IMPERCEPTIBLE ADVERSARIAL ATTACKS IN IMAGES

ZEYU DAI

PhD

The Hong Kong Polytechnic University

2026

The Hong Kong Polytechnic University
Department of Computing

Imperceptible Adversarial Attacks in Images

Zeyu Dai

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
May 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: Zeyu Dai

Abstract

In the last decades, benefiting from massive data and huge computation resources, deep neural networks (DNNs) have achieved significant progress in wide applications, such as image classification, object detection, natural language processing, and machine translation. Especially in image domain, DNNs with plenty of hidden layers and complicated architectures can capture useful and semantic visual patterns, and realize various tasks with a high confidence.

Despite these achievements, the safety and robustness of DNN systems become critical and attract more attention from research community. In 2013, DNNs were first found vulnerable to adversarial examples (AEs), that a slight perturbation of the images can lead to classifiers making erroneous prediction. This discovery unveiled the vulnerability of DNNs, which is a critical threat especially for some security-sensitive scenarios, such as autonomous driving where DNNs are used to recognize road signs. Adversarial attack is the process of generating AEs. It has two main goals: 1) attack successfulness, i.e., generate AEs that can successfully fool DNN models; 2) imperceptibility, i.e., ensure the difference between the generated AEs and real-world images imperceptible.

Although existing works have achieved very high attack success rate (ASR) in various attack settings, there is still a gap between the generated AEs and real-world images in terms of imperceptibility. On the one hand, achieving imperceptibility is basically contradictory to the goal of attack successfulness, most works achieve high ASR at the

cost of imperceptibility performance. On the other hand, how to define and evaluate imperceptibility is still an open question. Most works use L_p norms, such as L_0, L_2 and L_∞ to quantify the magnitude of the perturbation added to the original images. However, L_p norms are found non-suitable for simulating human vision system (HVS) in subjective experiments, not to mention some special attack tasks and settings where L_p norms are not applicable.

To fill in the research gap, this thesis aims to explore and enhance imperceptible adversarial attacks systematically. First, we propose a thorough taxonomy of AEs based on their mathematical definitions to classify all types of AEs into three main categories, i.e., input-specific AE, input-agnostic AE and input-free AE. Input-specific AE is generated based on a specific clean image, input-agnostic AE refers to apply a universal perturbation or transformation to any clean image to be adversarial, and input-free AE does not rely on given clean images and can be generated infinitely. Moreover, we study the imperceptibility of AEs from two perspectives: human imperceptibility and machine imperceptibility. Human imperceptibility focuses on the human capability to recognize AEs from real-world images, while machine imperceptibility refers to the research on whether machines can detect AEs.

Based on the above investigation, we then explore different imperceptible adversarial attacks for three types of AEs in image classification domain, considering both human and machine imperceptibility:

1. Our first work focuses on black-box input-specific attack setting, where global perturbation and high-frequency noises are used to pursue query-efficient attacks at the cost of imperceptibility. We propose Saliency Attack to restrict the perturbations to a small but classifier-sensitive region, and further refine the perturbations to generate efficient and imperceptible input-specific AEs in black-box setting.

2. Our second work studies both human and machine imperceptibility in input-

agnostic attack setting. Considering the interpretation discrepancy between the clean images and their AEs that could be used to detect AEs, we propose Joint Universal Adversarial Perturbations (JUAP) to generate imperceptible universal perturbations by jointly optimizing the adversarial loss and interpretation discrepancy.

3. Our third work explores the most general input-free AEs, which are generated without any input clean images. Given the poor naturalness of existing input-free attack methods, we propose SemDiff to utilize the semantic latent space of diffusion models to generate natural input-free AEs with semantically meaningful attribute shifts.

Extensive experiments on various tasks and datasets demonstrate the effectiveness of our proposed methods in generating more imperceptible and threatening AEs for different types of AEs, which also provide new insights into the robustness and security of DNNs.

Publications Arising from the Thesis

1. Zeyu Dai, Shengcai Liu, Qing Li, and Ke Tang, "Saliency attack: towards imperceptible black-box adversarial attack", in *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2023.

2. Liangbo Ning*, Zeyu Dai*, Wenqi Fan, Jingran Su, Chao Pan, Luning Wang, and Qing Li, "Joint Universal Adversarial Perturbations with Interpretations", manuscript submitted to *30th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2026.

3. Zeyu Dai, Shengcai Liu, Rui He, Jiahao Wu, Ning Lu, Wenqi Fan, Qing Li, and Ke Tang, "SemDiff: Generating Semantic Unrestricted Adversarial Examples with Diffusion Models", manuscript submitted to *IEEE Transactions on Image Processing (TIP)*.

4. Liang-bo Ning, Zeyu Dai, Jingran Su, Chao Pan, Luning Wang, Wenqi Fan, and Qing Li, "Interpretation-empowered neural cleanse for backdoor attacks", in *Companion Proceedings of the ACM Web Conference (WWW)*, 2024.

5. Zeyu Dai, Wei Fang, Ke Tang, and Qing Li, "An optima-identified framework with brain storm optimization for multimodal optimization problems" in *Swarm and Evolutionary Computation*, 2021.

6. Zeyu Dai, Wei Fang, Qing Li, and Wei-neng Chen, "Modified self-adaptive brain storm optimization algorithm for multimodal optimization", in *International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, 2019.

7. Xueying Zhan*, Zeyu Dai*, Qingzhong Wang, Qing Li, Haoyi Xiong, Dejing Dou, and Antoni B Chan, "Pareto Optimization for Active Learning under Out-of-Distribution Data Scenarios", in *Transactions on Machine Learning Research (TMLR)*, 2023.

8. Rui He, Zeyu Dai, Shan He, and Ke Tang, "Perturbation-Based Two-Stage Multi-Domain Active Learning", in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM)*, 2023.

Acknowledgments

As my thesis is about to be completed, I'm finally coming to the end of my PhD study, which has been nearly six years. It is a long, not very smoothing, and unforgettable journey. As a student of the Joint PhD Training Program offered by the The Hong Kong Polytechnic University (PolyU) and Southern University of Science and Technology (SUSTech), I would like to express my heartfelt gratitude to my supervisors Prof. Qing Li in PolyU and Prof. Ke Tang in SUSTech for their unwavering support throughout my academic journey. Their professional guidance, patient encouragement, and kindness not only trained my research skills but also shaped the way I think about problems. Without their inclusion, I would not have been able to explore freely during my PhD and complete my thesis. I am very grateful for them both on my academic path and my life journey. I would also like to thank PloyU and SUSTech for their support for my studies in both universities.

I also want to acknowledge Prof. Wei Fang in Jiangnan University and Prof. Shi Cheng in Shaanxi Normal University for their help and guidance in my early academic career. Without their support, I would not publish my first research paper and start my academic journey. I would like to thank Prof. Shengcai Liu in SUSTech and Prof. Wenqi Fan for their academic guidance and support during my PhD study. I would also like to thank many administrative staffs, Mr. IU Jason and Ms. HO Vivian (PolyU executive officers), Ms. Juan Tang, Ms. Hanyu Wei, Ms. Ruisi Ding (NICAL Lab administrators), Mr. Xing Zhang (SUSTech graduate student administrator).

I am fortunate to have the support and friendship of numerous friends and colleagues during my studies at PolyU and SUSTech. I want to express my gratitude to Dr. Xueying Zhan, Yaowei Wang, Liangbo Ning, Dr. Da Ren, Dr. Zehang Lin, Han Zhang, Xingming Zhang, Chao Pan, Dr. Chaoyu Liu, Dr. Zhejing Hu and Dr. Bruce X.B. Yu from PolyU, and Dr. Peng Yang, Dr. Guiyin Li, Dr. Wenjing Hong, Dr. Yu Zhang, Dr. Rui He, Fu Peng, Lan Tang, Jiahao Wu, Zhiyuan Wang, Ning Lu, Xuanfeng Li, Hui Ouyang, Shaofeng Zhang, Qixin, Guo, Qi Yang, Muyao Zhong from SUSTech. Your friendship and mentorship have been invaluable to my growth and learning.

I would like to thank my loving parents and my family for bestowing life and well-being upon me enabling me to explore this wonderful world and pursue my passions, always supporting, and caring about me. My gratitude is beyond words.

I am grateful to my beloved girlfriend for her accompany, support, and patience with me. Accompanied me through my last two tough years. I sincerely hope that we can explore this colorful world together in the future.

Lastly, I want to thank myself. I get through this long journey.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Deep Learning and Deep Neural Networks in Image Domain

Machine Learning (ML), as a subfield of Artificial Intelligence (AI), focuses on enabling systems to learn patterns from data and make predictions or decisions with minimal human intervention. ML techniques process data to perform various tasks, such as classification, regression, clustering, dimensionality reduction, and ranking. As the wild application of ML in our daily life, higher accuracy and adaptability are required in real-world applications, while traditional machine learning methods often fall short in handling the complexity of modern datasets. Deep learning (DL), with its layered architecture and end-to-end learning capabilities, has become powerful approach for tackling these challenges.

Over the last decade, benefiting from massive data and huge computation resources, deep neural network (DNNs) have achieved significant progress in various applications, such as computer vision [141, 145], natural language processing [69], and speech recognition [173]. Especially in image domain, early breakthroughs, such as

the development of convolutional neural networks (CNNs) [93] in the 1990s, laid the groundwork for modern architectures. In 2012, AlexNet [88] achieved human-level performance for the first time in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [36]. Subsequent works, like VGG [162], DenseNet [72], and ResNet [13], have further improved the performance of image classification task by designing deeper and more complex architectures to capture useful and semantic visual patterns. Beyond image classification, DNNs have further expanded the capabilities in tasks such as face recognition [141], object detection [145] and semantic segmentation [150].

1.2 Adversarial Examples and Imperceptibility

In 2013, Szegedy et al. [173] first revealed the existence of *adversarial example* in deep neural networks. This seminal work discovered that small and human-imperceptible perturbations added to the input images can lead to image classifiers making erroneous predictions, which raised significant concerns about the robustness and security of DNNs. Subsequently, researchers propose a variety of *adversarial attack* methods to generate adversarial examples to fool target models, considering different attack settings including *white-box*, *grey-box* and *black-box* settings according to the attacker's knowledge of the target model [201].

Adversarial Examples widely exist in diverse ML tasks and domains [3, 211, 201, 116]. Researchers have demonstrated that adversarial examples can be generated in image classification [173, 56, 20], object detection [199], semantic segmentation [64], and face recognition [176, 43]. In addition to the above tasks on 2D digital images, attackers also explore the adversarial examples in 3D objects [89, 159] in the physical world and timing-based videos [78, 190]. Beyond image domain, the vulnerability of DNNs to adversarial examples has also been unveiled in text [111], audio [207], tabular data [7], robotic sensory data [87], etc.

The existence of adversarial examples poses a critical threat to the deployment of DNNs in security-sensitive scenarios. For example, in autonomous driving, DNNs are used to recognize road signs and make decisions based on the recognition results. As demonstrated by Eykholt et al. [47], strategically placed stickers on a "Stop" sign can mislead DNN-based classifiers to predict "Speed Limit 45" sign with high confidence under varying illumination conditions. Xu et al. [201] show that a person, wearing a special T-shirt with adversarial patterns, can evade pedestrian detectors to be invisible. Furthermore, Finlayson et al. [52] revealed that attackers can change a patient's diagnosis to cancer from benign by adding adversarial noise to the medical images. All these examples demonstrate the potential risks and severe consequences of adversarial examples in real-world applications. It is crucial to study adversarial examples and develop robust and secure DNN systems to avoid such threats.

Imperceptibility serves as a fundamental criterion for adversarial examples. While random noises may also achieve model deception, their artificial nature renders them theoretically insignificant. Only those adversarial examples that are natural and legitimate to human eyes are meaningful to research on. Since early works [173, 56, 140, 20, 127, 131, 75] primarily generated adversarial examples through additive perturbations, the imperceptibility of adversarial examples here denotes the indistinguishability between the adversarial examples and the corresponding clean images. Researchers often measure the magnitude of perturbations using L_p norms, such as L_0, L_2 and L_∞, to quantify the imperceptibility of adversarial examples [140, 20, 56].

With the evolution of adversarial machine learning, diverse varients of adversarial examples are proposed. Researchers do not limit to produce adversarial examples indistinguishable from clean images, but design various adversarial examples as long as they are natural-looking to human eyes. For example, Xiao et al. [198] apply spatial transformations to clean images to deceive target models. Although the generated adversarial examples are far from the clean images in pixel space with large L_p distance, they are still natural and realistic to humans. Moreover, Song et al. [166]

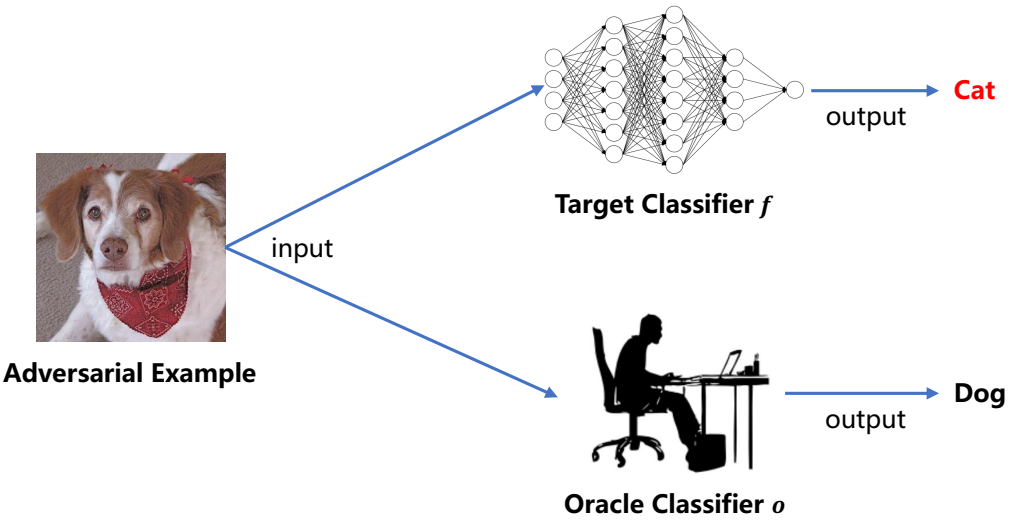


Figure 1.1: **Adversarial Example.** It shows an adversarial example can be misclassified by a DNN-based target classifier, but it still looks normal to an oracle classifier, such as human judgement.

propose unrestricted adversarial examples that can be generated by generative models without given input images, which means there is no reference image to compare the difference. Therefore, the explanation of imperceptibility extends from the indistinguishability to the natural looking of adversarial examples. In other words, the distance between a good adversarial example and the distribution of real-world images should be imperceptible. Hence, a more general definition of adversarial examples becomes:

$$\mathcal{A} \triangleq \{\boldsymbol{x} \in \mathcal{O} \mid \boldsymbol{o}(\boldsymbol{x}) \neq \boldsymbol{f}(\boldsymbol{x})\}, \tag{1.1}$$

where $\mathcal{O}$ is the set of all real-world images that look realistic to humans. $\boldsymbol{f}$ is the target model to be attacked, and $\boldsymbol{o}$ represents an oracle model that provides ground-truth results. An adversarial example can cause wrong prediction of the target model while looking realistic to human eyes, as shown in Fig. 1.1.

