIA	99.2	99.2	94.2	100	53.6	78.2	43.6	64.1	79.0
	DI-TI-BSR	99.0	99.2	95.3	100	56.8	80.4	44.8	67.9	80.4
	DI-TI-PIA	99.7	99.6	97.3	100	66.2	87.9	57.7	79.7	86.0

Table 5.2: Attack performance of combining DI-TI with existing methods against eight target models. All methods are based on MI. The best results are shown in bold.

success rates of the latest attack method, BSR, by 13.9%, 14.8%, 15.4% respectively. This demonstrates the superiority of our PIA in attacking ensemble-based adversarially trained models.

Table 5.5 presents the success rates of attacks using our PIA on the CIFAR-10 dataset.

Attack	RN-50*	DN-121*	Inc-v3*	MB-v2*	ViT	LeViT	ConViT	PiT	Average
DI	100*	100*	99.6*	100*	41.5	71.6	40.6	62.1	76.9
TI	100*	100*	99.7*	100*	42.0	59.6	38.1	53.4	74.1
SI	100*	100*	100*	100*	46.6	65.4	41.0	60.1	76.6
Admix	100*	100*	99.9*	100*	49.5	71.0	43.7	63.3	78.4
SSA	100*	100*	99.9*	100*	65.2	89.0	63.7	82.9	87.6
SIA	100*	100*	100*	100*	68.7	94.3	69.6	89.1	90.2
BSR	100*	100*	99.9*	100*	68.3	93.6	67.8	85.6	89.4
PIA	100*	100*	100*	100*	79.4	97.2	80.6	94.2	93.9

Table 5.3: Attack performance using an ensemble of models as the surrogate model against eight target models. All methods are based on MI. * indicates the surrogate models. The best results are shown in bold.

We employ three surrogate configurations: Inc-v3, MB-v2, and a combination of Inc-v3 and MB-v2. The adversarial examples generated are tested on four models: Inc-v3, MB-v2, RN-20-ens-gal, and RN-20-ens-dv, with RN-20-ens-gal and RN-20-ens-dv being adversarially trained models on CIFAR-10. The findings indicate that PIA significantly enhances the attack success rates of existing methods. For instance, with Inc-v3 as the surrogate model, our PIA achieves an average success rate of 86.8%, surpassing the next best method, SIA, by 3.9%. Notably, our PIA demonstrates strong performance against adversarially trained models RN-20-ens-gal and RN-20-ens-dv, highlighting its effectiveness in circumventing adversarial defense mechanisms. Specifically, with Inc-v3 as the surrogate model, PIA achieves attack success rates of 73.5% and 74.4% against RN-20-ens-gal and RN-20-ens-dv, respectively, surpassing the next best method, SIA, by significant margins of 7.9% and 8.0%.

Attacking Advanced Defense Methods. Table 5.6 shows the attack success rates of our PIA against various defense methods on the ImageNet-compatible dataset. The first line shows the clean accuracy of the defense methods. We mainly test our PIA on five defense methods: AT [86], RS [19], HGD [79], NRP [90], and DiffPure [93]. For HGD, AT, and RS, we utilize their official implementations. Since HGD, NRP,

Source	Attack	Inc-v3 _{ens3}	Inc-v3 _{ens4}	IR-v2 _{ens}	Avg
RN-50	DI	27.3	24.3	12.0	21.2
	TI	25.6	25.4	14.3	21.8
	SI	29.9	26.6	12.3	22.9
	Admix	31.0	28.9	14.2	24.7
	SSA	44.1	39.9	23.9	36.0
	SIA	46.0	42.3	23.2	37.2
	BSR	50.4	45.3	27.9	41.2
	PIA	65.5	60.4	39.3	55.1
DN-121	DI	27.8	26.1	13.6	22.5
	TI	29.1	26.4	14.9	23.5
	SI	36.3	33.1	16.6	28.7
	Admix	40.7	38.0	18.8	32.5
	SSA	56.5	47.8	30.3	44.9
	SIA	47.3	43.6	24.2	38.4
	BSR	50.7	46.5	28.0	41.7
	PIA	66.0	62.2	41.4	56.5
Ens	DI	44.5	39.7	21.2	35.1
	TI	46.0	41.1	25.9	37.7
	SI	55.7	52.6	29.1	45.8
	Admix	61.6	55.0	32.3	49.6
	SSA	78.3	72.6	51.8	67.6
	SIA	77.2	69.0	47.0	64.4
	BSR	77.0	70.0	48.8	65.3
	PIA	90.7	85.6	65.9	80.7

Table 5.4: Attack success rates (%) against three ensemble-based adversarially trained defenses. All methods are based on MI. The best results are shown in bold.

and DiffPure are purification-based defense methods, we can use them to protect any given DNNs. We choose ViT and ConViT as the protected models since they are the most difficult models to attack in Table 5.1. The results show that our PIA achieves the best attack success rates among all attack methods. Specifically, with RN-50 as the surrogate model, our PIA achieves an average attack success rate of 41.6% against all defense methods, demonstrating a substantial improvement of 6.3% over the next best performing method BSR (35.3%). The superiority of PIA is also significant when

Source	Attack	Inc-v3	MB-v2	RN-20- ens-gal	RN-20- ens-dv	Avg
Inc-v3	DI	88.8	78.5	34.4	33.5	58.8
	TI	86.0	77.3	37.6	35.7	59.2
	SI	99.9	88.4	41.6	39.9	67.5
	Admix	100	91.0	42.7	42.3	69.0
	SSA	98.4	98.2	48.0	47.4	73.0
	SIA	100	99.6	65.6	66.4	82.9
	BSR	100	96.0	63.2	62.6	80.5
	PIA	100	99.5	73.5	74.4	86.8
MB-v2	DI	99.4	100	46.7	50.1	74.1
	TI	97.9	100	49.4	49.7	74.2
	SI	96.0	100	36.8	40.1	68.2
	Admix	97.6	99.9	41.2	45.6	71.1
	SSA	99.8	100	47.2	51.0	74.5
	SIA	100	100	60.2	63.4	80.9
	BSR	99.5	100	58.6	59.5	79.4
	PIA	100	100	68.3	70.8	84.8
Ens	DI	96.8	99.2	48.1	52.5	74.2
	TI	88.3	95.6	50.1	50.2	71.0
	SI	100	99.9	46.5	49.1	73.9
	Admix	100	99.9	50.3	54.3	76.1
	SSA	99.7	99.9	57.9	63.9	80.3
	SIA	100	100	73.6	74.8	87.1
	BSR	100	100	69.3	74.3	85.9
	PIA	100	100	82.2	83.1	91.3

Table 5.5: Attack success rates (%) against four target models on the CIFAR-10 dataset. All methods are based on MI. The best results are shown in bold.

using DN-121 or the ensemble of RN-50 and DN-121 as the surrogate models.

5.4.4 Combining with L2T Framework

L2T [158] leverages reinforcement learning to optimize the selection of input transformations from a candidate pool to improve adversarial transferability, but its candidate pool does not include perspective transformation. We follow the official implementation of L2T and adopt the number of transformations as 2. It is worth noting that the number of copies for each image can reach 400, depending on the specific transforma-