Despite achieving very high attack success rate (ASR) in various attack settings [20, 5, 16, 126, 166], a significant discrepancy persists between the generated adversarial

examples and real-world images in terms of imperceptibility. This challenge arises from two reasons:

- **Attack-Imperceptibility Trade-off**: Achieving imperceptibility is basically contradictory to the goal of attack successfulness - maximizing attack success rate (ASR) while minimizing the perceptual distance between the adversarial example and the clean image. Current works often obtain high ASR with various tricks while compromising imperceptibility performance [125, 5, 179] (see Fig. 3.1).

- **Imperceptibility Evaluation Ambiguity**: So far, how to define and evaluate imperceptibility is still an open question. Most works use L_p norms, such as L_0, L_2 and L_∞ to bound perturbation magnitudes to guarantee imperceptibility [140, 20, 56]. However, L_p norms are found non-suitable for simulating human vision system in subjective experiments [158, 51, 210], not to mention some special attack tasks and settings where L_p norms are not applicable as introduced above [198, 166].

Considering the above issues, it is essential to thoroughly investigate the imperceptibility of adversarial examples in different attack settings and tasks, and design more effective imperceptible adversarial attacks to promote the research on adversarial machine learning.

1.3 Objectives and Research Questions

The thesis aims to systematically explore and enhance imperceptible adversarial attacks, that can generate adversarial examples with high attack success rate while maintaining their imperceptibility. To achieve the goal, a comprehensive investigation on the imperceptibility of adversarial examples in different attack settings and

tasks is required. Although imperceptibility is usually involved in an adversarial attack work, a systematic study on imperceptibility is still lacking. Meanwhile, different attack settings and tasks may possess distinct imperceptibility requirements, and the imperceptibility evaluation metrics may vary accordingly. Therefore, the first objective of this thesis is to survey the imperceptibility of adversarial examples in image domain and the imperceptibility problems in different attack settings and tasks. The second objective is to propose new approaches to mitigate the attack-imperceptibility trade-off and enhance both the attack success rate and imperceptibility performance in different attack settings and tasks. Specifically, we aim to explore the following three main research questions (RQs):

- **RQ1: How to improve the imperceptibility in the query-sensitive black-box input-specific adversarial attacks?** In comparison to white-box attacks, black-box attacks only have access to the model output, which is a more practical and difficult setting. Considering many real-world application programming interfaces (APIs) pose mandatory time or monetary limits to user queries [75], most black-box attack works pursue query-efficient attacks through various tricks, such as global perturbation and random vertical stripes [125, 5]. As a result, the generated adversarial examples contain obvious and perceptible perturbations. On the other hand, current online APIs usually integrate detectors into their services to detect anomalous inputs [103]. These adversarial examples with too visible perturbation are difficult to pass the detectors, not to mention human judgement. Hence, how to design imperceptible black-box attacks while maintaining high attack success rate and query efficiency is a critical challenge.

- **RQ2: Facing interpretation-based detectors, how to achieve both human and machine imperceptibility in the input-agnostic adversarial attacks?** Attribution-based interpretation methods usually produce a saliency map to evaluate the sensitivity of network classification decisions with respect

to the pixels in an input image [219]. Previous works found the phenomenon of interpretation discrepancy that an imperceptible adversarial example to deceive classifiers can lead to a significant change in a class-specific interpretation saliency map [202, 12]. Hence, some adversarial defense methods are proposed to utilize the interpretation discrepancy to detect adversarial examples [174, 183, 29], which poses a challenge for attackers to evade machine detection. We wonder whether we can generate adversarial examples that can achieve both human and machine imperceptibility, namely the imperceptible perturbation can also maintain the interpretation consistency with the clean images. In addition, we consider a more difficult attack setting, input-agnostic adversarial attacks, where the attacker require generating a universal perturbation to fool the target model on any clean image.

- **RQ3: Given no input images, how to generate natural-looking and imperceptible adversarial examples in the input-free adversarial attacks?** Different from input-specific and input-agnostic adversarial attacks, input-free adversarial attacks do not rely on given input images and can generate adversarial examples infinitely [166]. This kind of attack poses a severe threat to the safety of deep learning models, especially in some data-expensive attack scenarios, such as medical images [74], where collecting clean samples is difficult and expensive. Existing works use generative models to generate input-free adversarial examples, but are still limited in naturalness [33, 24]. In other words, the generated adversarial examples are still far from the distribution of real-world images. Therefore, how to generate natural-looking and imperceptible adversarial examples without given input images is a challenging problem.

1.4 Contributions and Thesis Structure

Given the above objectives and research questions, this thesis presents the following main contributions. A comprehensive literature review on adversarial examples and imperceptibility in image domain is presented in Chapter 2. Motivated by the different imperceptibility requirements of adversarial examples in various attack settings and tasks, we first propose a comprehensive taxonomy of adversarial examples with clear mathematical definitions to classify all types of adversarial examples into three main categories, i.e., *input-specific adversarial examples*, *input-agnostic adversarial examples* and *input-free adversarial examples* (see Section 2.1). Input-specific adversarial example is generated based on a specific clean image, input-agnostic adversarial example refers to apply a universal perturbation [126] or transformation [130] to any clean image to be adversarial, and input-free adversarial example does not rely on given clean images and can be generated infinitely.

Furthermore, we provide a detailed investigation on the imperceptibility of adversarial examples from two perspectives: *human imperceptibility* and *machine imperceptibility* (see Section 2.2). Human imperceptibility focuses on the human capability to recognize adversarial examples from real-world images, with varied imperceptibility in different type of adversarial examples. For example, input-specific and input-agnostic adversarial examples usually require indistinguishability from original examples, while input-free adversarial examples need to be natural-looking and realistic. Machine imperceptibility refers to the capability of adversarial examples to evade machine detections. Such detections utilize some properties of adversarial examples different from clean images, such as network uncertainty [50] and interpretability [174].

Next, according to the different imperceptibility problems in three types of adversarial examples, we propose new approaches to enhance the imperceptibility performance in input-specific, input-agnostic and input-free adversarial attacks, respectively.

- **Input-specific Black-box Saliency Attack.** In Chapter 3, we focus on the black-box input-specific adversarial attacks, where the attacker only has access to the model output. In light of the poor imperceptibility of existing black-box attack methods with global perturbation and high-frequency noises, we propose Saliency Attack to restrict the perturbations to a small but classifier-sensitive region, and further refine the perturbations to generate efficient and imperceptible input-specific adversarial examples in the black-box setting. The generated perturbations are more imperceptible and efficient than those generated by existing black-box attack methods.

- **Input-agnostic Joint Universal Adversarial Perturbations.** In Chapter 4, motivated by the interpretation discrepancy between the clean images and their adversarial examples that can be used to detect adversarial examples, we propose Joint Universal Adversarial Perturbations (JUAP) to generate imperceptible universal perturbations by jointly optimizing the adversarial loss and interpretation discrepancy. The generated universal perturbations can be applied to any clean images to fool target models while maintaining both human and machine imperceptibility.

- **Input-free Diffusion-based SemDiff Attack.** In Chapter 5, we explore the most general input-free adversarial examples, which are generated without any input clean images. Current works use generative models to generate input-free adversarial examples, but are still limited in naturalness with perceptible artifacts. In order to mitigate or even address the issue, we propose SemDiff to utilize the semantic latent space of diffusion models to change the output image along the direction of certain semantically meaningful attributes, such as *smiling* for gender classification task. We additionally devise a multi-attributes optimization approach to ensure attack success while maintaining the naturalness and imperceptibility of the generated adversarial examples. The generated input-free adversarial examples are natural and exhibit semantically meaningful

changes, in accord with the attributes' weights.

Finally, Chapter 6 summaries the conclusion of this thesis, propose a few perspectives, and discuss several unresolved challenges for future research.

Chapter 2

Background

In this section, we first introduce the definition of adversarial examples and propose a new taxonomy to classify all adversarial examples according to their definitions. Then, we review the related work on the imperceptibility of adversarial examples from two perspectives: human imperceptibility and machine imperceptibility. Finally, we introduce different attack settings and the evaluation metrics of adversarial attacks.

2.1 Adversarial Examples: Definition and Taxonomy

In this section, we discuss the definition of adversarial examples and propose a new taxonomy to classify all adversarial examples according to their different mathematical definitions. Next, adversarial attacks based on the taxonomy are reviewed.

In 2013, Szegedy et al. [173] originally defined an adversarial perturbation as "imperceptibly small perturbations to a correctly classified input image, so that it is no longer classified correctly". According to this definition, an adversarial example is supposed to appear indistinguishable from the original image to human eyes, and is

able to cause the misclassification of the target model. However, as more different types of adversarial samples are proposed, this definition is not sufficient to cover all adversarial examples. For example, Xiao et al. [198] apply spatial transformations to clean images to deceive target models. There is an obvious visual difference between the original image and the adversarial example, which does not satisfy the original definition. Moreover, Song et al. [166] propose unrestricted adversarial examples that can be generated by generative models without given input images, which means there is even no original image to compare the difference.

In a more general sense, adversarial example can be any natural-looking image that can deceive the target model, as long as the semantics of the object in the image keep invariant to human eyes [166, 99]. The mathematical definition of adversarial examples is expressed as 1.1. To cover both indistinguishable adversarial examples and natural-looking adversarial examples, we classify adversarial examples into three categories based on their mathematical definitions: input-specific adversarial examples, input-agnostic adversarial examples and input-free adversarial examples. Assume $\mathcal{O}$ is the set of all real-world images that look realistic to humans, $\boldsymbol{o}$ represents an oracle model that provides ground-truth results. Given the target model $\boldsymbol{f}$ to be attacked and a selected test dataset $\mathcal{T} \subset \mathcal{O}$, the definitions of different types of adversarial examples and their mathematical formulations are as follows:

- **Input-specific Adversarial Examples**: Given one test image x' from the test set $\mathcal{T}$, an adversarial example x is generated that can look realistic to humans with invariant semantics and can deceive the target model $\boldsymbol{f}$.

$$\mathcal{A}_{is} \triangleq \{x \in \mathcal{O} \mid \exists x' \in \mathcal{T}, f(x') = o(x') = o(x) \neq f(x)\}. \tag{2.1}$$

  - **Input-specific Indistinguishable Adversarial Examples**: While satisfying the definition of input-specific adversarial examples, the adversarial example x is indistinguishable from the test image x' with certain distance

metric (e.g., L_0, L_2 and L_∞) constraint controlled by threshold ϵ.

$$\mathcal{A}_{isi} \triangleq \{x \in \mathcal{O} \mid \exists x' \in \mathcal{T}, d(x, x') \leq \epsilon \wedge f(x') = o(x') = o(x) \neq f(x)\}. \tag{2.2}$$

- **Input-agnostic Adversarial Examples**: Given any test image x' from the test set $\mathcal{T}$, an adversarial example x is generated with a universal perturbation or transformation that can look realistic to humans with invariant semantics and can deceive the target model $\boldsymbol{f}$.

$$\mathcal{A}_{ia} \triangleq \{x \in \mathcal{O} \mid \forall x' \in \mathcal{T}, f(x') = o(x') = o(x) \neq f(x)\}. \tag{2.3}$$

  - **Input-agnostic Indistinguishable Adversarial Examples**: While satisfying the definition of input-agnostic adversarial examples, the adversarial example x is indistinguishable from the test image x' with certain distance metric (e.g., L_0, L_2 and L_∞) constraint controlled by threshold ϵ.

$$\mathcal{A}_{iai} \triangleq \{x \in \mathcal{O} \mid \forall x' \in \mathcal{T}, d(x, x') \leq \epsilon \wedge f(x') = o(x') = o(x) \neq f(x)\}. \tag{2.4}$$

- **Input-free Adversarial Examples**: Given no test image, an adversarial example x is generated from scratch that can look realistic to humans and can deceive the target model $\boldsymbol{f}$.

$$\mathcal{A}_{if} \triangleq \{x \in \mathcal{O} \mid o(x) \neq f(x)\}. \tag{2.5}$$

Among them, input-specific adversarial examples have a subset, input-specific indistinguishable adversarial examples, which possess an additional constraint of indistinguishability from the test image in certain distance metric. The same applies to input-agnostic adversarial examples and input-agnostic indistinguishable adversarial examples. To better clarify the relationship among different types of adversarial examples, we present a Venn diagram in Fig. 2.1.

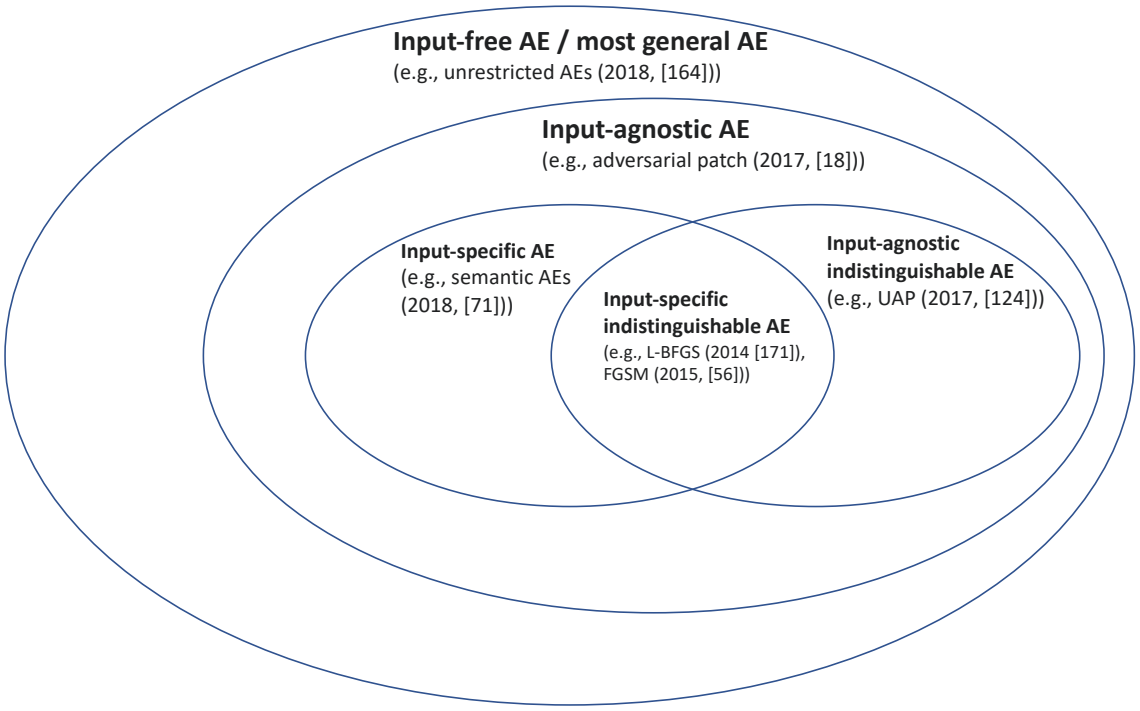


Figure 2.1: **Taxonomy of Adversarial Examples.**

It is shown that the set of input-specific indistinguishable adversarial examples, as the intersection of the set of input-specific adversarial examples and the set of input-agnostic indistinguishable adversarial examples, only occupy a small part of the set of all adversarial examples. Most early works on adversarial examples belong to this category, such as L-BFGS attack [173] and FGSM [56]. The set of input-agnostic adversarial examples contains the set of input-specific adversarial examples and the set of input-agnostic indistinguishable adversarial examples. The set of input-free adversarial examples is the most general one, containing all other types of adversarial examples. Theoretically, an input-agnostic adversarial attack can also generate input-specific adversarial examples, while an input-free adversarial attack can generate both input-specific and input-agnostic adversarial examples. Hence, there is much room to explore for input-free adversarial examples, which is also the most challenging one to implement theoretically.

2.1.1 Input-specific Adversarial Examples

Input-specific Adversarial Example, also known as Image-Dependent Adversarial Example or Instance-agnostic Adversarial example, refers to the adversarial example that is generated based on a specific clean image. More specifically, most of the Input-specific Adversarial Attacks pursue the indistinguishability of the adversarial example from the original image. Hence, we term this type of attack as **Input-specific Indistinguishable Adversarial Attack**. Most of the existing adversarial attacks belong to this category, they try to find a small and imperceptible adversarial perturbation that can be added to the clean image to cause wrong predictions of the target model. Input-specific Indistinguishable Adversarial attacks can be further divided into two groups: Target Distortion Attacks and Target Success Attacks [213].

- **Target Success Attacks**: The goal of this type of attack is to maximize the attack success rate given a constraint on the perturbation magnitude. Given a well-trained DNN classifier $h : [0,1]^n \rightarrow \mathbb{R}^K$, where n is the dimension of the input x, and K is the number of classes. Let $h_k(x)$ denote the predicted score that x belongs to class k. The classifier assigns the class that maximizes $h_k(x)$ to the input x. Target Success Attacks aim to find an adversarial perturbation δ such that:

$$\arg\max_{\delta} \; h_k(x+\delta) \neq y_{gt}, \quad s.t.\; d(x, x+\delta) \leq \epsilon \text{ and } x+\delta \in [0,1]^n, \tag{2.6}$$

  where y_{gt} is the ground-truth label of the input x, $d(x, x+\delta)$ measures the difference between the perturbed and original image, with a threshold ϵ below which the perturbation is considered imperceptible. Meanwhile, the second constraint indicates that the adversarial example $x+\delta$ must be a valid image.

  The first Target Success Attack is the Fast Gradient Sign Method (FGSM), which is a simple one-step attack algorithm that computes the perturbation by maximizing the loss function of the target model with respect to the input

image with a L_∞ norm constraint [56]. Iterative-FGSM (I-FGSM) performs FGSM iteratively to generate a stronger adversarial perturbation [89]. At each iteration, the perturbation is projected back to the L_∞ ball around the original image. Another widely used Target Success Attack is the Projected Gradient Descent (PGD) attack, which is also an iteration method. The difference between PGD and I-FGSM lies in that PGD attack uses L_2 norm constraint and initializes the perturbation with random noise while I-FGSM just initializes the perturbation with zero values. Furthermore, Momentum Iterative FGSM (M-IFGSM) [42] incorporates momentum into the gradient iteration process to improve the attack performance.

- **Target Distortion Attacks**: The goal of this type of attack is to minimize the perturbation magnitude between the original image and the adversarial example, while ensuring that the target model misclassifies the adversarial example. Target Distortion Attacks can be formulated as the following optimization problem:

$$\arg\min_{\delta} \; d(x, x+\delta), \quad s.t.\; h_k(x+\delta) \neq y_{gt} \text{ and } x+\delta \in [0,1]^n. \tag{2.7}$$

Szegedy et al. [173] proposed the first Target Distortion Attack by minimizing the perturbation with the adversarial loss together:

$$\arg\min_{\delta} \; \mathcal{L}(x+\delta, y_{gt}) + ||\lambda||_2, \tag{2.8}$$

where λ controls the trade-off between the adversarial loss and the perturbation magnitude. The unconstrained optimization problem is solved by Limited-memory Broyden-Fletcher-Goldfarb-Channo (L-BFGS) algorithm.

Later, Carlini and Wagner investigate multiple loss functions and find one performs best [20]:

$$\mathcal{L}(x+\delta, y_{gt}) = -\max(Z(x+\delta)_{y_{gt}} - \max_{i \neq y_{gt}}(Z(x+\delta)_i),\; -\kappa) \tag{2.9}$$

where $Z(x+\delta)_{y_{gt}}$ is the logit of the adversarial example with respect to the ground-truth class of the original image. In this way, the adversarial example tends to be misclssified into the class i, with a margin κ of the loss function between the ground-truth class and other classes.

Instead of optimizing the perturbation and adversarial loss together, some works [151, 215] decouple the two objectives and optimize them step by step. Specifically, Rony et al. [151] propose Decoupling Direction and Norm (DDN) attack, that increases or decreases the perturbation norm according to whether the perturbed image is adversarial. Zhang et al. [215] propose boundary projection(BP) attack, which maximizes the adversarial loss when the perturbed image is not adversarial, and then walk on the manifold of the class boundary to minimize the perturbation.

The above attacks all focus on the visual indistinguishability of the adversarial examples from the original images. Beyond indistinguishability, some **Input-specific Adversarial Attack** works believe that a natural-looking adversarial example is not necessarily indistinguishable from the original image, as long as the semantics of the object in the image keep invariant to human eyes [71, 198, 59].

For example, Hosseini et al. [71] propose a Semantic Adversarial Examples by perturbing original images in HSV (Hue, Saturation, Value) color space. The generated adversarial examples visually differ from the original images with different hue and saturation components, but they are still semantically similar to the original images. Xiao et al. [198] propose to apply spatial transformations to original images to deceive target models. The transformed images are visually different from the original images, but they still look natural with invariant semantics. We discuss more details of these attacks in Section 2.2.1.

2.1.2 Input-agnostic Adversarial Examples

Input-agnostic Adversarial Example, also known as Image-agnostic Adversarial Example or Instance-agnostic Adversarial example, refers to a universal perturbation or transformation that can be applied to any clean image to generate an adversarial example. Similarly, some works pursue the indistinguishability of the adversarial example from the original image, namely **Input-agnostic Indistinguishable Adversarial Attack**. The first Input-agnostic Indistinguishable Adversarial Attack is the Universal Adversarial Perturbation (UAP) attack [126], which aims to find a single universal perturbation such that most perturbed images can fool the target model:

$$h_k(x+\delta) \neq h_k(x), \text{ for "}most\text{" } x \in \mathcal{T}, \quad s.t.\ d(x, x+\delta) \leq \epsilon, \tag{2.10}$$

where $\mathcal{T}$ denotes a distribution of clean images. The proposed attack algorithm accumulates the UAP δ by iteratively crafting input-specific perturbations for the clean images. Specifically, if the accumulated UAP does not push the current clean image across the decision boundary, the minimal perturbation $\Delta\delta$ is computed to push the image across the decision boundary. After each iteration update, the perturbation $\delta + \Delta\delta$ is projected on the L_2 ball of radius ϵ. Subsequent works also use trained Generative Adversarial Networks (GANs) to generate UAPs [142, 60].

The adversarial examples with the UAP above generally look indistinguishable from the original images. On the other hand, some works [18, 130, 159, 98, 203] explore **Input-agnostic Adversarial Attacks** in a variety of ways beyond indistinguishability. Adversarial patch [18] is an optimized image patch that can be placed on a local region of clean images to degrade the performance of target models. Compared with global UAPs, adversarial patches are local perturbations but are more salient since there is no constraint on the perturbation magnitude. In addition, Naderi et al. [130] learn a universal spatial transformation with a limited number of parameters, which can be applied to any clean image to generate adversarial examples. Further-

more, physical attacks elaborate some adversarial mediums, such as stickers [98], clothes [203] and eyeglasses [159] to be replaced in physical space to mislead image recognition systems.

2.1.3 Input-free Adversarial Examples

Input-free Adversarial Example, also known as Unrestricted Adversarial Examples, refers to the adversarial example that is generated from scratch without given any clean image. **Input-free Adversarial Attack** gets rid of the dependence of input images, as long as the generated adversarial examples can look realistic to humans and deceive target models. Song et al. [166] propose the first Input-free Adversarial Attack, and use well-trained Generative Adversarial Networks (GANs) to generate adversarial examples by optimizing the input noise vector. In addition, Khoshpasand and Ghorbani [85] trained a generator for each class to improve the transferability of adversarial examples. Zhang et al. [216] designed a novel augmented triple-GAN and train a MLP to produce the noise vector instead of optimizing it directly. Another line of GAN-based Input-free Adversarial Attacks is end-to-end method, which trains generators to directly output realistic adversarial examples that can deceive the target model [185, 196] without the optimization of noise vector.

Inspired by recent advances of diffusion models in image synthesis, some recent works turn to utilize diffusion models to generate input-free adversarial examples [33, 24]. Chen et al. [24] first proposed AdvDiffuser with a well-trained conditional DDPM [37]. Starting from an initial random noise x_T, AdvDiffuser performs PGD to add perturbation into each intermediate latent noise in the reverse process, and obtains the final output x_0 as the adversarial example. Dai et al. [33] proposed AdvDiff and added another noise sampling guidance to update the initial noise x_T in each iteration, achieving a higher attack success rate.

2.2 Imperceptibility of Adversarial Examples

The imperceptibility of AEs is essential for practical attackers. However, how to define and evaluate imperceptibility is still an open question. In this section, we study the imperceptibility of adversarial examples from two perspectives: human imperceptibility and machine imperceptibility. Human imperceptibility focuses on the human capability to recognize adversarial examples from real-world images. Based on our discussion above, human imperceptibility can be further divided into "indistinguishable" adversarial examples and "natural-looking" adversarial examples." On the other hand, machine imperceptibility refers to the capability of adversarial examples to evade machine detections. We investigate the machine imperceptibility from the perspective of model uncertainty, model interpretability and data distribution. The taxonomy of imperceptibility is shown in Fig. 2.2.

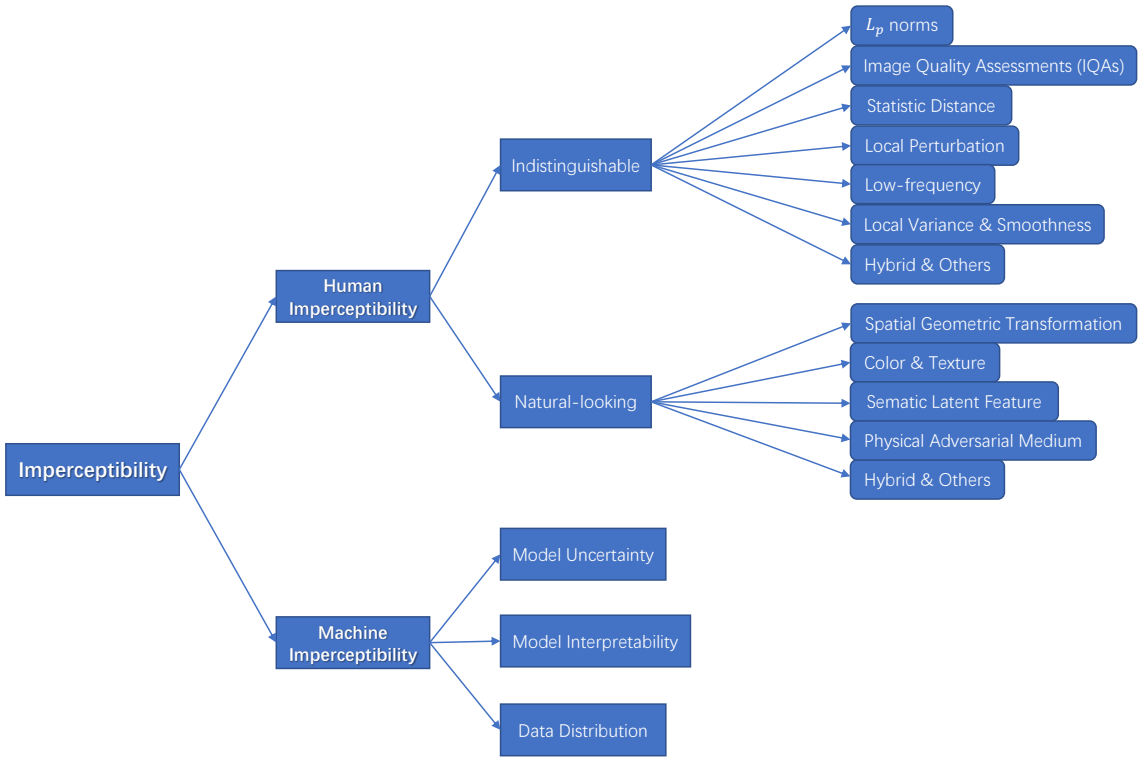


Figure 2.2: **Taxonomy of imperceptibility of adversarial examples.**

2.2.1 Human Imperceptibility

Indistinguishable Adversarial Examples pursues the adversarial examples that look indistinguishable from the original images to human eyes. The core of indistinguishable adversarial examples is to make the perturbation imperceptible and stealthy.

- L_p **Norms**: Most existing works use L_p norms, such as L_0, L_2 and L_∞ to quantify the magnitude of the perturbation added to the original images. Specifically, L_0 norm counts the number of pixels that are changed[140, 169], L_2 norm computes the Euclidean distance between the original image and the perturbed image[173, 20, 127], and L_∞ norm measures the maximum pixel change [56, 89, 42]. However, despite the popularity of L_p norms, they are found not suitable enough for simulating the human vision system [158, 51, 210]. Therefore, researchers explore other metrics to replace L_p norms to measure the imperceptibility of adversarial examples.

- **Image Quality Assessments**: In addition to L_p norms, some works leverage full-reference image quality assessment (IQA) metrics to evaluate the difference between the adversarial example and the original image. For example, Li et al. [101] use the structural similarity (SSIM) [187] and Learned Perceptual Image Patch Similarity (LPIPS) [90] to measure the perceptual similarity and add them into the optimization objective to improve the imperceptibility of adversarial examples. A recent study in [51] systematically examines various IQA metrics including L_p norms through large-scale human evaluation on the imperceptibility of different AEs. Among all the metrics, they find the most apparent distortion (MAD) [92] metric is closest to subjective scores.

- **Statistic Distance**: Different from L_p norms that focus on the change of each pixel value, some works [192, 178] regard the pixel values of an image as a

probability distribution, and use statistical distance, such as Wasserstein distance [191] to calculate the distance between the distributions of adversarial examples and clean images. Compared with L_p norm based perturbation that tends to change the background pixels, Wasserstein distance based perturbation has a structure resembling the actual content of the image, thus is more imperceptible [192].

- **Local Perturbation**: Most adversarial attacks generate global perturbations that are added to the whole image, resulting in obvious artifacts especially in the background [34]. Hence, a strict idea is to restrict the perturbation to a small but classifier-sensitive region. To extract such a region, Xiang et al. [197] apply Grad-CAM [156] to compute the heat map of a reference model with respect to the input image, and construct a binary mask with important pixels. Dai et al. [34] use the Pyramid Feature Attention (PFA) network [222] to extract the salient object in an image, and then binarize the saliency map to obtain the mask.