Source	Attack	HGD	AT	RS	Protected Model: ViT			Protected Model: ConViT			Average
					HGD	NRP	DiffPure	HGD	NRP	DiffPure	
–	–	0.7	36.7	19.9	11.9	18.7	15.3	5.1	10.3	7.6	14.0
RN-50	DI	22.8	38.7	28.0	18.9	28.8	20.0	12.2	17.3	14.9	22.4
	TI	25.0	38.7	30.8	23.4	30.1	21.1	14.1	17.9	16.4	24.2
	SI	21.1	38.3	27.8	19.1	29.1	18.1	11.5	17.3	12.9	21.7
	Admix	26.1	38.4	28.8	21.1	29.6	17.7	12.8	18.2	14.9	23.1
	SSA	33.9	41.6	37.7	23.8	37.4	24.1	15.3	25.6	19.8	28.8
	SIA	63.5	41.0	37.2	29.6	36.9	24.5	25.1	25.6	19.9	33.7
	BSR	63.7	40.9	36.0	34.0	37.3	26.0	29.1	27.9	22.9	35.3
	PIA	76.3	42.4	41.4	38.4	43.3	31.2	37.4	34.6	29.1	41.6
DN-121	DI	20.5	38.5	28.0	21.1	31.5	20.4	13.2	19.4	14.7	23.0
	TI	24.0	38.6	29.8	25.6	30.4	22.7	16.4	18.7	16.5	24.7
	SI	21.5	39.4	30.1	21.9	31.3	19.2	14.7	20.6	15.2	23.8
	Admix	29.4	39.7	31.7	24.4	33.0	21.3	17.8	19.9	16.2	25.9
	SSA	43.4	41.2	37.8	28.3	37.9	27.1	21.2	26.7	21.7	31.7
	SIA	55.9	40.7	36.0	31.5	37.1	26.3	26.2	26.1	22.5	33.6
	BSR	58.9	40.5	35.9	35.3	36.7	26.7	31.9	25.5	22.8	34.9
	PIA	71.0	43.1	42.1	42.7	40.8	32.6	39.3	32.9	29.0	41.5
Ensemble	DI	38.5	40.4	32.9	23.7	34.6	22.9	19.6	24.8	19.3	28.5
	TI	52.9	40.7	35.8	30.8	35.0	26.1	27.1	26.3	23.1	33.1
	SI	48.7	41.6	35.9	26.4	35.7	23.2	21.4	26.6	20.9	31.2
	Admix	50.2	42.1	37.4	28.0	37.8	25.9	22.5	26.4	21.0	32.4
	SSA	67.1	45.3	48.9	35.7	47.1	38.0	34.1	43.5	33.2	43.7
	SIA	82.1	45.1	44.5	41.4	44.6	33.3	43.3	39.2	32.0	45.1
	BSR	85.7	44.0	44.0	49.9	44.6	34.6	51.0	38.5	31.5	47.1
	PIA	92.0	46.8	52.2	57.0	50.3	42.7	60.7	48.6	41.5	54.6

Table 5.6: Success rates (%) against various defense methods on ImageNet. All methods are based on MI. The best results are shown in bold.

tions randomly sampled in each iteration, making L2T computationally expensive. Figure 5.5 shows that adding our perspective transformation to the candidate pool of L2T can further improve its attack transferability. Specifically, we use RN-50 and DN-121 as surrogate models and evaluate the success rates on eight target models. We use the ImageNet-compatible dataset for evaluation. The results show that L2T performs poorly on ViT-based models, while our PIA can significantly improve L2T’s

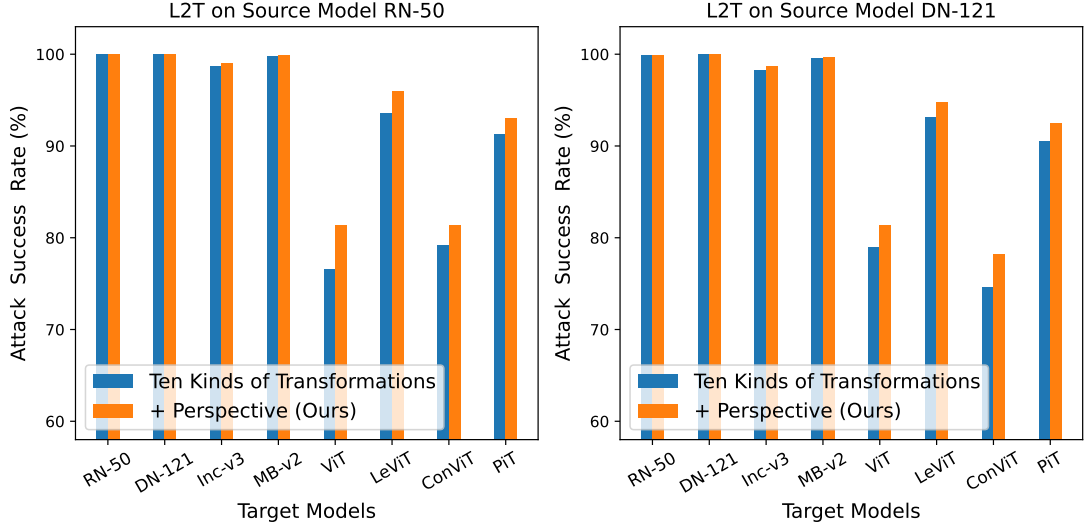


Figure 5.5: Combining with L2T framework. Our perspective transformation can be added into the candidate pool of L2T to further improve its attack success rates.

success rates. This shows that our method can enhance the latest learning-based input transformation framework, highlighting the generality of PIA.

5.4.5 Results on Multimodal LLMs

Table 5.7 shows the attack success rates of PIA against multimodal LLMs. A set of 100 images is randomly chosen from the ImageNet-compatible dataset to serve as target images. ResNet-50 is employed as the surrogate model to assess the attack success rates on three multimodal LLMs: Qwen2-VL, Claude-3.5, and GPT-4o. These three LLMs are popular commercial LLMs that serve as black-box models. For each generated adversarial example, we use the prompt "Is this image a photo of true label? Yes or No?" to obtain predictions from the LLMs. The results show that our PIA can effectively improve the attack success rates of existing attack methods. Specifically, PIA achieves an average attack success rate of 67% against all three LLMs, which is 4% higher than the next best method, SIA. Notably, Qwen2-VL is the most vulnerable to adversarial examples, with an attack success rate of 90% when

Attack	Qwen2-VL	Claude-3.5	GPT-4o	Avg
DI-TI	76	31	30	46%
DI-TI-SI	81	46	40	56%
DI-TI-Admix	85	50	44	60%
DI-TI-SSA	77	48	42	56%
DI-TI-SIA	86	52	51	63%
DI-TI-BSR	81	49	47	59%
DI-TI-PIA	90	56	54	67%

Table 5.7: Attack success rates (%) against multimodal LLMs. We randomly sample 100 images from the ImageNet-compatible dataset as the victim images, and use ResNet-50 as the surrogate model.

facing PIA. GPT-4o is the most robust among the three LLMs, with an attack success rate of 54% when facing our PIA.

5.4.6 Computational Complexity

Referring to the illustration of transformation-based attacks in Figure 5.3, the computational complexity of PIA and its competitors is mainly determined by the number of attack iterations T and the number of image copies in each iteration N . Additionally, if a transformation is time-consuming, such as the spectrum transformation in SSA, the computational complexity will be higher. Table 5.8 shows the average time for each attack to generate an adversarial example on the ImageNet-compatible dataset. We use an RTX 3090 GPU for the experiments. SIA and BSR require 0.823 and 0.729 seconds per image, respectively. Our PIA achieves comparable efficiency at 0.766 seconds per image.

Attack	Copy Number	Iteration Number	Average Time (s)
DI	1	10	0.041
TI	1	10	0.040
SI	5	10	0.184
Admix	15	10	0.537
SSA	20	10	0.986
SIA	20	10	0.823
BSR	20	10	0.729
PIA	20	10	0.766

Table 5.8: Average time (s) for each attack to generate an adversarial example on the ImageNet-compatible dataset. The surrogate model is RN-50.

5.4.7 Ablation Study

(1) Ablation Study on the Distortion Parameter of PIA. Figure 5.6 presents an ablation study examining the impact of the distortion parameter δ_{max} in PIA. We experiment with varying distortion values to generate adversarial examples on the ImageNet-compatible dataset. Three surrogate models are employed in our study: ResNet-50 (RN-50), DenseNet-121 (DN-121), and Inception-v3 (Inc-v3). The generated adversarial examples are evaluated across eight target models. When the distortion parameter is set to 0, our PIA degenerates into a standard MI-FGSM attack. Our results demonstrate that the attack success rate achieves optimal and stable performance when the distortion parameter δ_{max} is approximately 0.6.

(2) Parameter Study on the Number of Image Copies. Figure 5.7 shows the impact of the number of image copies on the attack success rate of our PIA. We use two surrogate models: RN-50 and DN-121. We test the attack success rates on eight target models. Our results demonstrate that the attack success rate benefits from increasing the number of image copies. While increasing the number of image copies

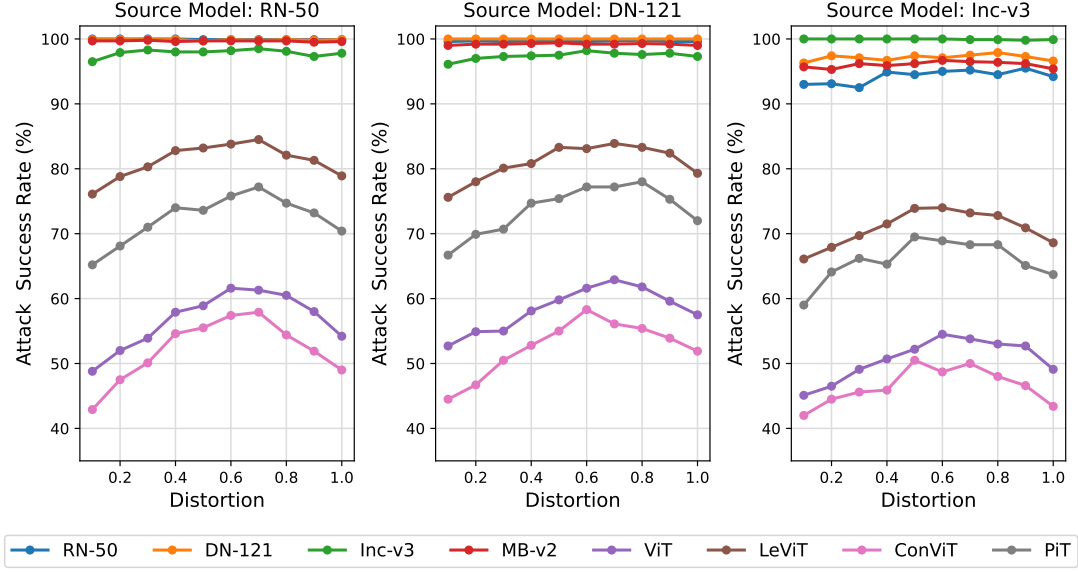


Figure 5.6: Impact of the distortion parameter on the attack success rate of our PIA.

beyond 20 would further improve the attack success rate, we set our PIA to use 20 image copies to maintain a fair comparison with other attacks.

(3) Parameter Study on the Number of Attack Iterations. Figure 5.8 shows the impact of the number of attack iterations on the attack success rate of our PIA. We use the default setting of 20 image copies in each attack iteration. The surrogate models are RN-50 and DN-121. We test the attack success rates on eight target models. The results demonstrate that the attack success rate exhibits rapid growth during the initial attack iterations, before stabilizing as the number of iterations approaches 10.

(4) Ablation Study of Transformation Function Selection. Our PIA uses three transformation functions: perspective transformation, block-wise shuffle, and block-wise rotation. The last two transformations are adopted from SIA and BSR, respectively. While SIA and BSR represent state-of-the-art attack methods, they fail to recognize the significance of perspective transformation in enhancing attack effectiveness. Figure 5.9 shows the success rates of different combinations of these

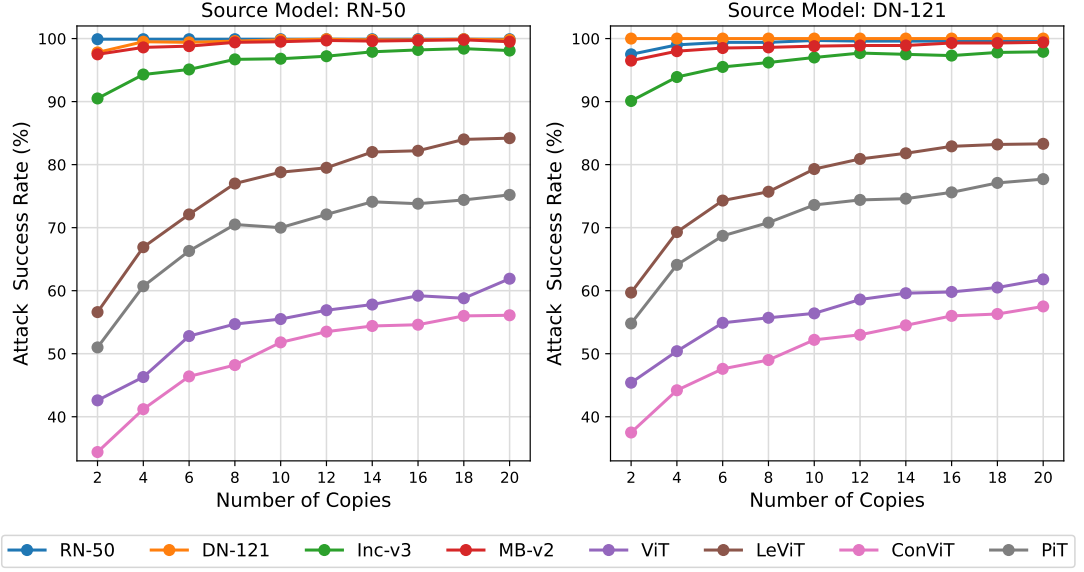


Figure 5.7: Impact of the number of image copies on the attack success rate of our PIA.

transformations. We use RN-50 as the surrogate model, and the number of image copies is fixed to 20. The number of attack iterations is set to 10. The combination of Perspective-SIA-BSR demonstrates remarkable attack success rates, particularly when targeting ViT-based models. The experimental results clearly indicate that incorporating our proposed perspective transformation significantly enhances the overall attack success rate.

Figure 5.10 shows the impact of varying the number of image copies on attack success rates across different transformation function combinations. Since Figure 5.7 shows that 20 image copies may not be enough for PIA to achieve the best attack success rate, we set the number of image copies from 5 to 50 in Figure 5.10. We denote our PIA method without the perspective transformation component as SIA-BSR. We fix the number of attack iterations to 10. Figure 5.10 shows that the attack success rate of SIA-BSR is significantly lower than that of our PIA. Notably, the attack success rate of our PIA with only 20 image copies is already much higher than that of SIA-BSR with 50 image copies. This demonstrates that employing our perspective

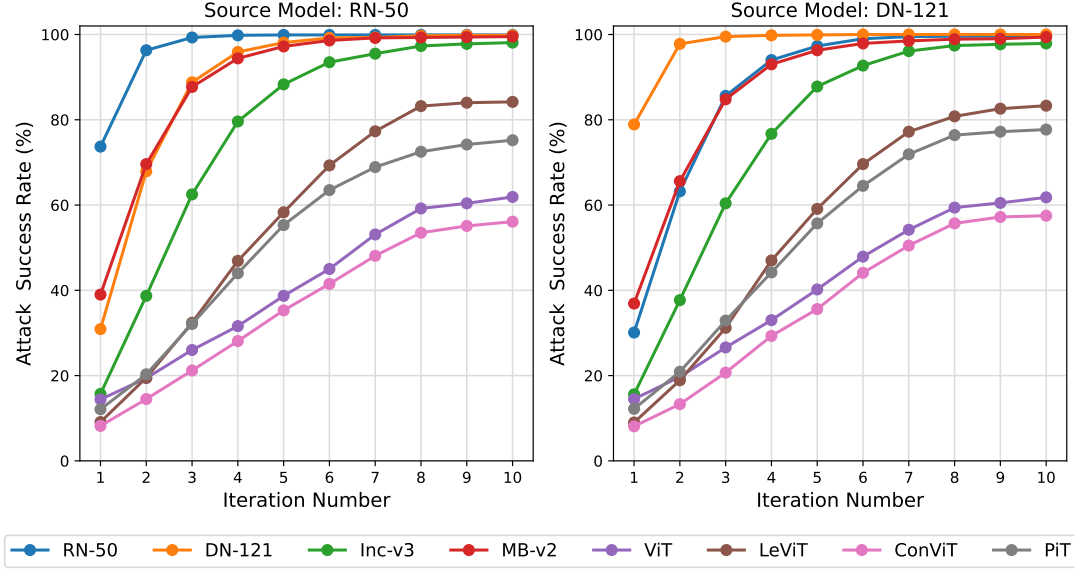


Figure 5.8: Impact of the number of attack iterations on the attack success rate of our PIA.

transformation technique in PIA yields superior performance compared to the latest SIA and BSR methods.