- **Low-frequency**: For an image, its low-frequency components represent the global structure, while high-frequency components represent the local details. Some works [58] believe that low-frequency perturbations are imperceptible, and thus adding perturbations in the low-frequency subspaces of an image. For example, Guo et al. [58] utilize the discrete cosine transform (DCT) [2] to transform an image from spatial domain to frequency domain, and decompose it into different cosine wave components. Chen et al. [119] instead use the discrete wavelet transform (DWT) [63] to decompose an image into different frequency components.

- **Local Variance & Smoothness**: It is found that human eyes are more sensitive to the perturbations on pixels in low variance regions [106], such as sky and grass in the background. Luo et al. [118] believe an attack should per-

turb the pixels in high variance regions rather than low variance regions. They measure the local variance of a pixel by computing the standard deviation of $n \times n$ neighbor pixels' values. Moreover, Zhang et al. [214] propose smooth adversarial examples and integrate Laplacian smoothing [227] into the attack.

- **Hybrid & Others**: There are also some hybrid and alternative methods that combine different metrics or imperceptibility concepts for better performance. For example, Croce and Hein [30] use a combination of L_0 and L_∞ norms to generate sparse and imperceptible perturbations. Liu et al. [108] consider four factors affecting the imperceptibility of human eyes, including
  - Just noticeable distortion (JND) [208], quantifies the sensitivity of the human vision system to different background luminance.
  - Weber-Fechner law [35], reveals an important principle in psychophysics about the logarithmic mapping between the magnitude of a physical stimulus and its perceived intensity.
  - Texture masking [118], indicates that human eyes are more sensitive to the perturbations on pixels in smooth regions than those in textured regions
  - Channel modulation [154], points out that human vision has different sensitivity to the perturbations in different color channels.

Natural-looking Adversarial Examples are not necessarily indistinguishable from the original images, as long as the semantics of the object in the image keep invariant to human eyes [166, 99].

- **Spatial Geometric Transformation**: Some works apply spatial geometric transformations, such as translation, rotation [198, 46] and optical flow [184], to original images to generate adversarial examples. Although there is a shift in the pixel values, such geometric changes are believed locally smooth, leading to high perceptual quality [198]. Furthermore, Naderi et al. [130] learn a limited number

of parameters for a universal spatial transformation, which can be applied to any clean image to generate adversarial examples.

- **Color & Texture**: Different from most attacks that focus on RGB images, some studies turn to different color spaces and textures. For example, Hosseini et al. [71] transforms a RGB image into HSV (Hue, Saturation, Value) color space to change the hue and saturation components while keeping the value component unchanged. Shamsabadi et al. [157] instead transforms a RGB image into Lab color space [152] to change the de-correlated a and b channels without changing the lightness channel. Zhao et al. [224] also consider a perceptual color distance, CIDE2000 [120], and add it into the optimization objective as the distance term. In addition, Bhattad et al. [10] utilize a colorization model a CNN-based texture transfer to control the color and texture of images.

- **Semantic Latent Feature**: Some researchers choose to perturb in the latent space of generative models, where disentangled high-level semantic features can be used to manipulate image content while preserving the naturalness of images. For instance, Na et al. [129] utilize StyleGAN2 [82] to extract highly disentangled latent representations of images, and then perturb in the latent space to find adversarial examples. Moreover, Mohaghegh et al. [124] train Normalizing Flows (NFs) [146] to mimic the distribution of real-world images, and search in its latent space.

- **Physical Adversarial Medium**: In addition to the manipulation in digital images, physical attacks elaborate various adversarial mediums, such as stickers [98], clothes [203] and eyeglasses [159] to be replaced in physical space. Such physical adversarial mediums are things in our daily life, but are very sensitive and adversarial to image recognition systems.

- **Hybrid & Others**: There are also some hybrid and alternative methods that can generate natural-looking adversarial examples. For example, Laidlaw et

al. [91] propose the Functional adversarial examples, which combine CIELUV color space with constraints on local variance for better imperceptibility. Furthermore, Hendrycks et al. [65] propose a ImageNet-C benchmark consisting of 15 diverse corruption types, including four main categories: noise, blur, weather, and digital. Such corruptions simulate various phenomena in the real world, such as fog, snow, and motion blur.

2.2.2 Machine Imperceptibility

Machine imperceptibility refers to the capability of adversarial examples to evade machine detections. We investigate the machine imperceptibility from the perspective of model uncertainty, model interpretability and data distribution.

- **Model Uncertainty**: This type of work detects adversarial examples by measuring the uncertainty of model predictions on the given input. The hope is that a natural image will have a same and correct prediction regardless of the randomness [19]. In contrast, the prediction of an adversarial example often varies because adversarial examples are usually the examples that just cross the decision boundary of classifiers. The randomness can be added into the model itself or the model input. For example, Feinman et al. [50] measure the uncertainty of model on a given input with the Dropout [167] technique. Xu et al. [205] propose Feature Squeezing, a preprocessing-based detection method that evaluates the model on both the original input and the input after being pre-processed by various feature squeezers, including squeezing color bits, local smoothing and non-local smoothing.

- **Model Interpretability**: Various interpretability methods have been proposed to explain the decision-making process of models [161, 226, 156]. Among them, attribution-based interpretation methods usually produce a saliency map to evaluate the sensitivity of network classification decisions with respect to the

pixels in an input image [219]. Previous works found the phenomenon of interpretation discrepancy that an imperceptible adversarial example to deceive classifiers can lead to a significant change in a class-specific interpretation saliency map [202, 12]. Hence, some adversarial defense methods are proposed to utilize the interpretation discrepancy to detect adversarial examples [174, 183, 29]. For example, Wang et al. [183] use multiple interpreters obtain different saliency maps given an input image, and then train CNN-based classifiers to recognize whether the input image is adversarial or not.

- **Data Distribution**: Some researchers believe that the distribution of adversarial examples are different from that of natural images, and thus devising detectors based on the distribution difference [57, 66]. Specifically, Grosse et al. [57] propose to retrain the classifier with an addictive $N + 1$st class soleLy for adversarial examples on both clean and adversarial examples. Mehdi et al. [14] further propose a CNN-based binary classifier to distinguish adversarial examples from clean ones. In addition, Hendrycks & Gimpel [66] use Principal Component Analysis (PCA) to reduce the dimensionality of input images, and then train a classifier to detect adversarial examples in a low-dimensional space.

2.3 Adversarial Attacks

This section presents a quick overview of the settings and evaluation methods of adversarial attacks.

2.3.1 Settings: White, grey and black box

According to the accessibility of the target model to attackers, adversarial attack can be divided into three categories: white-box, grey-box and black-box. In the **white-box** attack setting, attackers have full access to the target model, including

the model architecture, parameters and training set. Hence, white-box attacks can be easily implemented by computing the gradients of the adversarial loss or model output with respect to the input image [173, 56, 20, 127].

The **black-box** attack setting is much stricter than the white-box attack setting. Attackers have no knowledge about the target model, but they can query the target model with input images and obtain the outputs. Black-box attacks can be further categorized as score-based black-box attacks and decision-based black-box attacks according to the type of output information available to attackers. In the **score-based black-box** attack setting, attackers can obtain the output scores (e.g. class probabilities or logits) of the target model for each class. With such score vectors, score-based black-box attacks can either numerically estimate the gradients of the target model by sampling around the image [23, 175, 75, 76] and then apply white-box attack methods, or use gradient-free optimization methods to directly optimize perturbations [131, 125, 5, 4]. By contrast, in the **decision-based black-box** attack setting, attackers have less information, i.e., only the predicted label of the target model, which is more practical in real-world scenarios. This means if we start from adding perturbations around the original image, probability we cannot get effective direction information from the output label. Therefore, decision-based attacks usually start from an image of the target class, and then gradually approach the original image by performing rejection sampling [16, 26, 25, 22]. The difference between black-box and white-box attack settings is depicted in Fig. 2.3.

The **grey-box** attack setting is somewhere between the white-box and black-box attack setting. The most common grey-box attack setting is the transfer-based attack [139, 113, 200, 100], where attackers do not known the target model information but have access to the training data. Hence, attackers can train a substitute model (or use a on-the-shelf model) to approximate the target model's decision boundary Based on the transparent substitute model, white-box attacks can be used to generate adversarial examples, and transfer them to attack the target model [139].

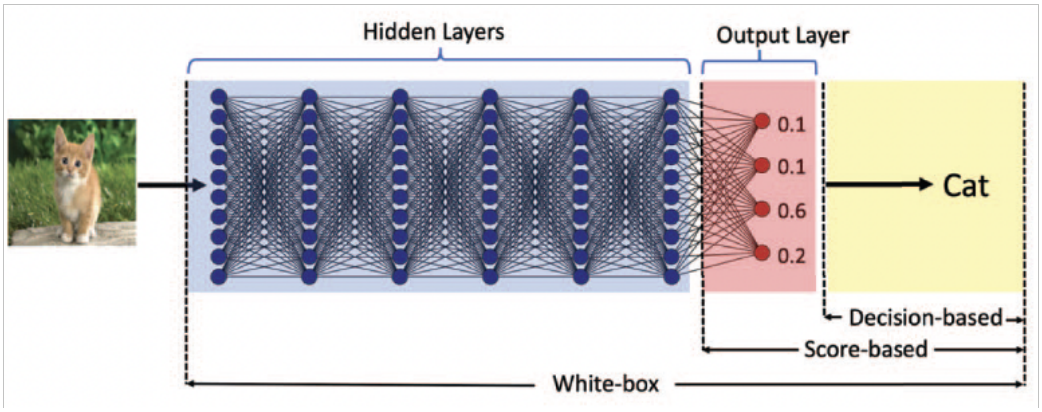


Figure 2.3: **An illustration of accessible components of the target model for different attack settings.** The image is sourced from the paper of HopSkipJumpAttack [22].

2.3.2 Evaluation Methods

As we discussed above, attack successfulness and imperceptibility are the two objectives of adversarial attacks. Thus, the evaluation methods of adversarial attacks can be divided into two groups accordingly:

- **Evaluate Attack Successfulness**:
  - **Attack Success Rate (ASR)**: The most common metric to evaluate the attack successfulness is the attack success rate (ASR), which is defined as the ratio of the number of adversarial examples that successfully fool the target model to the total number of test images.
  - **Transferability**: To evaluate the generalization of adversarial examples, especially for transfer-based attacks, transferability is used to measure the ASR of the adversarial examples generated on model f_1 against model f_2.
- **Evaluate Imperceptibility**:
  - **Visualization**: A simple way to evaluate the imperceptibility of adversarial examples is to provide visual examples for readers to check the in-

distinguishability or naturalness of the generated adversarial examples. Additionally, some works [100, 185] visualize the difference between the distribution of adversarial examples and that of natural images using t-SNE [177].

- **Quantitative Metrics**: Various quantitative metrics are common used to evaluate the imperceptibility, such as L_p norms, image quality assessments like SSIM [187] and MAD [92], and some metrics used in image synthesis like FID [68] and Inception score [153].
- **Human Evaluation**: Human evaluation is a more reliable way to evaluate the imperceptibility of adversarial examples. Some works [51, 101] employ humans or use crowd-sourcing platforms like Amazon Mechanical Turk (MTurk) to judge the adversarial examples compared with baseline images and clean images.

2.4 Discussions on Adversarial Examples

2.4.1 Connections with Digital Watermarking

Similar to adversarial examples, digital watermarking techniques also operate by embedding specific perturbations into an image. However, their fundamental objectives diverge significantly. The primary goal of watermarking is information embedding and verification, such as copyright protection or source tracing for AI-generated content [188]. It emphasizes robustness against common image processing or removal attacks. In contrast, adversarial examples aim at model deception and evasion, focusing on the attack effectiveness and imperceptibility of perturbations to mislead DNNs. While watermarking serves as an object to be protected (a defense), adversarial examples typically function as an offensive tool (an attack).

Despite differing purposes, the two fields are tightly connected technically, primarily

manifesting in the following aspects:

- **Watermarking Techniques Can Be Utilized for Attacks.** The methodology of embedding watermark signals can be repurposed to construct novel attack perturbations. For instance, Zhang et al. [228] proposed a framework that integrates a personal watermark directly into the adversarial example generation process. When such an adversarial example is used to mimic style or content by a diffusion model, it forces the generated output to retain the attacker's visible watermark, thereby claiming ownership and preventing unauthorized imitation. This approach transforms a traditional defense mechanism (watermarking) into an active attack carrier.

- **Attackers Can Erase Watermarks via Adversarial Attack Methods.** With the rise of watermarking for tracing AI-generated content (AIGC), it has become a target for adversarial removal. Recent research focuses on breaking these "defensive" watermarks. A representative work is UnMarker [83], a universal attack that removes imperceptible watermarks without relying on the detector's feedback. It finds that robust watermark information is often encoded in the amplitude of the frequency spectrum. By employing adversarial optimization to destructively perturb these critical spectral components, UnMarker can effectively erase watermarks, even threatening more advanced "semantic watermarks". This line of work treats the watermark itself as an adversarial target.

- **Adversarial Examples Can Enhance and Protect Watermarks**. Conversely, concepts from adversarial attacks inspire more robust watermarking schemes. One direction is developing "watermark vaccines," [112] where a pre-embedded, imperceptible adversarial perturbation is designed to protect a visible watermark. If an attacker uses a DNN-based model to attempt removal, this perturbation activates, causing the removal to fail or severely degrade image

quality. This work can fool an unauthorized model for privacy protection while allowing authorized recovery of the original image via watermark extraction, balancing attack and restoration.

Building upon the discussion of the connections between adversarial attacks and digital watermarking, it is crucial to clarify the terminology of **real image**, **modified image**, and **generated image** across these two domains. In the context of adversarial attacks:

- A **real image** refers to a genuine image captured from real world without manipulation or perturbation, serving as the original input or baseline for an attack.
- A **modified image** typically denotes an adversarial example created by perturbing a real image with an additive perturbation or transformation to deceive DNN models.
- A **generated image** also refers to an adversarial example in this thesis, but specifically one produced from scratch without any real image input, often via generative models.

While in the context of digital watermarking, the focus and meaning of these terms shift:

- A **real image** (or more precisely, a host image) is the original, copyright-protected content before any watermark is embedded.
- A **modified image** refers to the watermarked image, namely the output after a watermark (visible or invisible) has been embedded into the host image.
- A **generated image**, particularly in the era of AIGC, is the AI-synthesized output. Here, watermarking techniques are applied to the generated image to label its AI origin or ownership.

In summary, while both fields operate on images through modification, the adversarial domain uses modification offensively to deceive model perception, whereas the watermarking domain uses it defensively to establish and protect identity. The terms "real," "modified," and "generated" images thus take on context-dependent meanings aligned with the goals of each field.