(5) Clean Accuracy after Input Transformations. Figure 5.11 shows the prediction accuracies of benign images after block-wise shuffle, block-wise rotation, and perspective transformation. Results show that extreme parameter settings significantly reduce model accuracy, particularly for CNN-based models. Our optimal parameter settings maintain moderate clean accuracy levels. For perspective transformation, we set the distortion parameter δ_{max} to 0.6, achieving clean accuracies above 50%, as shown in the right plot of Figure 5.11. Similarly, using three shuffle blocks (following SIA) and a 24-degree rotation angle (following BSR) yields clean accuracies above 50% in the left and middle plots, respectively. These moderate clean accuracy levels are ideal because too low accuracy would compromise the surrogate models' predictive ability, whereas too high accuracy may indicate potential overfitting.

(6) Visualization of Adversarial Examples. Figure 5.12 shows the visualization

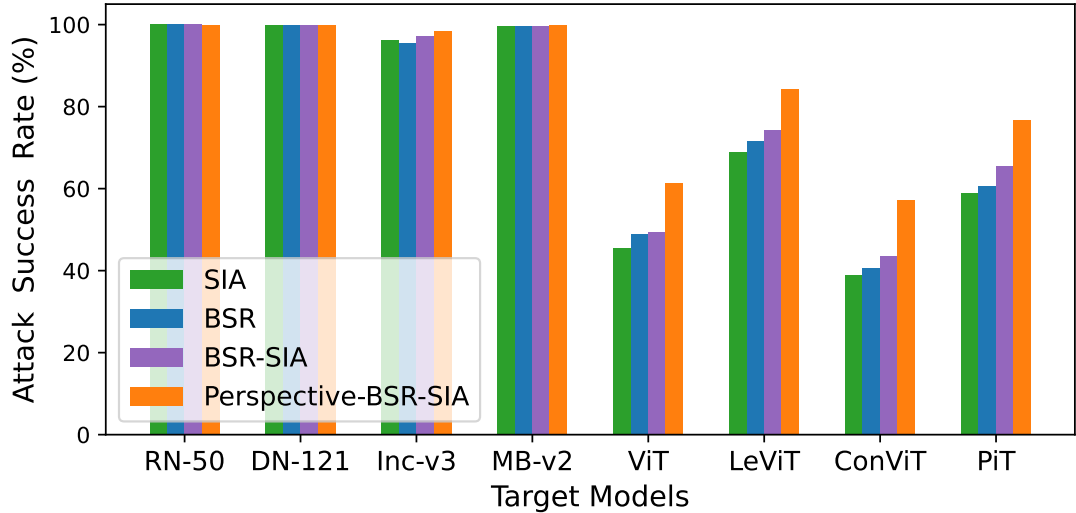


Figure 5.9: Impact of different combinations of transformation functions on the attack success rate of our PIA.

of adversarial examples generated by the baseline attack and our PIA. Each row displays the original image, the adversarial examples, and the difference between the original and adversarial examples (referred to as perturbations). The perturbation patterns generated by our PIA exhibit more semantic awareness, making them more effective at deceiving the target model.

5.4.8 Discussion of Non-targeted Attacks under Defenses

In this section, we explore whether incorporating our proposed transfer-based attacks into the training loop can improve the adversarial robustness of current DNN models against black-box transfer attacks. We utilize Adversarial Training (AT), the most popular defense mechanism to date, as our baseline framework.

Standard adversarial training formulates the defense as a min-max optimization problem using Projected Gradient Descent (PGD) attacks. The objective is to minimize the loss on the worst-case examples generated within a perturbation budget ϵ . For-

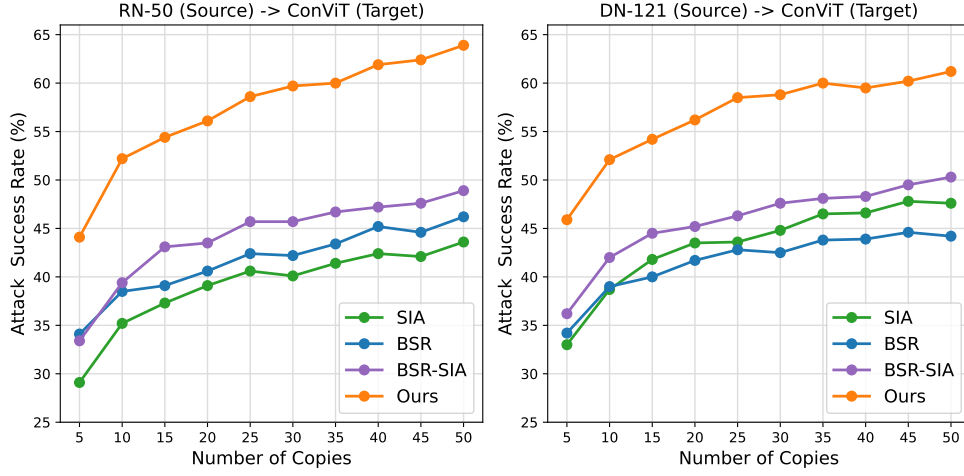


Figure 5.10: Impact of varying the number of image copies on attack success rates across different transformation function combinations.

mally, the standard AT objective is:

$$\min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\max_{\delta \in \mathbb{S}} \mathcal{L}(f_{\theta}(x + \delta), y) \right]$$

where $\mathcal{D}$ represents the data distribution, $\mathbb{S} = \{\delta : \|\delta\|_p \leq \epsilon\}$ is the feasible perturbation set, and $\mathcal{L}$ is the loss function.

To enhance the model’s resistance specifically against transfer-based black-box attacks, we propose augmenting the inner maximization step by integrating our proposed attack methods. Instead of relying solely on PGD, which may overfit to specific gradient patterns, we introduce a set of attack strategies $\mathcal{A}$ containing our proposed methods, including StyLess, SODA, and PIA. The training process thus involves sampling an attack generation function $g \sim \mathcal{A}$ to craft adversarial examples. The modified optimization objective becomes:

$$\min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \mathbb{E}_{g \sim \mathcal{A}} [\mathcal{L}(f_{\theta}(\hat{x}_{adv}), y)], \quad \text{where } \hat{x}_{adv} = g(x, y; \theta).$$

By exposing the model to the diverse and aggressive perturbation patterns generated by our attack methods during training, the model is encouraged to learn more robust feature representations. To verify the effectiveness of this defense method, we

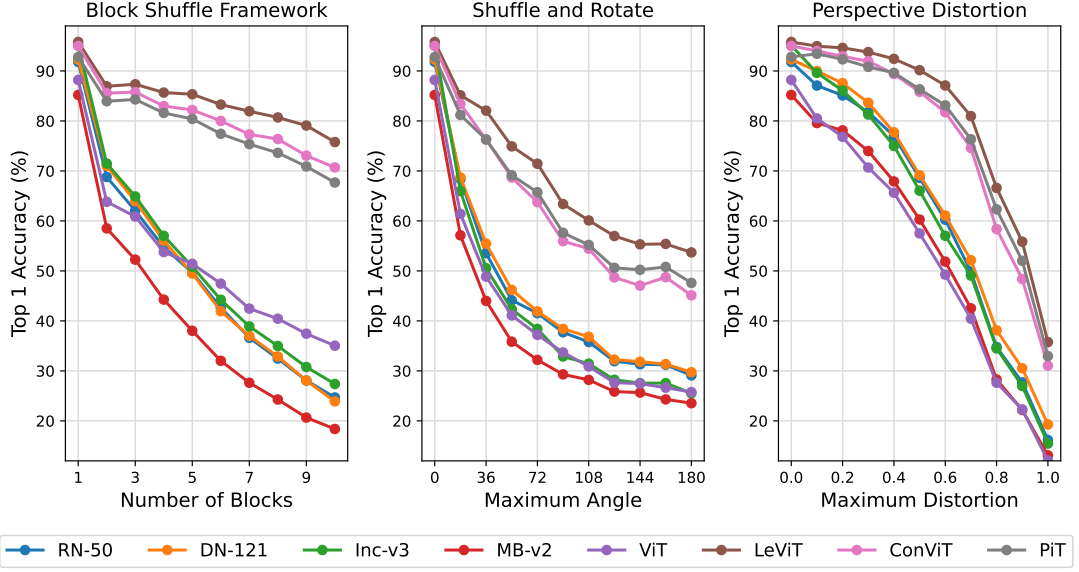


Figure 5.11: Clean accuracy of the input images after block-wise shuffle, block-wise rotation, and perspective transformation.

conducted experiments using ResNet-50 (RN-50) and Inception-v3 (Inc-v3) as the target defense models. We evaluate the robustness of these models against various transfer-based attacks, including DI, TI, SI, StyLess, SODA, and PIA. The adversarial examples are generated on a hold-out surrogate model, DenseNet-121 (DN-121), to ensure a strict black-box setting. The results are reported in Table 5.9.

Analysis of Results. As shown in Table 5.9, normally trained models are highly vulnerable to advanced transfer attacks. For instance, the robust accuracy of RN-50 drops to as low as 0.3% under effective attacks like PIA. While standard AT significantly mitigates this vulnerability, raising the average robustness of RN-50 to 59.6%, it still leaves room for improvement against highly transferable perturbations.

Incorporating our proposed methods into the adversarial training loop consistently outperforms the standard AT baseline. For RN-50, our method improves the average robustness from 59.6% to 60.8%, achieving the best performance across all evaluated attack metrics. A more pronounced improvement is observed on the Inc-v3 architec-

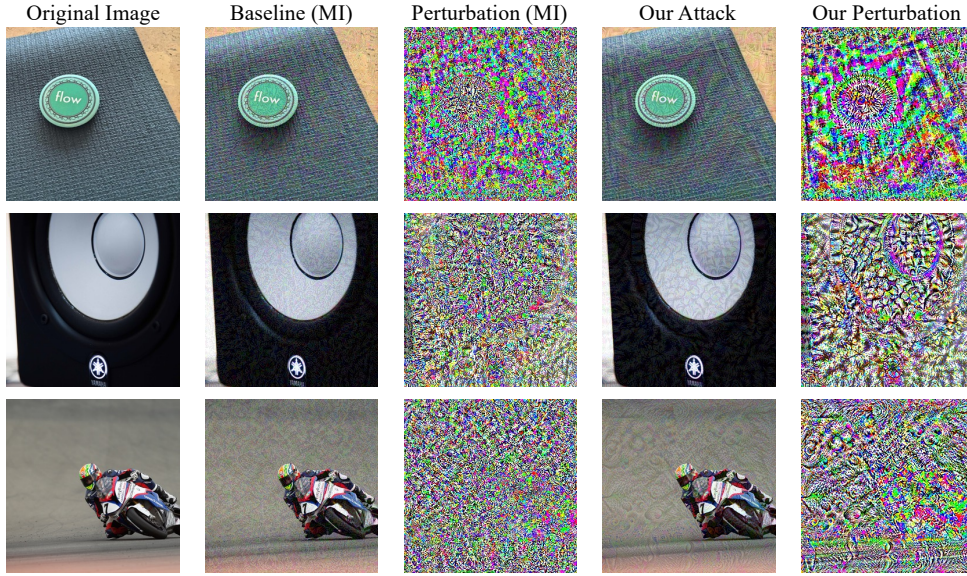


Figure 5.12: Visualization of adversarial examples generated by the baseline attack and our PIA.

ture, where our method boosts the average robustness from 53.6% to 56.8%. These results indicate that our proposed attacks are not only effective tools for evaluating model vulnerability but also serve as powerful regularizers for defense. Training with the strong transfer gradients induced by our methods improves black-box robustness to diverse adversarial perturbations.

5.4.9 Comparison between SODA and PIA

In this section, we conduct a comprehensive comparison between the proposed PIA and our previously introduced SODA method. We investigate their individual performances, analyze their distinct underlying mechanisms, explore the potential for combining these two strategies, and perform a gradient alignment analysis.

Performance Comparison. As presented in Table 5.10, PIA demonstrates superior transferability compared to SODA in the majority of experimental settings. Specifically, when RN-50 and DN-121 are utilized as surrogate models, PIA achieves average

Defense	DI	TI	SI	StyLess	SODA	PIA	Average
RN-50	3.4	9.5	5.9	0.3	0.4	0.3	3.3
RN-50 + AT	61.5	61.4	60.6	59.4	58.1	56.9	59.6
RN-50 + AT + Ours	62.7	62.7	62.1	60.5	59.4	57.8	60.8
Inc-v3	22.5	37.1	24.4	4.3	3.1	1.9	15.5
Inc-v3 + AT	70.2	70.2	62.1	52.1	35.8	31.4	53.6
Inc-v3 + AT + Ours	74.8	74.9	65.3	55.0	37.8	33.2	56.8

Table 5.9: Adversarial robustness (%) of current pretrained models and our adversarial training methods under various transfer-based black-box attacks. The surrogate model used for adversarial attacks is DN-121. The best results are shown in bold.

attack success rates of 84.7% and 84.3%, respectively, outperforming SODA by clear margins of 5.4% and 2.9%. The performance advantage of PIA is particularly evident on Transformer-based target models (e.g., ViT and ConViT), suggesting that geometric invariance is a more critical factor for transferring attacks to ViTs. However, we observe a slight divergence when Inc-v3 serves as the source model. In this specific case, SODA (80.0%) marginally surpasses PIA (79.4%), possibly due to the architectural characteristics of Inc-v3.

Mechanism Analysis. The performance differences can be explained by the fundamental distinction in how each method augments the input space. While both methods aim to enhance transferability through input transformations, they operate on orthogonal dimensions. **(1) SODA (pixel-intensity modulation):** SODA mainly focuses on modifying the *values* of the pixels. By utilizing soft masking to modulate the image content, it explicitly targets the model’s reliance on specific textures or patterns. **(2) PIA (coordinate-space transformation):** In contrast, PIA focuses on modifying the *positions* of the pixels. By applying perspective transformations, it alters the spatial geometric structure of the input without directly manipulating the pixel values. This forces the generation of adversarial perturbations that are invariant to viewpoint changes.

Combination. Given that SODA and PIA focus on different domains (pixel values vs. geometry), they are naturally complementary. As shown in the “SODA + PIA” rows of Table 5.10, combining these two methods consistently yields the best performance across all source models. For instance, with RN-50 as the source, the combined method achieves an average success rate of 86.2%, further improving the performance of PIA. This empirical evidence confirms that simultaneously perturbing the pixel values and spatial coordinates creates an effective augmentation, effectively preventing the adversarial examples from overfitting to the specific artifacts of the surrogate model.

Gradient Alignment Analysis. To further investigate the underlying cause of the performance gap, we analyze the similarity of the adversarial gradients between the surrogate and target models during the attack process. The gradient alignment is quantified using the cosine similarity, defined as:

$$Sim(\mathbf{g}_{src}, \mathbf{g}_{tgt}) = \frac{\mathbf{g}_{src} \cdot \mathbf{g}_{tgt}}{\|\mathbf{g}_{src}\|_2 \cdot \|\mathbf{g}_{tgt}\|_2},$$

where $\mathbf{g}_{src}$ and $\mathbf{g}_{tgt}$ denote the gradients of the loss function w.r.t the input on the source and target models, respectively. A higher similarity value suggests that the attack direction optimized on the surrogate model is more consistent with the direction required to deceive the target model.

We visualize the cosine similarity across attack iterations in Figure 5.13, with RN-50 as the source and ViT as the target. We find that the gradient similarity of PIA consistently surpasses that of SODA throughout the iterative process. Specifically, at the final iteration, PIA achieves a similarity score of approximately 0.018, whereas SODA only reaches around 0.015. This result suggests that the geometric transformations in PIA serve as a more effective method when optimizing for target models like ViT, filtering out surrogate-specific gradients.

Source	Attack	RN-50	DN-121	Inc-v3	MB-v2	ViT	LeViT	ConViT	PiT	Average
RN-50	SODA	100	99.9	97.2	99.7	49.7	77.3	44.6	65.7	79.3
	PIA	100	99.9	98.3	99.8	61.4	84.2	57.3	76.6	84.7
	SODA + PIA	100	100	98.6	99.8	65.7	86.6	60.6	78.5	86.2
DN-121	SODA	99.6	100	96.9	99.4	56.2	80.8	48.0	70.5	81.4
	PIA	99.7	100	98.1	99.3	61.5	82.9	56.1	76.8	84.3
	SODA + PIA	99.7	100	98.9	99.5	66.8	86.8	61.0	79.1	86.5
Inc-v3	SODA	96.7	97.8	100	96.9	55.0	74.5	48.8	70.2	80.0
	PIA	95.4	97.5	100	96.8	53.6	73.7	50.0	68.3	79.4
	SODA + PIA	97.5	98.2	100	98.0	57.5	77.4	53.7	72.9	82.0

Table 5.10: Attack performance of SODA and PIA on eight target models.