Chapter 3

Saliency Attack: Towards Imperceptible Black-box Adversarial Attack

Deep neural networks are vulnerable to adversarial examples, even in the black-box setting where the attacker is only accessible to the model output. Recent studies have devised effective black-box attacks with high query efficiency. However, such performance is often accompanied by compromises in attack imperceptibility, hindering the practical use of these approaches. In this paper, we propose to restrict the perturbations to a small *salient* region to generate adversarial examples that can hardly be perceived. This approach is readily compatible with many existing black-box attacks and can significantly improve their imperceptibility with little degradation in attack success rate. Further, we propose the Saliency Attack, a new black-box attack aiming to refine the perturbations in the salient region to achieve even better imperceptibility. Extensive experiments show that compared to the state-of-the-art black-box attacks, our approach achieves much better imperceptibility scores, including most apparent distortion (MAD), L_0 and L_2 distances, and also obtains significantly better true

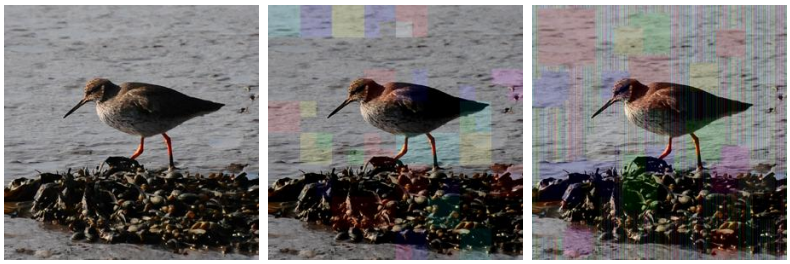
success rate and effective query number judged by a human-like threshold on MAD. Importantly, the perturbations generated by our approach are interpretable to some extent. Finally, it is also demonstrated to be robust to different detection-based defenses.

3.1 Introduction

Deep neural networks (DNNs) have achieved significant progress in wide applications, such as image classification [36], face recognition [141], object detection [145], speech recognition [69] and machine translation [6]. Despite their success, deep learning models have exhibited vulnerability to adversarial attacks [173] [56] [109] [111]. Crafted by adding some small perturbations to benign inputs, adversarial examples (AEs) can fool DNNs into making wrong predictions, which is a critical threat especially for some security-sensitive scenarios such as autonomous driving [171].

Existing adversarial attacks can be divided into *white-box* and *black-box* attacks according to the accessibility to the target model. White-box attacks [173] [56] [140] [20] have full access to the architecture and parameters of the target model, and can generate successful AEs easily via backpropagation. However, in practice, the model internals are often unavailable to attackers. This gives rise to the more realistic and challenging black-box attacks [131] [75] [76] [125] [5] [103] that only require the output of the target model.

Motivated by the fact that many real-world online application programming interfaces (APIs) often pose mandatory time or monetary limits to user queries [75], most recent research on black-box attacks concerns improving query efficiency, which indeed has achieved notable progress. For example, the state-of-the-art (SOTA) Square Attack [5] can succeed in untargeted attack on ImageNet dataset [36] with only tens of queries on average. On the other hand, such performance is often achieved by



(a) Original image (b) Parsimonious Attack (c) Square Attack

Figure 3.1: Some AEs generated by the SOTA black-box attacks.

applying large-region or even global perturbations (e.g., random vertical stripes in Square Attack) to the original input, eventually resulting in unnatural AEs with significant visual differences from the original (see Figure 3.1 for some examples). In effect, imperceptibility is essential for attackers. Current online APIs usually integrate detectors into their services to detect anomalous inputs [103]. Those AEs with too visible perturbation are difficult to pass these detectors, not to mention human judgment, hindering the practical use of these black-box attacks.

This paper aims to address the above issue. The key insight is that we can lead black-box attacks to search in a small *salient* region that has the most significant influences on the model output, to achieve successful attacks with imperceptible perturbations. Concretely, it has been shown in previous DNNs visualization work [161] that one can generate the so-called BP-saliency map to illustrate which features can influence the model prediction by calculating derivatives of the model output with respect to the input. As shown in the first column of Figure 3.2, the brighter pixels in the BP-saliency map have greater impacts on the model output, and therefore perturbing them is more likely to alter the model prediction. Actually, the classical white-box attack JSMA [140] constructs exactly such a map and iteratively selects pixels from it to perturb. Despite the fact that the BP-saliency map cannot be used for black-box attacks due to inaccessibility to model gradients, one can observe from the second column of Figure 3.2 that the region of bright pixels roughly represents the position

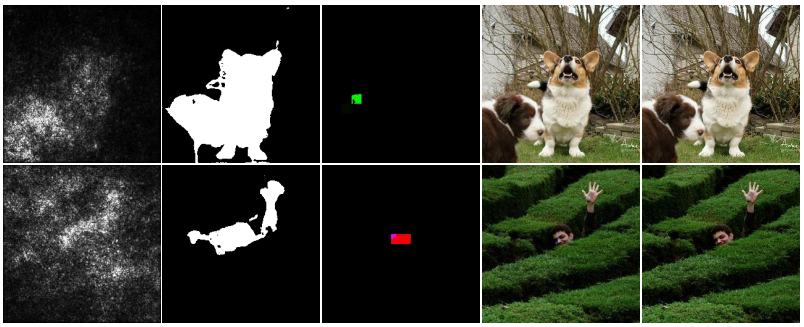


Figure 3.2: An illustration of our Saliency Attack. **1st column:** BP-saliency map, specific to a given image and the corresponding class; **2nd column:** salient region, generated by salient object detection and roughly accords with the region of salient pixels in BP-saliency map; **3rd column:** perturbation, generated by Saliency Attack and restricted in salient regions; **4th column:** final adversarial example; **5th column:** original image.

of the main object in the image. That is to say, in black-box settings, we can leverage the existing salient object detection model [222] that requires no information other than the input image to approximately obtain the salient region, and then restrict the perturbations to it. This approach is appealing because it is readily compatible with most existing black-box attacks. By integrating it into SOTA black-box attacks, we find the modified attacks enjoy significant improvement in attack imperceptibility, with little degradation in success rate (see the results in Tab. 3.1). Nevertheless, they still suffer from the strategy of perturbing globally, eventually generating complex and irregular perturbations that span almost the entire salient region (see Figure 3.6 for some examples).

To achieve more imperceptible attacks, we then seek to further restrict the perturbations to even smaller regions. The intuition is that even in the salient region, some sub-regions are more critical. For example, it has been shown that the dog face region is the brightest in the salient region of a dog image [226], and obscuring it will dramatically change the model output [212]. Furthermore, by combining internal feature

visualization with output prediction, it has been revealed in [136] that within dog face region, the ears and eyes seem to be more important than others. Therefore, it is reasonable to assume the salient region of an image is progressive with respect to its impact on model output. If we could find smaller but more salient sub-region, the perturbations will be more efficient, leading to more imperceptible attacks (see the third column of Fig. 3.2). Thus, we propose the Saliency Attack, a novel black-box attack that recursively refines the perturbations in the salient region.

It is worth mentioning that except white-box attack JSMA, the idea of restricting perturbations to a small region has also been implemented in transfer-based attacks [41] [197], where class activation mapping (CAM) [226] and Grad-CAM [156] are adopted to generate the saliency maps. However, transfer-based attacks assume the data distribution for training the target model is available and thus could build a substitute model to approximate it, which actually belong to the grey-box setting where partial knowledge of the target model is known. As a result, they cannot be applied to the more strict black-box setting where only the model output is available (see Section 3.2.3 for details).

In summary, we make the following contributions in this work.

- To the best of our knowledge, for crafting AEs in the black-box setting, we are the *first* to restrict perturbations to a salient region. This approach is readily compatible with many existing black-box attacks and significantly improves their imperceptibility with little degradation in success rate.

- We propose the Saliency Attack, a novel gradient-free black-box attack that iteratively refines perturbations in salient regions to keep them minimal and essential. Compared with the SOTA black-box attacks, our approach achieves much better attack imperceptibility in terms of most apparent distortion (MAD), L_0 and L_2 distances, and also obtains significantly better success rates and effective query number judged by a human-like threshold on MAD.

- We demonstrate that the perturbations generated by Saliency Attack are more robust against detection-based defenses, including Feature Squeezing and binary classifier detection.

The rest of this paper is organized as follows: Section 3.2 presents a literature review on black-box attacks. Next, in Section 3.3, we define the optimization problem of our attack and detail the proposed Saliency Attack. Section 3.4 shows the experimental results and analysis, followed by a conclusion in Section 3.5.

3.2 Related Work

In this part, we first overview the recent work of black-box attacks. Based on the generation method of AEs, these attacks can be divided into gradient estimation attacks and gradient-free attacks. Besides, we also introduce some work related to the imperceptibility in adversarial attacks. Finally, we discuss some methods that extract salient regions in an image.

3.2.1 Black-box Attacks

Gradient Estimation Attacks

Gradient estimation attacks first estimate the gradients by querying the target model and then use them to run white-box attacks. ZOO attack [23] first adopts symmetric difference quotient to approximate the gradients and then performs white-box Carlini-Wagner (CW) attack [20]. AutoZOOM [175], a variant of ZOO, uses a random vector based gradient estimation to reduce the query number per iteration from $2D$ in ZOO to $N+1$ (D is the dimensionality and N is the sample size). To further enhance query efficiency, Ilyas et al. [76] propose the "tiling trick" that updates a square of pixels simultaneously instead of a single pixel. This dramatically decreases the

dimensionality by a factor of k^2 (k is tile length).

Gradient-free Attacks

Gradient-free attacks do not estimate gradients and directly generate AEs with search heuristics according to the query result. Su et al. [169] propose One-pixel attack that adopts differential evolution algorithm to perturb the most important pixel in the image. Alzantot et al. [4] propose GenAttack, which uses genetic algorithm to generate AEs. To improve query efficiency, Moon et al. [125] consider a discrete surrogate optimization problem that transforms the original constraint of a continuous range $[-\epsilon, +\epsilon]$ to a discrete set $\{-\epsilon, +\epsilon\}$, achieving a massive reduction in the search space. This is motivated by linear program (LP) where the optimal solution is attained at an extreme point of the feasible set [155]. Combing tiling trick [76] and discrete optimization [125], Square Attack [5] has obtained the best result on success rate and query performance so far with random search.

3.2.2 Attack Imperceptibility

The imperceptibility of AEs is essential for practical attackers, which has been investigated by previous studies from different perspectives. Guo et al. [58] consider low frequency perturbations as imperceptible, thus searching for AEs in frequency domain. But Zhang et al. [214] regard imperceptibility as visual smoothness in an image and integrate Laplacian smoothing into optimization. Croce and Hein [30] use a combination of L_0 and L_∞ norms to generate sparse and imperceptible perturbations. Besides, some studies also leverage color distance [225] and image quality assessment (IQA) [186] [101] to improve attack imperceptibility.

On the other hand, determining how to assess the imperceptibility of AEs is still an open question. Most existing adversarial attacks use L_p norms (L_0, L_2 and L_∞) to measure the human perceptual distance between the perturbed image and the origi-

nal one. Nonetheless, it has been shown that L_p norms are not suitable enough for human vision system [158]. A recent study in [51] systematically examines various IQA metrics including L_p norms through large-scale human evaluation on the imperceptibility of different AEs, and finds that among all the metrics the most apparent distortion (MAD) [92] metric is closest to subjective scores. Hence, in this research, we adopt MAD as our main metric to assess attack imperceptibility.

Most Apparent distortion (MAD) metric [92] is one of the state-of-the-art full-reference image quality assessment methods. MAD attempts to merge two separate strategies for two kinds of distorted images, respectively. For high-quality images with near-threshold distortion (just visible), MAD focuses on detection-based strategy to look for distortions in the presence of the image. While for low-quality images with suprathreshold distortion (clearly visible), MAD focuses on appearance-based strategy to look for image content in the presence of the distortions. MAD will control the weight of two strategies according to the type of distorted images. The calculation process of MAD can be summarized as following steps and we recommend interested readers to read the original literature.

1. Compute locations of visible distortions based on luminance images.
2. Combine the visibility map with local error image.
3. Decompose both the distorted and original images into log-Gabor subbands.
4. Calculate different statistics of each subband.
5. Calculate the adaptive blending score.

3.2.3 Extracting Salient Region

As aforementioned, there have been white-box attacks or transfer-based attacks that restrict the perturbations to a small salient region. Specifically, white-box attack

JSMA [140] constructs a BP-saliency map by calculating derivatives of the model output w.r.t input pixels [161], while the two transfer-based attacks [41] [197] utilize CAM and Grad-CAM to extract salient regions, respectively. CAM [226] replaces the final fully connected layers with convolutional layers and global average pooling of a CNN, and localizes class-specific salient regions through forward propagation. Grad-CAM [156] improves CAM with on need to modify the network architecture, but it still requires access to the inner parameters of the model to calculate the gradients. Thus, CAM and Grad-CAM can only be applied to white-box attacks or transfer-based attacks, which first construct a transparent substitute model, craft AEs with gradient-based white-box attacks, and then transfer the generated AEs to the target model. It is conceivable that for transfer-based attacks, the transferability of both AEs and saliency maps highly depends on the similarity between the substitute model and the target model, which further depends on the prior knowledge of the training data distribution. Unfortunately, neither such knowledge nor model gradients are available in black-box settings. Considering the limitations of the above methods, we adopt a salient object detection model [222] to directly generate the saliency map given an input image with no need to access the target model's architecture or parameters.

3.3 Proposed Method

In this section, we first formulate the problem of crafting AEs for image classification models, and then detail our approach. Figure 3.3 illustrates the overall flowchart of Saliency Attack.

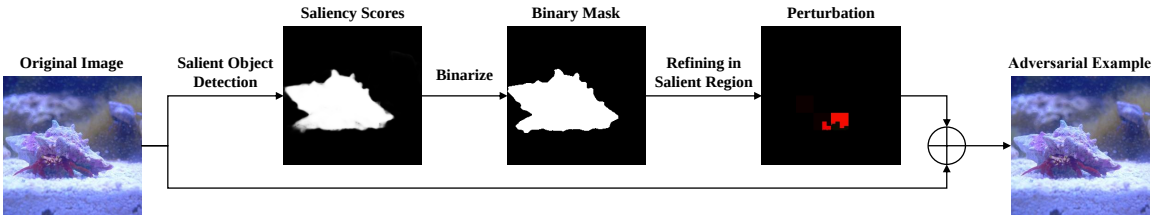


Figure 3.3: The overall flowchart of the proposed Saliency Attack.