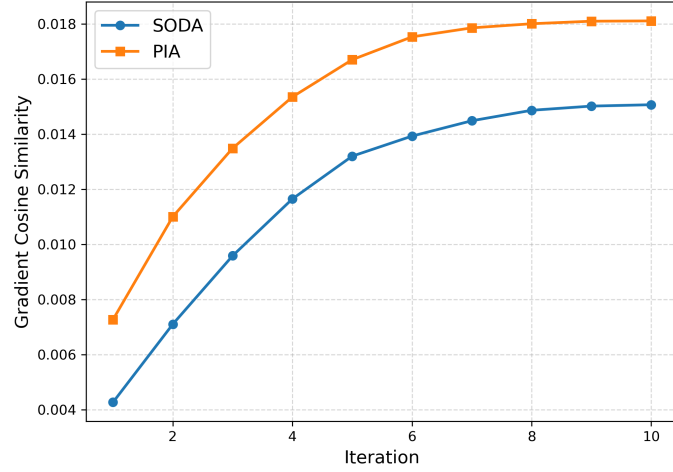


Figure 5.13: Comparison of cross-architecture gradient similarity between SODA and PIA. The curves illustrate the cosine similarity between the gradients computed on the source model (RN-50) and the target model (ViT) across attack iterations.

5.5 Summary

This work reveals that current attacks utilizing input transformations frequently rely on fixed perspective changes, which restrict adversarial transferability. We propose Perspective-Invariant Attack (PIA), a novel method that randomly applies perspective transformations to input images during the attack process. PIA can generate adversarial examples that are invariant to perspective transformations, which are more transferable to other DNN models. Comprehensive experiments demonstrate that PIA can significantly improve the transferability of adversarial examples across different DNN models and datasets. PIA maintains computational efficiency and can be easily integrated into existing attack frameworks, making it practical for real-world applications.

Algorithm 4 Perspective-Invariant Attack (PIA) Algorithm

Input: Surrogate model F , clean image x , true label y , iteration number T , perturbation budget ϵ , step size η , decay factor μ , height h , width w

Input: Maximum distortion budget δ_{max}

Input: Transformation functions set $\mathcal{P}$

Input: Number of transformed copies N

Output: An adversarial example x^{adv}

- 1: $x_0^{\text{adv}} = x, g_0 = 0$.
 - 2: **for** $i = 0$ **to** $T - 1$ **do**
 - 3: $g' = 0$ ▷ Gradients of N transformed images
 - 4: **for** $j = 0$ **to** $N - 1$ **do**
 - 5: Randomly select a function $\phi_j(\cdot)$ from $\mathcal{P}$
 - 6: **if** $\phi_j(\cdot)$ is our $\Phi(\cdot)$ **then** ▷ Get matrix $[h_{ij}]$
 - 7: Sample four corner positions with δ_{max}
 - 8: For each corner pair (u, v) and (u', v') , obtain:
$$u' = \frac{h_{11}u + h_{12}v + h_{13}}{h_{31}u + h_{32}v + 1},$$

$$v' = \frac{h_{21}u + h_{22}v + h_{23}}{h_{31}u + h_{32}v + 1},$$
 - 9: Collect eight equations and solve for $[h_{ij}]$
 - 10: **end if**
 - 11: $g' = g' + \nabla_x \mathcal{L}(F(\phi_j(x_i^{\text{adv}})), y)$
 - 12: **end for**
 - 13: Calculate $g_{i+1} = \mu \cdot g_i + g'_{i+1} / \|g'_{i+1}\|_1$ ▷ Momentum
 - 14: Update example $x_{i+1}^{\text{adv}} = x_i^{\text{adv}} - \eta \cdot \text{sign}(g_{i+1})$
 - 15: Clip x_{i+1}^{adv} to be ℓ_∞ -bounded by ϵ
 - 16: **end for**
 - 17: $x^{\text{adv}} = x_T^{\text{adv}}$
 - 18: **return** x^{adv}
-

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we have proposed a series of methods aimed at enhancing adversarial transferability across different DNN models. The transferability of adversarial examples poses a substantial threat to the reliability and security of current deep learning systems. We systematically examined the limitations and challenges of existing gradient-based transferable attacks under various scenarios and subsequently introduced novel methods to address these issues.

In Chapter 2, we presented a new perspective on the underlying mechanism of adversarial transferability by identifying that non-robust style features in benign images limit the transferability. To mitigate this issue, we proposed Style-less Perturbations (StyLess), which reduces dependence on these non-robust style features, thereby improving the transferability. StyLess introduces stylized surrogate models by inserting adaptive instance normalization layers into traditional surrogate models.

In Chapter 3, we tackled the challenging task of generating targeted attacks, a setting in which existing methods typically suffer from high computational cost and low suc-

cess rates. We introduced Feature Tuning Mixup (FTM), which generates transferable targeted adversarial examples effectively and efficiently. FTM incorporates attack-specific learnable perturbations into the feature space and utilizes a momentum-based stochastic update strategy to optimize multi-layer feature perturbations.

In Chapter 4, we adapted the dropout mechanism to the input space, aiming to introduce adaptive input diversity that improves adversarial transferability. To this end, we proposed Soft Dropout Adaptation (SODA), which dynamically co-adapts two complementary components during attacks: a patch-based soft mask that perturbs image content, and a lightweight border prompt that enriches contextual cues. SODA enhances transfer-based attacks without additional data or significant computational overhead.

In Chapter 5, we revisited commonly used input transformation-based attack methods and highlighted their limitation of relying on fixed perspective transformations. To address this issue, we introduced Perspective-Invariant Attack (PIA), which is capable of producing adversarial examples that remain unaffected by perspective alterations. PIA significantly improves transferability and can be seamlessly combined with other attack strategies without losing computational efficiency.

6.2 Future Work

This section explores possible extensions to the current work and highlights promising directions for future research.

Surrogate model design for enhanced transferability. Most existing studies assume that attackers rely on publicly available pre-trained models as surrogate models. However, the choice of surrogate models plays a critical role in the success of transfer-based attacks. Our work demonstrates that inserting adaptive instance normalization layers into surrogate models can flexibly simulate diverse surrogate models. Never-

theless, how to design and train surrogate models that are inherently more effective for generating transferable adversarial examples remains an open question.

Adversarial transferability of targeted attacks. Generating targeted adversarial examples that transfer effectively to other DNN models, especially across different architectures, remains a significant challenge. While our work substantially improves the transferability of targeted attacks, there is still considerable room for further advancement. Given the high potential threat posed by transferable targeted attacks, it is crucial to continue exploring this direction.

Adversarial transferability across multimodal LLMs. Large language models (LLMs) have achieved remarkable success across a wide range of tasks and are rapidly becoming integral to many applications. The integration of vision and LLMs brings tremendous convenience but also raises serious security concerns. The study of adversarial examples in the context of multimodal LLMs remains largely unexplored. Adapting existing unimodal attack methods to multimodal LLMs, as well as designing attacks from a multimodal perspective, are both promising and urgent directions for future research.

Theoretical analysis of adversarial transferability. The lack of rigorous theory is a common gap in the field of deep learning, and adversarial transferability is no exception. Most existing methods on adversarial transferability are empirical; even when theoretical insights are provided, they often lack rigorous mathematical proof. A solid theoretical understanding of adversarial transferability and current attack mechanisms would not only help explain their underlying principles but also uncover hidden vulnerabilities in DNNs. Such analysis would be critical for designing more robust and reliable models in the future.

References

- [1] Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. In *NeurIPS*, 2020.
- [2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.
- [3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *ICML*, 2018.
- [4] Jimmy Ba and Brendan Frey. Adaptive dropout for training deep neural networks. In *NeurIPS*, 2013.
- [5] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.
- [6] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022.
- [7] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. In *NeurIPS*, 2022.
- [8] Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. Revisiting batch normalization for improving corruption robustness. In *WACV*, 2021.

- [9] Philipp Benz, Chaoning Zhang, and In So Kweon. Batch normalization increases adversarial vulnerability and decreases adversarial transferability: A non-robust feature perspective. In *ICCV*, 2021.
- [10] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *ICLR*, 2018.
- [11] Junyoung Byun, Seungju Cho, Myung-Joon Kwon, Hee-Seon Kim, and Changick Kim. Improving the transferability of targeted adversarial examples through object-based diverse input. In *CVPR*, 2022.
- [12] Junyoung Byun, Myung-Joon Kwon, Seungju Cho, Yoonji Kim, and Changick Kim. Introducing competition to boost the transferability of targeted adversarial examples through clean feature mixup. In *CVPR*, 2023.
- [13] Nicholas Carlini, Florian Tramer, Krishnamurthy Dj Dvijotham, Leslie Rice, Mingjie Sun, and J Zico Kolter. (certified!!) adversarial robustness for free! In *ICLR*, 2023.
- [14] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, 2017.
- [15] Bin Chen, Jiali Yin, Shukai Chen, Bohao Chen, and Ximeng Liu. An adaptive model ensemble adversarial attack for boosting adversarial transferability. In *ICCV*, 2023.
- [16] Huanran Chen, Yichi Zhang, Yinpeng Dong, Xiao Yang, Hang Su, and Jun Zhu. Rethinking model ensemble in transfer-based adversarial attacks. In *ICLR*, 2024.
- [17] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017.

-
- [18] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *NeurIPS*, 2021.
 - [19] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.
 - [20] Xuelong Dai, Kaisheng Liang, and Bin Xiao. Advdiff: Generating unrestricted adversarial examples using diffusion models. In *ECCV*, 2024.
 - [21] Stéphane d’Ascoli, Hugo Touvron, Matthew L. Leavitt, Ari S. Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *ICML*, 2021.
 - [22] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018.
 - [23] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR*, 2019.
 - [24] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.
 - [25] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A. K. Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *CVPR*, 2020.
 - [26] Ranjie Duan, Xiaofeng Mao, A Kai Qin, Yuefeng Chen, Shaokai Ye, Yuan He, and Yun Yang. Adversarial laser beam: Effective physical-world attack to dnns in a blink. In *CVPR*, 2021.

- [27] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *ICLR*, 2017.
- [28] Gamaleldin F Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. In *ICLR*, 2019.
- [29] Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *CVPR*, 2018.
- [30] Mingyuan Fan, Xiaodan Li, Cen Chen, Wenmeng Zhou, and Yaliang Li. Transferability bound theory: Exploring relationship between adversarial transferability and flatness. In *NeurIPS*, 2024.
- [31] Hao Fang, Jiawei Kong, Bin Chen, Tao Dai, Hao Wu, and Shu-Tao Xia. Clip-guided generative networks for transferable targeted adversarial attacks. In *ECCV*, 2024.
- [32] Chun-Mei Feng, Kai Yu, Yong Liu, Salman Khan, and Wangmeng Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *ICCV*, 2023.
- [33] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*, 2020.
- [34] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.
- [35] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.

- [36] Zhijin Ge, Hongying Liu, Wang Xiaosen, Fanhua Shang, and Yuanyuan Liu. Boosting adversarial transferability by achieving flat local maxima. In *NeurIPS*, 2023.
- [37] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.
- [38] Ross Girshick. Fast r-cnn. In *ICCV*, 2015.
- [39] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.
- [40] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.
- [41] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet’s clothing for faster inference. In *ICCV*, 2021.
- [42] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *ICLR*, 2018.
- [43] Yiwen Guo, Qizhang Li, and Hao Chen. Backpropagating linearly improves transferability of adversarial examples. In *NeurIPS*, 2020.
- [44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [45] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *ICCV*, 2021.

- [46] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
- [47] Hossein Hosseini and Radha Poovendran. Semantic adversarial examples. In *CVPR Workshops*, 2018.
- [48] Shengshan Hu, Xiaogeng Liu, Yechao Zhang, Minghui Li, Leo Yu Zhang, Hai Jin, and Libing Wu. Protecting facial privacy: Generating adversarial identity masks via style-robust makeup transfer. In *CVPR*, 2022.
- [49] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.
- [50] Lifeng Huang, Chengying Gao, Yuyin Zhou, Cihang Xie, Alan L. Yuille, Changqing Zou, and Ning Liu. Universal physical camouflage attacks on object detectors. In *CVPR*, 2020.
- [51] Qian Huang, Isay Katsman, Zeqi Gu, Horace He, Serge J. Belongie, and Ser-Nam Lim. Enhancing adversarial example transferability with an intermediate level attack. In *ICCV*, 2019.
- [52] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.
- [53] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *ICML*, 2018.
- [54] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019.
- [55] Nathan Inkawhich, Kevin J. Liang, Lawrence Carin, and Yiran Chen. Transferable perturbations of deep feature distributions. In *ICLR*, 2020.

-
- [56] Nathan Inkawhich, Kevin J. Liang, Binghui Wang, Matthew Inkawhich, Lawrence Carin, and Yiran Chen. Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability. In *NeurIPS*, 2020.
 - [57] Nathan Inkawhich, Wei Wen, Hai (Helen) Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *CVPR*, 2019.
 - [58] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.
 - [59] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022.
 - [60] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. Comdefend: An efficient image compression model to defend adversarial examples. In *CVPR*, 2019.
 - [61] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.
 - [62] Sanjay Kariyappa and Moinuddin K Qureshi. Improving adversarial robustness of ensembles with diversity training. *arXiv preprint arXiv:1901.09981*, 2019.
 - [63] Hoki Kim, Woojin Lee, and Jaewook Lee. Understanding catastrophic overfitting in single-step adversarial training. In *AAAI*, 2021.
 - [64] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *NeurIPS*, 2015.
 - [65] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
 - [66] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

- [67] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *ICLR (Workshop)*, 2017.
- [68] Cassidy Laidlaw and Soheil Feizi. Functional adversarial attacks. In *NeurIPS*, 2019.
- [69] Maosen Li, Cheng Deng, Tengjiao Li, Junchi Yan, Xinbo Gao, and Heng Huang. Towards transferable targeted attack. In *CVPR*, 2020.
- [70] Qizhang Li, Yiwen Guo, and Hao Chen. Practical no-box adversarial attacks against dnns. In *NeurIPS*, 2020.
- [71] Qizhang Li, Yiwen Guo, and Hao Chen. Yet another intermediate-level attack. In *ECCV*, 2020.
- [72] Qizhang Li, Yiwen Guo, Wangmeng Zuo, and Hao Chen. Making substitute models more bayesian can enhance transferability of adversarial examples. In *ICLR*, 2023.
- [73] Yanjie Li, Yiquan Li, Xuelong Dai, Songtao Guo, and Bin Xiao. Physical-world optical adversarial attacks on 3d face recognition. In *CVPR*, 2023.
- [74] Yanjie Li, Kaisheng Liang, and Bin Xiao. UV-attack: Physical-world adversarial attacks on person detection via dynamic-NeRF-based UV mapping. In *ICLR*, 2025.
- [75] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *NeurIPS*, 2017.
- [76] Yingwei Li, Song Bai, Yuyin Zhou, Cihang Xie, Zhishuai Zhang, and Alan Yuille. Learning transferable adversarial examples via ghost networks. In *AAAI*, 2020.
- [77] Kaisheng Liang, Xuelong Dai, Yanjie Li, Dong Wang, and Bin Xiao. Improving transferable targeted attacks with feature tuning mixup. In *CVPR*, 2025.

-
- [78] Kaisheng Liang and Bin Xiao. Styleless: Boosting the transferability of adversarial examples. In *CVPR*, 2023.
- [79] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *CVPR*, 2018.
- [80] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *ICLR*, 2020.
- [81] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *ICLR*, 2017.
- [82] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In *CVPR*, 2019.
- [83] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.
- [84] Yuyang Long, Qilong Zhang, Boheng Zeng, Lianli Gao, Xianglong Liu, Jian Zhang, and Jingkuan Song. Frequency domain model augmentation for adversarial attack. In *ECCV*, 2022.
- [85] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, 2018.
- [86] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.
- [87] Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. On the robustness of vision transformers to adversarial examples. In *ICCV*, 2021.