3.3.1 Preliminary

Given a well-trained DNN classifier $h : [0,1]^d \to \mathbb{R}^K$, where d is the dimension of the input x, and K is the number of classes. Let $h_k(x)$ denote the predicted score that x belongs to class k. The classifier assigns the class that maximizes $h_k(x)$ to the input x. The goal of an untargeted attack is to find an AE x_{adv}, that results in the model misclassification from the ground-truth class y_{gt}, and meanwhile keeps the distance between x_{adv} and the benign input x smaller than a threshold ϵ:

$$\underset{k=1,\ldots,K}{\arg\max}\ h_k(x_{adv}) \neq y_{gt}, \quad s.t.\ ||x_{adv} - x||_p \leq \epsilon,\ x_{adv} \in [0,1]^d. \tag{3.1}$$

Note the last constraint indicates that x_{adv} is a valid image. In this study, we focus on L_∞ norm, following [125] [5]. Conventionally, the task of finding x_{adv} can be rephrased as solving a constrained continuous problem:

$$\max_{x_{adv} \in [0,1]^d}\ f(x_{adv}), \quad s.t.\ ||x_{adv} - x||_\infty \leq \epsilon. \tag{3.2}$$

where $f(x) = L(x, y_{gt})$ is the loss function. Similar to [125], we transform the continuous problem into a discrete surrogate problem, where the perturbation $x_{adv} - x = \delta \in \{-\epsilon, +\epsilon, 0\}^d$. Note unlike [125], where all pixels are perturbed either by $-\epsilon$ or $+\epsilon$, we allow the pixels to remain unperturbed to refrain from global perturbation. Besides, only the pixels in salient regions can be perturbed. The final problem is defined as the following set maximization problem.

$$\max_{\substack{\mathcal{S}^+, \mathcal{S}^- \subseteq \mathcal{S} \\ \mathcal{S}^+ \cap \mathcal{S}^- = \emptyset}} \left\{ F(\mathcal{S}^+, \mathcal{S}^-) \triangleq f\left(x + \epsilon \sum_{i \in \mathcal{S}^+} e_i - \epsilon \sum_{i \in \mathcal{S}^-} e_i \right) \right\}. \tag{3.3}$$

where $\mathcal{S}$ denotes the set of pixels in salient regions, $\mathcal{S}^+$ and $\mathcal{S}^-$ denote the set of *selected* pixels with $+\epsilon$ and $-\epsilon$ perturbations respectively, and e_i is the i-th standard basis vector. Let $\mathcal{V}$ denotes the ground set which is the set of all pixel locations ($|\mathcal{V}| = d$). Note that $\mathcal{S} \subseteq \mathcal{V}$, and $\mathcal{S} \setminus (\mathcal{S}^+ \cup \mathcal{S}^-)$ is the set of pixels in salient regions that are unperturbed. The goal of Eq. (3.3) is to find $\mathcal{S}^+$ and $\mathcal{S}^-$ that will maximize the objective set function F.

3.3.2 Salient Object Detection

Salient object detection aims to automatically and accurately extract salient object(s) in an image. Compared with other salient region extraction methods like BP-saliency map [161], CAM [226] and Grad-CAM [156], this type of models do not require any information other than the input image, which is very suitable for the black-box setting. We adopt the Pyramid Feature Attention (PFA) network[1] [222] that achieves the SOTA performance on multiple datasets via capturing high-level context features and low-level spatial structural features simultaneously. Pyramid Feature Attention (PFA) network [222] is a novel salient object detection method considering the different characteristics level features. Specifically, the saliency maps from low-level features contain many noises, while the saliency maps from high-level features only get an approximate area. Therefore, PFA network first devises a context-aware pyramid feature extraction (CPFE) module to get multi-scale multi-receptive-field high-level features, and then uses channel-wise attention (CA) to select appropriate scale and receptive-field for generating saliency regions. On the other hand, to refine the boundaries of saliency regions, PFA network uses spatial attention to better focus on the effective low-level features, and obtain clear saliency boundaries. After the processing of different attention mechanisms, the high-level features and low-level features are complementary-aware and suitable to generate saliency map. Besides,

[1]Implementation of PFA: https://github.com/sairajk/PyTorch-Pyramid-Feature-Attention-Network-for-Saliency-Detection

an edge preservation loss is proposed to guide the network to learn more detailed information in boundary localization. With these powerful feature extraction methods and attention mechanisms, PFA network can achieve robust and effective salient object detection, outperforming SOTA methods under different evaluation metrics.

Specifically, given an input image, PFA can generate a saliency score between 0 and 1 for each pixel. The higher value denotes higher visual saliency. We then use a threshold ϕ to transform the saliency scores into a binary saliency mask, which determines the salient region. The binarization can be expressed as

$$s_i^* = \begin{cases} 0 & s_i < \phi \\ 1 & s_i \geq \phi \end{cases} \tag{3.4}$$

where s_i and s_i^* are the saliency score and the binary mask at the i-th pixel position, respectively. Thus, the salient region is the set of the pixels masked by 1.

$$\mathcal{S} \triangleq \{i \mid s_i^* = 1,\ i \in \mathcal{V}\}. \tag{3.5}$$

3.3.3 Refining Perturbations in Salient Region

The proposed Saliency Attack is outlined in Algorithm 1. The search process (lines 6-35) recursively refines perturbations based on a tree structure for an image, which is illustrated in Fig. 3.4. Concretely, an input image is first split into some initial blocks (a coarse grid in Fig. 3.4), and only the blocks in salient regions are kept (lines 7-11). Then we try to add $+\epsilon$ or $-\epsilon$ perturbations on each initial block individually (lines 12-19) and sort them according to their influence (F) on model output (line 26). After that, we choose the block with most influence for further refinement if its F is better than the current best $\hat{F}$ (lines 27-35). We then recursively refine the current best block (line 32), split this block into smaller blocks (a finer grid in Fig. 3.4), and again try to add a perturbation on each block individually (at this time we just flip the perturbation (lines 21-24) for convenience, e.g., $+\epsilon$ to $-\epsilon$) to find the best smaller

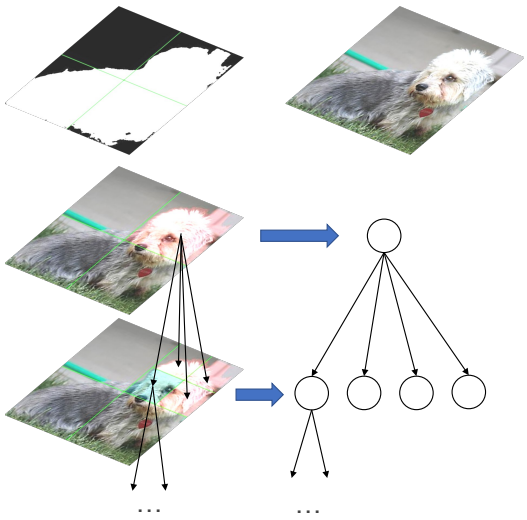


Figure 3.4: The illustration of refining on a tree structure of the image blocks. Among initial blocks, each block is a root node of a tree, and its child nodes are its finer blocks. For each block, only the parts in salient regions will be perturbed.

block. The refinement process recurs until the minimal block (e.g., 1 pixel, line 30) or no smaller block has a better F (line 28). Afterwards, we backtrack to the last level of split blocks and use the second-best block for further perturbation. In this way, the most important block will be explored first, and the perturbations could be as small as possible, with no need to initialize a global perturbation like Parsimonious Attack [125].

To make full use of query budget and combine the advantages of different initial block sizes k_{int} (large k_{int} can quickly lead to successful attacks with large perturbations, while small k_{int} enables the refinement for small perturbations in finer grids), we leverage an outer iteration to run the aforementioned *Refine* search with decreasing k_{int} (lines 2-5). During the iterations, the algorithm will stop if the generated x_{adv} succeeds to fool the model ($F > 0$) or the termination condition is reached (line 2).

We use the loss function from CW attack [20] of untargeted attack for Eq. (3.2):

$$L(x, y_{gt}) = -\max(Z(x_{adv})_{y_{gt}} - \max_{i \neq y_{gt}}(Z(x_{adv})_i),\ -\kappa). \tag{3.6}$$

where $Z(x_{adv})_{y_{gt}}$ is the logit of the AE with respect to the ground-truth class of the original image. In this way, the AE tends to be misclssified into the class i, with a margin κ of the loss function between the ground-truth class and other classes (We set $\kappa = 0$ in our work).

3.4 Experiments

The main goal of the experiments is to validate: *(1) whether salient regions could improve the imperceptibility of existing black-box attacks; (2) whether our Saliency Attack could further enhance the imperceptibility performance.* Hence, we first compare our Saliency Attack with the baselines, including SOTA black-box attacks and their modified version restricted in salient regions. Then we perform ablation studies to verify the effectiveness of salient regions and *Refine* search separately. Besides, we measure the hyperparameter sensitivity of our approach. Finally, we test different detection-based defenses to evaluate the imperceptibility from the defensive perspective.

3.4.1 Settings

We compare our Saliency Attack against SOTA black-box attacks including Boundary Attack [16], HSJA [22], TVDBA [101], Parsimonious attack [125] and Square attack [5]. Among them, Boundary Attack and HSJA start from an already misclassified noise and gradually approach the original image, TVDBA tries to minimize the perturbation via integrating Structural SIMilarity (SSIM) [187] into the loss, Parsimonious Attack proposes discrete optimization to improve query efficiency, and Square Attack can achieve the current SOTA query efficiency and success rate in the black-box attack setting. In addition, we design Parsimonious-sal Attack and Square-sal Attack as baselines that restrict their perturbations in salient regions. For

Algorithm 1: Saliency Attack

Input: Objective set function F, Ground set $\mathcal{V}$, salient region $\mathcal{S}$, initial block size k_{int}, query budget

Output: $\mathcal{S}^+, \mathcal{S}^-$ set of pixels with $+\epsilon$ and $-\epsilon$ perturbations

$\mathcal{S}^+ \leftarrow \emptyset$; $\mathcal{S}^- \leftarrow \emptyset$; $\hat{F} \leftarrow -\infty$;

while $k_{int} > 1$ *and not exceeding query budget* **do**

  Refine$(\mathcal{V},\ k_{int},\ 1)$;

  $k_{int} \leftarrow k_{int}/2$;

Procedure *Refine*$(B,\ k,\ split\_level)$

$k' \leftarrow \sqrt{|B|}$ is the block size of B; $n \leftarrow (k'/k)^2$ is the number of split blocks;

$\{B_1, B_2, ..., B_n\} \leftarrow$ split B into n finer blocks;

for $i \leftarrow 1$ **to** n **do**

  $B_i \leftarrow B_i \cap \mathcal{S}$;

if $split\_level = 1$ **then**

  for $j \leftarrow 1$ **to** n **do**

    if $F(\mathcal{S}^+ \cup B_j, \mathcal{S}^- \setminus B_j) \geq F(\mathcal{S}^+ \setminus B_j, \mathcal{S}^- \cup B_j)$ **then**

      $\mathcal{S}_j^+ \leftarrow \mathcal{S}^+ \cup B_j$; $\mathcal{S}_j^- \leftarrow \mathcal{S}^- \setminus B_j$; $F_j \leftarrow F(\mathcal{S}^+ \cup B_j, \mathcal{S}^- \setminus B_j)$;

    else

      $\mathcal{S}_j^+ \leftarrow \mathcal{S}^+ \setminus B_j$; $\mathcal{S}_j^- \leftarrow \mathcal{S}^- \cup B_j$; $F_j \leftarrow F(\mathcal{S}^+ \setminus B_j, \mathcal{S}^- \cup B_j)$;

else

  for $j \leftarrow 1$ **to** n **do**

    $\mathcal{S}_j^+ \leftarrow (\mathcal{S}^+ \setminus B_j) \cup (\mathcal{S}^- \cap B_j)$; $\mathcal{S}_j^- \leftarrow (\mathcal{S}^- \setminus B_j) \cup (\mathcal{S}^+ \cap B_j)$;

    $F_j \leftarrow F(\mathcal{S}^+, \mathcal{S}^-)$;

$\{F_{\pi(1)}, F_{\pi(2)}, ..., F_{\pi(n)}\} \leftarrow$ sort $\{F_1, F_2, ..., F_n\}$ in descending order;

our Saliency Attack, the threshold ϕ to produce binary saliency masks is chosen to be 0.1. For splitting, original images are resized to 256×256, and we set the initial

block size k_{int} to 16. All parameters of the compared attacks remain consistent with those recommended in their papers.

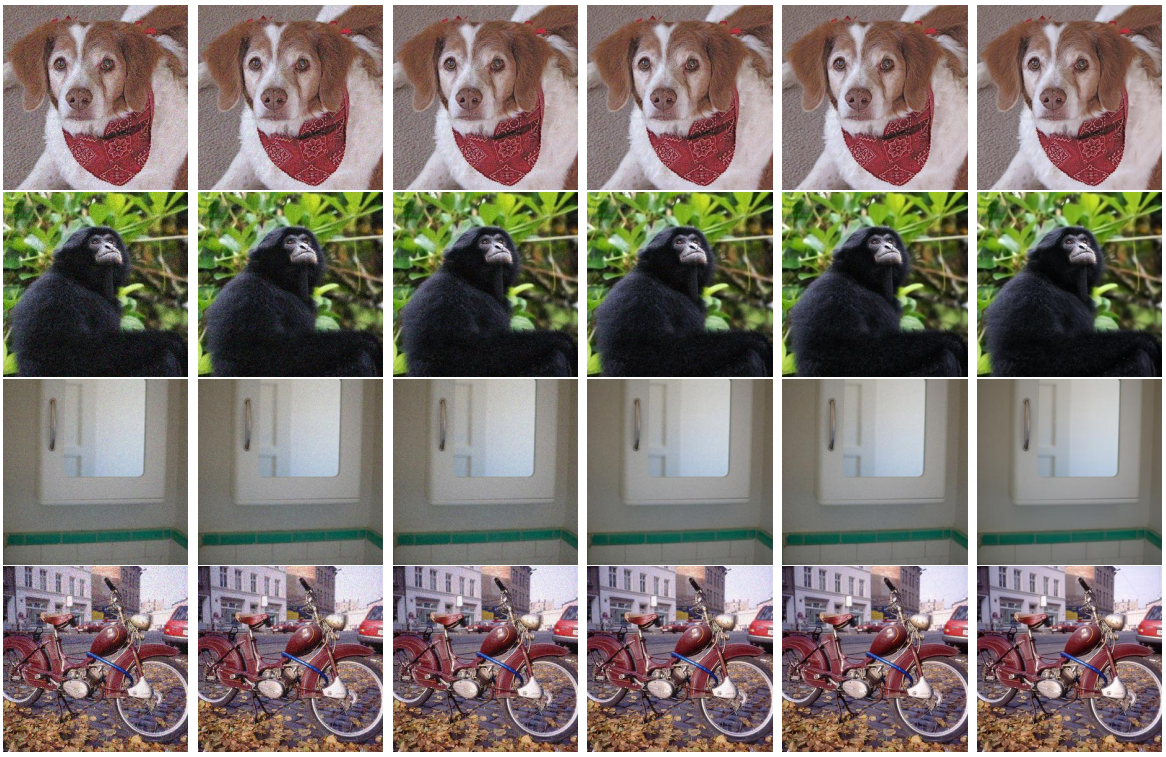
In this study, we consider L_∞ threat model on ImageNet dataset [36], and set the perturbation magnitude ϵ to 0.05 in [0,1] scale. We use Inception v3 [172] and ResNet50 [61] as the target model, and the query budget is 10,000 for untargeted attack. We provide our implementation publicly[2].

For performance metrics, we employ commonly used success rate (SR) and average number of queries (AQ). Only the AEs that can fool the target model below the max query budget will be recorded as successful. And the number of queries used by successful AEs will be counted. To evaluate the imperceptibility of AEs, we consider L_0, L_2 and MAD. All these three imperceptibility metrics are the smaller the better. In practice, a successful AE with imperceptible perturbation is what we need. So besides SR, we use a new metric SR_{true}. Since Boundary Attack [16] can reduce the distortion gradually, we generate multiple adversarial examples with different MAD scores in Fig. 3.5 to find a proper threshold for MAD metric. It indicates the rate of successful AEs whose MAD values are below a human-like threshold, i.e., 30, that AEs with MAD ≤ 30 are basically imperceptible to human eyes. We further calculate the effective number of queries $\text{EQ} = \text{AQ}/\text{SR}_{\text{true}}$ to denote the number of queries needed to generate a successful AE with imperceptible perutrbations. EQ is a comprehensive metric, taking into account attack success rate, query efficiency and imperceptibility.