- [88] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deep-fool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016.
- [89] Muhammad Muzammal Naseer, Salman H Khan, Muhammad Haris Khan, Fahad Shahbaz Khan, and Fatih Porikli. Cross-domain transferability of adversarial perturbations. In *NeurIPS*, 2019.
- [90] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *CVPR*, 2020.
- [91] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. On generating transferable targeted perturbations. In *ICCV*, 2021.
- [92] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. Stylized adversarial defense. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022.
- [93] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *ICML*, 2022.
- [94] Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *ICLR*, 2021.
- [95] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.
- [96] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *CVPR*, 2018.

- [97] Zeyu Qin, Yanbo Fan, Yi Liu, Li Shen, Yong Zhang, Jue Wang, and Baoyuan Wu. Boosting the transferability of adversarial attacks with reverse adversarial perturbation. In *NeurIPS*, 2022.
- [98] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.
- [99] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 2015.
- [100] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.
- [101] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision (IJCV)*, 115:211–252, 2015.
- [102] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? In *NeurIPS*, 2020.
- [103] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.
- [104] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *NeurIPS*, 2020.
- [105] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

- [106] Ali Shahin Shamsabadi, Ricardo Sanchez-Matilla, and Andrea Cavallaro. Colorfool: Semantic adversarial colorization. In *CVPR*, 2020.
- [107] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *CCS*, 2016.
- [108] Xu Shen, Xinmei Tian, Tongliang Liu, Fang Xu, and Dacheng Tao. Continuous dropout. *IEEE transactions on neural networks and learning systems*, 29(9):3926–3937, 2017.
- [109] Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. What does clip know about a red circle? visual prompt engineering for vlms. In *ICCV*, 2023.
- [110] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. In *NeurIPS*, 2022.
- [111] Manli Shu, Zuxuan Wu, Micah Goldblum, and Tom Goldstein. Encoding robustness to image style via adversarial feature perturbations. In *NeurIPS*, 2021.
- [112] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- [113] Jacob M. Springer, Melanie Mitchell, and Garrett T. Kenyon. A little robustness goes a long way: Leveraging robust features for targeted transfer attacks. In *NeurIPS*, 2021.
- [114] Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, et al. Towards efficient and effective adversarial training. In *NeurIPS*, 2021.

- [115] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [116] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NeurIPS*, 2014.
- [117] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.
- [118] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.
- [119] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.
- [120] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.
- [121] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, 2019.
- [122] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.
- [123] Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In *NeurIPS*, 2020.

- [124] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *ICLR*, 2018.
- [125] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016.
- [126] Francisco Utrera, Evan Kravitz, N Benjamin Erichson, Rajiv Khanna, and Michael W Mahoney. Adversarially-trained deep nets transfer better: Illustration on image classification. In *ICLR*, 2020.
- [127] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [128] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *ICML*, 2013.
- [129] Kunyu Wang, Xuanran He, Wenxuan Wang, and Xiaosen Wang. Boosting adversarial transferability by block shuffle and rotation. In *CVPR*, 2024.
- [130] Shuo Wang, Surya Nepal, Alsharif Abuadbba, Carsten Rudolph, and Marthie Grobler. Adversarial detection by latent style transformations. *IEEE Transactions on Information Forensics and Security (TIFS)*, 17:1099–1114, 2022.
- [131] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *CVPR*, 2021.
- [132] Xiaosen Wang, Xuanran He, Jingdong Wang, and Kun He. Admix: Enhancing the transferability of adversarial attacks. In *ICCV*, 2021.
- [133] Xiaosen Wang, Zeliang Zhang, and Jianping Zhang. Structure invariant transformation for better adversarial transferability. In *ICCV*, 2023.

-
- [134] Zhibo Wang, Hengchang Guo, Zhifei Zhang, Wenxin Liu, Zhan Qin, and Kui Ren. Feature importance-aware transferable adversarial attacks. In *ICCV*, 2021.
 - [135] Zhibo Wang, Hongshan Yang, Yunhe Feng, Peng Sun, Hengchang Guo, Zhifei Zhang, and Kui Ren. Towards transferable targeted adversarial examples. In *CVPR*, 2023.
 - [136] Zhipeng Wei, Jingjing Chen, Micah Goldblum, Zuxuan Wu, Tom Goldstein, and Yu-Gang Jiang. Towards transferable adversarial attacks on vision transformers. In *AAAI*, 2022.
 - [137] Zhipeng Wei, Jingjing Chen, Zuxuan Wu, and Yu-Gang Jiang. Enhancing the self-universality for transferable targeted attacks. In *CVPR*, 2023.
 - [138] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.
 - [139] Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. In *ICLR*, 2020.
 - [140] Weibin Wu, Yuxin Su, Michael R Lyu, and Irwin King. Improving the transferability of adversarial samples with adversarial transformations. In *CVPR*, 2021.
 - [141] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.
 - [142] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *ICLR*, 2018.
 - [143] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. Improving transferability of adversarial examples with input diversity. In *CVPR*, 2019.

- [144] Yifeng Xiong, Jiadong Lin, Min Zhang, John E. Hopcroft, and Kun He. Stochastic variance reduced ensemble adversarial attack for boosting the adversarial transferability. In *CVPR*, 2022.
- [145] Qiuling Xu, Guanhong Tao, Siyuan Cheng, and Xiangyu Zhang. Towards feature space adversarial attack by style perturbation. In *AAAI*, 2021.
- [146] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *NDSS*, 2018.
- [147] Huanrui Yang, Jingyang Zhang, Hongliang Dong, Nathan Inkawhich, Andrew Gardner, Andrew Touchet, Wesley Wilkes, Heath Berry, and Hai Li. Dverge: diversifying vulnerabilities for enhanced robust generation of ensembles. In *NeurIPS*, 2020.
- [148] Xiao Yang, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Boosting transferability of targeted adversarial examples via hierarchical generative networks. In *ECCV*, 2022.
- [149] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.
- [150] Jianping Zhang, Yizhan Huang, Weibin Wu, and Michael R Lyu. Transferable adversarial attacks on vision transformers with token gradient regularization. In *CVPR*, 2023.
- [151] Jianping Zhang, Weibin Wu, Jen-tse Huang, Yizhan Huang, Wenxuan Wang, Yuxin Su, and Michael R Lyu. Improving adversarial transferability via neuron attribution-based attacks. In *CVPR*, 2022.
- [152] Shengming Yuan Qilong Zhang, Lianli Gao, Yaya Chen, and Jingkuan Song. Natural color fool: Towards boosting black-box unrestricted attacks. In *NeurIPS*, 2022.

- [153] Yechao Zhang, Shengshan Hu, Leo Yu Zhang, Junyu Shi, Minghui Li, Xiaogeng Liu, Wei Wan, and Hai Jin. Why does little robustness help? a further step towards understanding adversarial transferability. In *IEEE Symposium on Security and Privacy (SP)*, 2024.
- [154] Mengnan Zhao, Lihe Zhang, Yuqiu Kong, and Baocai Yin. Fast adversarial training with smooth convergence. In *ICCV*, 2023.
- [155] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. Adversarial color enhancement: Generating unrestricted adversarial images by optimizing a color filter. In *BMVC*, 2020.
- [156] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. On success and simplicity: A second look at transferable targeted attacks. In *NeurIPS*, 2021.
- [157] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *ECCV*, 2018.
- [158] Rongyi Zhu, Zeliang Zhang, Susan Liang, Zhuo Liu, and Chenliang Xu. Learning to transform dynamically for better adversarial transferability. In *CVPR*, 2024.
- [159] Yao Zhu, Jiacheng Sun, and Zhenguo Li. Rethinking adversarial transferability from a data distribution perspective. In *ICLR*, 2021.
- [160] Junhua Zou, Zhisong Pan, Junyang Qiu, Xin Liu, Ting Rui, and Wei Li. Improving the transferability of adversarial examples with resized-diverse-inputs, diversity-ensemble and region fitting. In *ECCV*, 2020.