3.4.2 Results and Analysis

Some examples of different attacks are shown in Fig. 3.6. We can easily find the perturbations of original Parsimonious Attack and Square Attack are very obvious in the entire image due to global perturbation, while their modified versions are relatively

[2]https://github.com/Daizy97/SaliencyAttack



(a) MAD=50 (b) MAD=40 (c) MAD=30 (d) MAD=20 (e) MAD=10 (f) Original

Figure 3.5: Adversarial examples generated by boundary attack with different MAD scores. We can find the threshold MAD $\leq$ 30 is roughly enough to indicate an imperceptible adversarial example.

more imperceptible since no perturbation exists in background regions. However, their perturbations are still complex and irregular, taking up almost all salient regions. In comparison, even restricted in the same salient regions, the perturbations generated by Saliency Attack are smaller and more critical, roughly corresponding to the bright pixels in BP-saliency map. They further represent the positions of dogs' noses or ears, which accord with our inspiration before. Besides, TVDBA produces global and irregular perturbations, and the AEs of Boundary Attack contain noticeable coarse textures due to inadequate query budget. We omit HSJA since it generates similar but blurrier AEs with Boundary Attack's.

The quantitative results are reported in Tab. 3.1 and Table 3.2. In the experiments,

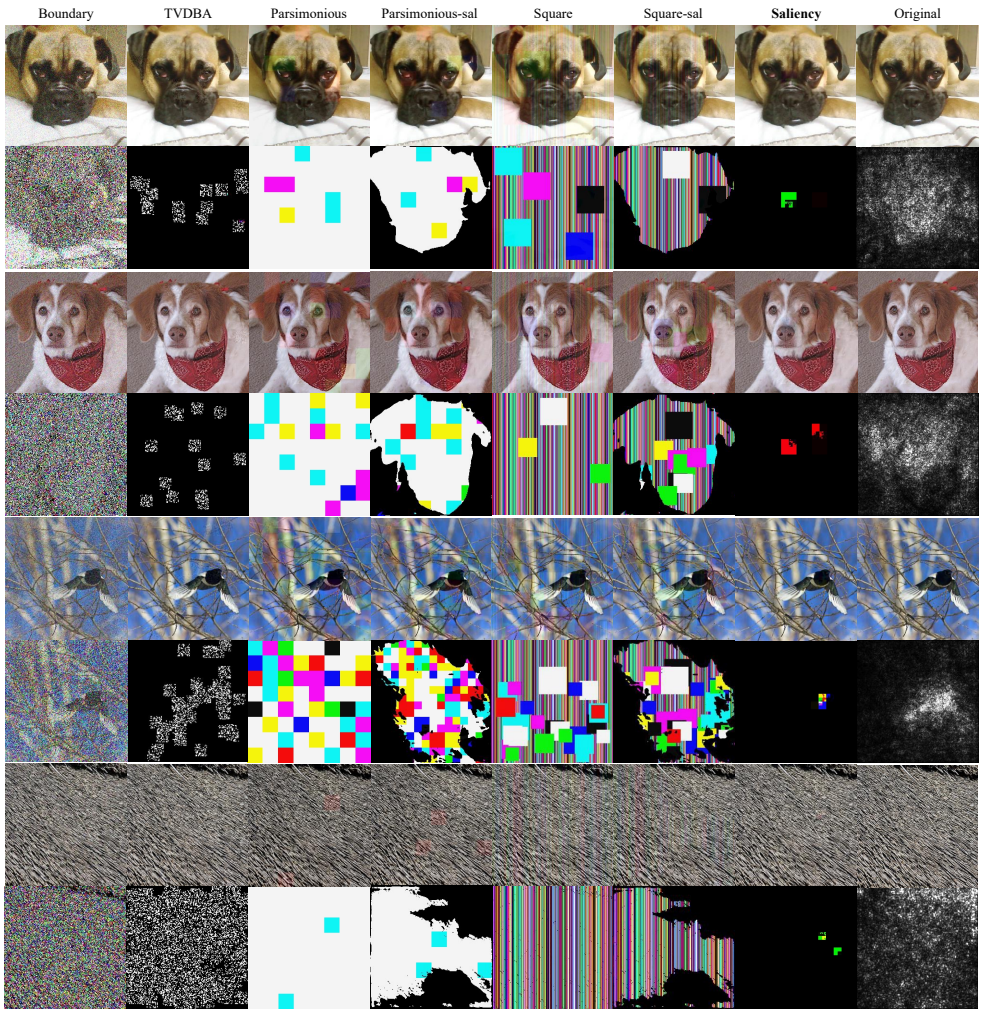


Figure 3.6: Examples generated by different attacks. For each example, the upper row is AEs and original image. The lower row is perturbations and BP-saliency map.

we randomly choose 10,000 images from the ImageNet validation set and divide them into ten groups. Then we calculate the mean and standard deviation (SD). The best results are recorded in bold based on Wilcoxon signed-rank test with significance level at 0.05. We can find in all query budgets, our Saliency Attack statistically significantly outperforms all the baselines on different target models with a huge gap in terms of $\mathrm{SR}_{\mathrm{true}}$, EQ and three imperceptibility metrics, which demonstrates the superiority of our method to refine the perturbations in salient regions. Specifically, for Boundary Attack and HSJA, although they gradually sample AEs to approach

the original images while keeping adversarial, their perturbations are hard to decrease to the L_∞ constraint under the max query budget. As an example, it takes at least hundreds of thousands of queries for Boundary Attack to converge [16], which is infeasible in practice. That's why they have very bad SR and imperceptibility performance (Boundary Attack and HSJA can exhaust all the query budget to reduce perturbations, so we don't count their query number). On the contrary, the remaining baselines and our Saliency attack start from original images and gradually add perturbations, constrained in the pre-defined L_∞ distance. Among TVDBA, Parsimonious Attack, Square Attack and Saliency Attack, they all adopt the "tiling trick" [76] to sample or perturb a square of pixels to improve efficiency, and Square Attack achieves the best SR and AQ due to the random vertical stripes (RVS) [5] as initializations (a part of test images can directly fool the target model with RVS using only one query). However, this performance significantly sacrifices the imperceptibility of Square attack with worst $\mathrm{SR}_{\mathrm{true}}$, EQ and three imperceptibility metrics among four attacks. For our Saliency Attack, despite slight degradation in SR, it can achieve much better $\mathrm{SR}_{\mathrm{true}}$, EQ and three imperceptibility metrics, which means Saliency Attack can efficiently generate successful AEs with imperceptible perturbations, and are more likely to evade various defenses or human judgements in practical.

Table 3.1: Comparison of different attacks against Inception v3 model.

Method	SR ± SD	$\mathrm{SR}_{\mathrm{true}}$ ± SD	AQ ± SD	EQ ± SD	L_2 ± SD	L_0 ± SD	MAD ± SD
Boundary	2.1%±0.01	0.2%±0.00	-	-	79.46±2.29	99.5%±0.00	99.07±2.15
HSJA	1.1%±0.01	0.1%±0.00	-	-	97.39±1.71	100%±0.00	121.07±1.65
TBDBA	96.9%±0.01	23.7%±0.02	699±16	2949±263	10.83±0.18	19.8%±0.00	37.87±3.61
Parsi	98.5%±0.01	14.8%±0.01	745±21	5034±925	22.17±0.00	100.0%±0.00	53.01±0.14
Parsi-sal	96.0%±0.01	12.6%±0.01	843±25	6690±1154	13.47±0.48	35.5%±0.03	45.44±0.22
Square	**99.6%±0.00**	2.4%±0.00	**222±16**	9250±2884	25.35±0.02	99.0%±0.00	57.74±0.05
Square-sal	96.0%±0.01	22.8%±0.02	576±23	2526±337	14.56±0.51	34.3%±0.03	38.65±0.56
Saliency	92.4%±0.01	**85.8%±0.01**	1947±125	**2282±198**	**3.63±0.05**	**3.4%±0.00**	**12.62±0.21**

For Parsi-sal and Square-sal attack, they obtain better $\mathrm{SR}_{\mathrm{true}}$ and three imperceptible

Table 3.2: Comparison of different attacks against ResNet50 model.

Method	SR ± SD	SR_{true} ± SD	AQ ± SD	EQ ± SD	L_2 ± SD	L_0 ± SD	MAD ± SD
Boundary	17.4%±0.02	4.5%±0.01	-	-	30.14±0.93	99.2%±0.00	57.90±1.22
HSJA	13.8%±0.02	2.8%±0.00	-	-	42.33±1.07	100%±0.00	81.34±1.52
TBDBA	99.9%±0.00	23.4%±0.02	381±14	1628±202	9.62±0.15	18.7%±0.00	39.02±4.13
Parsi	99.9%±0.00	13.9%±0.01	178±8	1292±113	22.17±0.00	100.0%±0.00	51.34±0.52
Parsi-sal	99.0%±0.00	8.7%±0.02	344±31	4078±720	12.52±0.55	34.0%±0.03	48.73±1.17
Square	**100%±0.00**	1.6%±0.00	**47±4**	3356±1200	19.00±0.01	99.1%±0.00	57.19±0.37
Square-sal	97.6%±0.01	22.3%±0.02	328±48	1475±194	10.79±0.44	34.2%±0.03	39.47±1.01
Saliency	97.2%±0.01	**67.3%±0.02**	676±21	**1006±56**	**4.16±0.13**	**5.4%±0.00**	**24.53±0.48**

tibility metrics than their original version due to restricting perturbations in small but critical regions, which demonstrates different black-box attack can also benefit from salient regions. Their SR and AQ also have some slight degradation because the search space is reduced and some test images need more queries to generate successful AEs. In terms of the comprehensive metric EQ, We can find Square-sal attack is improved while Parsi-sal Attack is degraded. We further study the change of their MAD scores and find most samples' MAD scores of Square-sal Attack are enhanced due to its random search scheme. But Parsimonious Attack adopts greedy local search, that plays a similar role with salient regions, so the benefit from salient regions is relatively limited.

As shown in Fig. 3.7, via restricting perturbations in salient region, 68.1% and 94.3% examples of Parsimonious-sal attack and Square-sal attack have better MAD scores compared with their original version, respectively. Since Parsimonious Attack uses a greedy local search that plays a similar role with salient regions, while Square attack adopts a random search for sampling perturbations, it is obvious that limiting the search in salient regions is more helpful to Square Attack than Parsimonious Attack.

We also plot the convergence curve of SR_{true} versus query number under different MAD thresholds in Fig. 3.8. We omit Boundary Attack and HSJA since they have

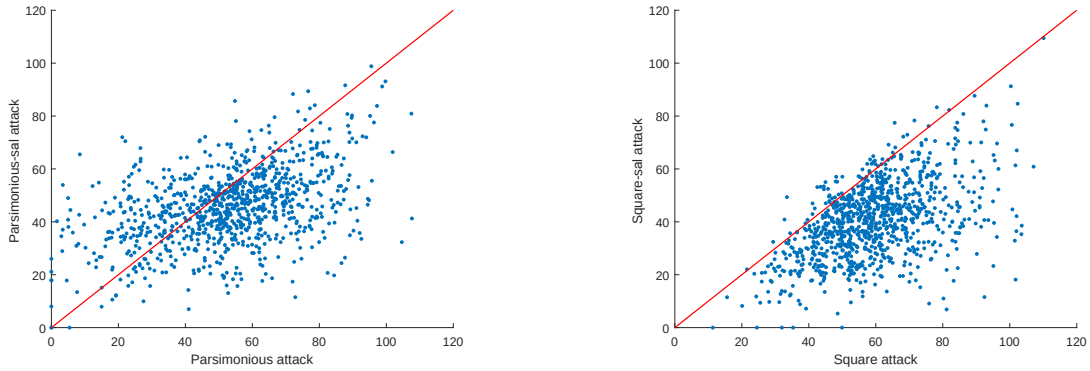


Figure 3.7: MAD scores of Parsimonious Attack vs. Parsimonious-sal attack and Square Attack vs. Square-sal Attack.

very low SR. It shows that Saliency Attack can lead other attacks stably most of the time. For some baselines such as Square Attack and Parsimonious Attack, they can generate some successful AEs with very few queries due to global perturbations and some tricks like RVS as initializations, but they lack the ability to further improve the imperceptibility. Instead, Saliency Attack conservatively selects small regions to perturb, hence its query efficiency is a little lower than some baselines at the beginning but soon exceeds them.

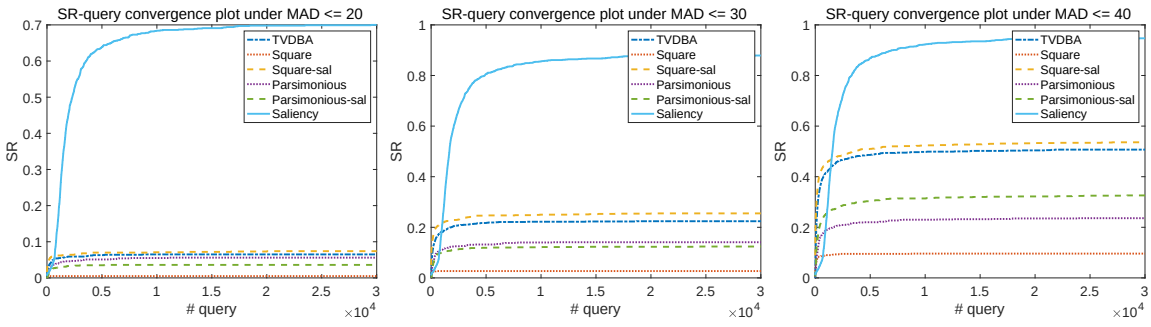


Figure 3.8: Queries vs. True Success Rate under different thresholds of MAD scores

3.4.3 Ablation Study

We carry out ablation studies of Saliency Attack, including refining in salient region, in non-salient region, and without saliency (refining in the whole image). We also design a greedy search as a baseline to verify our *Refine* search. We test different block sizes for greedy search in salient region in Tab. 3.3, and choose the best one (block size equals 32) to be compared. The results on 1,000 randomly selected images and some examples are given in Tab. 3.4 and Figure 3.9. Note that refining in salient region and refining without saliency generate the same or almost the same perturbations, which means the salient regions indeed contain useful parts and enhance the query efficiency by limiting the search space. But for refining in non-salient region, its perturbation is more complex and visible with worse query efficiency and SR due to unuseful search space. Compared with greedy search, our *Refine* search has much better query efficiency and SR_{true}, which demonstrates its superiority. Therefore, we can conclude that both salient region and *Refine* search facilitate Saliency Attack.

Table 3.3: Greedy search in salient region with different block sizes.

Block size	SR	SR_{true}	AQ	EQ	L_2	L_0	MAD
128	20.6%	18.4%	57	310	7.32	12.8%	11.50
64	37.4%	31.3%	512	1636	6.37	10.2%	15.18
32	56.0%	**50.7%**	2727	5379	4.37	4.7%	12.87
16	35.4%	35.2%	4039	11474	1.79	0.8%	4.84

3.4.4 Hyperparameter Sensitivity

The hyperparameter k_{int} is the initial block size that determines the first level of split blocks. We test the effect of different k_{int} on SR, query number, and imperceptibility (L_2 and MAD) over 1,000 randomly selected examples in Fig. 3.10. As k_{int} decreases, SR and imperceptibility can be improved, and SR reaches the peak when k_{int} equals

Table 3.4: Ablation study of Saliency attack against Inception v3 model.

Method	SR	SR_{true}	AQ	EQ	L_2	L_0	MAD
Refining in salient region	93.6%	**86.2%**	**1958**	**2271**	**3.71**	**3.8%**	**12.88**
Refining in non-salient region	78.2%	57.0%	3128	5488	4.94	6.5%	21.58
Refining without saliency	**95.5%**	79.6%	2563	3220	3.84	4.2%	16.35
Greedy search in salient region	56.0%	50.7%	2727	5379	4.37	4.7%	**12.87**

(a) Salient region (b) In salient region (c) Without saliency (d) In non-salient region

Figure 3.9: Examples generated by Saliency Attack with different strategies.

16. This is because, with smaller initial blocks, Saliency Attack can search for perturbations more finely and accurately leading to higher SR and better imperceptibility. Meanwhile, inevitably more queries are needed, especially for sorting initial blocks. Under a limited query budget like 10,000, the remaining query budget for searching in finer blocks could be inadequate. That's why a turning point occurs in SR.

We also show some examples in Fig. 3.11. It can be observed that as k_{int} decreases, the generated perturbations become smaller but more salient. For instance, in the first row, the perturbation with $k = 128$ roughly covers the region of the dog face, which is also the brightest region in the BP-saliency map. While the perturbation with $k = 32$ or $k = 16$ focuses on smaller region of the dog ear. This indicates that our Saliency Attack could find smaller and more important perturbations progressively

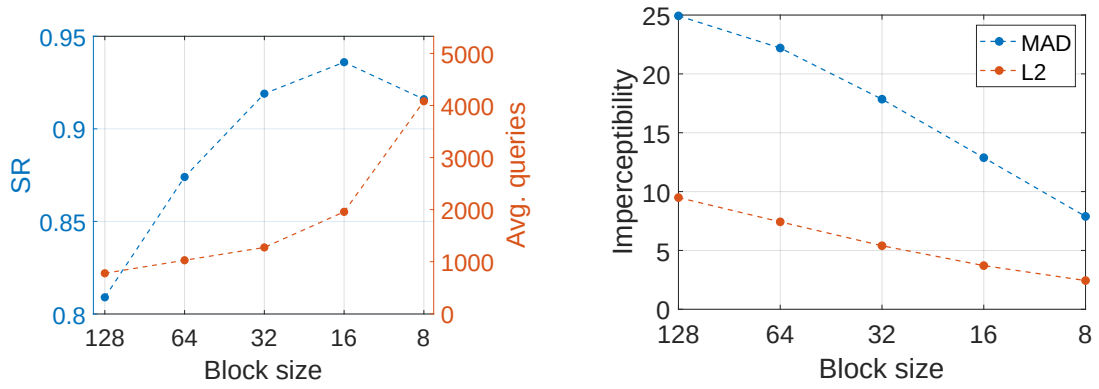
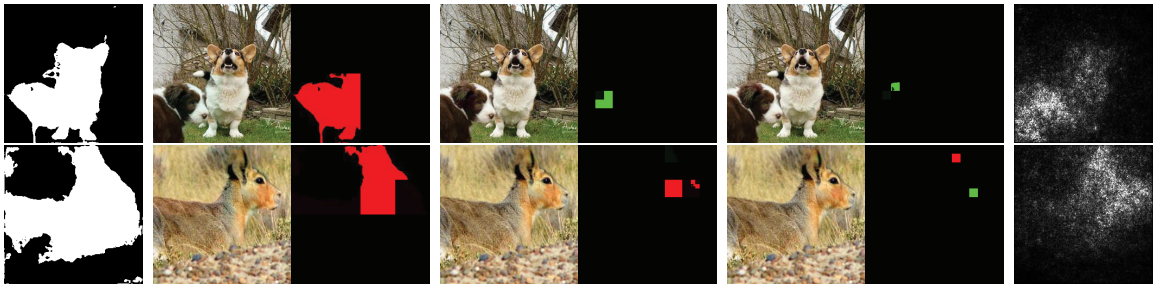


Figure 3.10: The effect of initial block size on SR, average query number and imperceptibility (L_2 and MAD).

as the k_{int} decreases, which coincides with our assumption before. The perturbations are also interpretable to some extent, like perturbing the specific regions of a dog's ears, eyes or nose.



(a) Salient region (b) $k_{int} = 128$ (c) $k_{int} = 32$ (d) $k_{int} = 16$ (e) BP-saliency map

Figure 3.11: Examples generated by Saliency Attack with different initial block sizes.

For another hyperparameter ϕ, which is the threshold to produce binary saliency masks from the saliency maps generated by salient object detection models. We use SOTA salient object detection models PFA [222] and TRACER [94] to generate saliency masks with different thresholds {0.025, 0.05, 0.1, 0.2, 0.3}. The results are reported in Tab. 3.5 and some examples are displayed in Fig. 3.12. We can find using different salient object detection models with different thresholds generate similar

salient masks, extracting the regions of main object(s) in images. Although PFA and TRACER may generate some distinct saliency masks (e.g., the first dog example in Fig. 3.12), these saliency masks all cover the critical regions (e.g., the region of the dog's face). That's why only slight differences exist in the quantitative results using PFA or TRACER with different thresholds. In our work, we select PFA with $\phi = 0.1$, which slightly outperform others in terms of SR and SR_{true}.

Table 3.5: Results of Saliency Attack against Inception v3 model using different salient object detection models with different thresholds.

Method	Threshold	SR	SR_{true}	AQ	EQ	L_2	L_0	MAD
PFA	0.025	92.90%	85.30%	1891	2217	3.69	3.80%	12.52
	0.05	92.00%	85.60%	1842	2152	3.67	3.70%	12.39
	0.1	93.90%	86.00%	1979	2301	3.73	3.90%	12.94
	0.2	89.90%	84.30%	1751	2077	3.58	3.50%	11.89
	0.3	89.60%	84.50%	1721	2037	3.59	3.50%	12.00
TRACER	0.025	92.90%	85.90%	1891	2201	3.69	3.80%	12.53
	0.05	92.00%	85.60%	1842	2152	3.67	3.70%	12.51
	0.1	92.10%	85.30%	1785	2093	3.66	3.70%	12.51
	0.2	92.10%	85.80%	1784	2079	3.66	3.70%	12.45
	0.3	92.00%	85.30%	1772	2077	3.66	3.70%	12.40

3.4.5 Attacking Detection-based Defense

To further validate the imperceptibility of AEs, we attack against two detection-based defenses that attempt to detect AEs from clean images. The first is Feature Squeezing[3] [205]. Its basic idea is to squeeze out unnecessary input features that may be utilized by an adversary to generate AEs. Comparing the model's prediction on

[3]Implementation of Feature Squeezing: https://github.com/mzweilin/EvadeML-Zoo

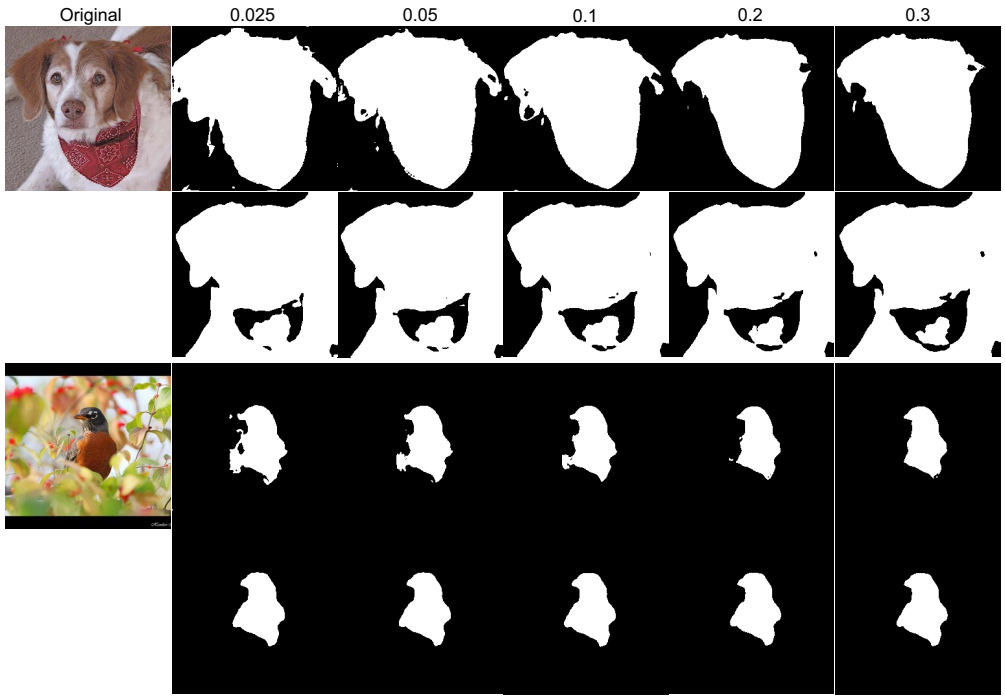


Figure 3.12: Binary saliency masks from the saliency maps generated by PFA and TRACER with different thresholds. For each example, the upper row is generated by PFA. The lower row is generated by TRACER.

the original image with its prediction on the image after squeezing, the input is likely to be adversarial if the original and squeezed inputs produce substantially different outputs. Feature Squeezing adopts color bits reduction of each pixel and spatial smoothing (local and non-local smoothing) as squeezers. We use the recommended parameters $(5 - bit, 2 \times 2, 11 - 3 - 4)$ for ImageNet dataset. Another is a CNN-based binary classifier based on a Steganography detector [13], which is designed for detecting small noise specifically and outperforms classic classification models such as Inception-v3 or ResNet. We finetune the pre-trained model[4] with AEs generated by different attacks. Concretely, we randomly select 1,000 clean images and generate the corresponding AEs with Saliency Attack and six baselines, respectively. Thus, we use $2 \times 7,000$ pairs of clean images and AEs as the training set to finetune the model.

[4]https://github.com/brijeshiitg/Pytorch-implementation-of-SRNet

The training is run for 100 epochs, and the learning rate is 0.001. We test another 1,000 samples and record the detection rate (DR), which is the lower the better.

Table 3.6: Results for attacking against detection-based defenses

Method	SR	SR_{true}	Feature Squeezing DR	Classifier DR
TVDBA	96.9%	23.2%	26.5%	61.9%
Parsimonious	98.2%	14.1%	33.3%	89.7%
Parsimonious-sal	96.7%	12.0%	29.7%	67.6%
Square	99.7%	1.9%	47.8%	85.4%
Square-sal	97.1%	22.5%	26.1%	63.1%
Saliency (ours)	93.6%	86.2%	**21.2%**	**50.4%**

From Table 3.6, we can find our Saliency attack can achieve lower detection rate against both Feature Squeezing and binary classifier detection than other baselines. For Feature Squeezing, although the area of perturbation generated by Saliency Attack is quite small and thus suffering a higher risk of being denoised by squeezers, the detection rate of Saliency Attack is still better than other attacks due to our effective perturbation. For the binary classifier detector, our Saliency Attack obtains nearly 50% accuracy/detection rate, which means the perturbation is invisible enough that the classifier will be difficult to converge and results in random guess. Therefore, our Saliency Attack is able to generate imperceptible AEs that evade different detection-based defenses.

3.5 Conclusion

In this paper, we propose restricting black-box attacks to perturb in salient regions to improve the imperceptibility of AEs, and propose a novel black-box attack, Saliency Attack. Experiments show that SOTA black-box attacks restricted in salient regions

can indeed achieve better imperceptibility performance, while Saliency Attack further enhances this by recursively refining perturbations in salient regions. Its perturbation is much smaller, imperceptible and interpretable to some extent. Besides, we also find that the salient regions of some examples are indeed progressive with respect to their impact on model output, which is consistent with our assumption from the previous visualization studies. Finally, the evaluation demonstrates that the AEs generated by Saliency Attack are harder to be detected, which validates its imperceptibility from the defensive perspective.

Although our *Refine* search provides significant benefit to the imperceptibility, it inevitably consumes more queries compared with the black-box attacks with global perturbations. In the future, we will try to find better ways to balance the three objectives of query efficiency, imperceptibility and attack success rate for black-box attacks. Furthermore, we will apply the salient region to more black-box attacks to improve their imperceptibility. Our Saliency Attack could also be tested against other kinds of defenses, such as robustness-based defenses (adversarial training, network distillation, etc.), to validate its effectiveness.

Chapter 4

Joint Universal Adversarial Perturbations with Interpretations

Deep neural networks (DNNs) have significantly boosted the performance of many challenging tasks. Despite the great development, DNNs have also exposed their vulnerability. Recent studies have shown that adversaries can manipulate the predictions of DNNs by adding a universal adversarial perturbation (UAP) to benign samples. On the other hand, increasing efforts have been made to help users understand and explain the inner working of DNNs by highlighting the most informative parts (i.e., attribution maps) of samples with respect to their predictions. Moreover, we first empirically find that such attribution maps between benign and adversarial examples have a significant discrepancy, which has the potential to detect universal adversarial perturbations for defending against adversarial attacks. This finding motivates us to further investigate a new research problem: whether there exist universal adversarial perturbations that are able to jointly attack DNNs classifier and its interpretation with malicious desires. It is challenging to give an explicit answer since these two objectives are seemingly conflicting. In this paper, we propose a novel attacking framework to generate joint universal adversarial perturbations (**JUAP**), which can

fool the DNNs model and misguide the inspection from interpreters simultaneously. Comprehensive experiments on various datasets demonstrate the effectiveness of the proposed method JUAP for joint attacks. To the best of our knowledge, this is the first effort to study UAP for jointly attacking both DNNs and interpretations.

4.1 Introduction

As one of the most important data mining techniques, Deep Neural Networks (DNNs) have achieved massive remarkable performance in various challenging downstream domains [107, 48, 39, 114], e.g., image classifications [223, 204, 220, 221], language generation [209, 138], and social media [38, 182]. These deep learning methods have significantly benefited our humans in daily life and created great economic outcomes in society. Along with their impressive development and great achievement, DNNs techniques have also exposed their untrustworthy aspects [102, 97, 49, 189, 133], which might perform unreliable predictions and cause severe economic, social, and security consequences, especially in safety-critical scenarios. For example, adversaries can maliciously generate well-designed patterns in road signs, such that DNNs-enhanced autonomous vehicles might recognize a stop sign as an acceleration sign [168], leading to great potential risks to personal safety.

Current adversarial perturbations in computer vision tasks can be divided into two types, i.e., image-dependent perturbations (**IDPs**) and image-agnostic universal adversarial perturbations (**UAPs**). Image-dependent perturbations need to be crafted for each image by using iterative/non-iterative gradient decent [121, 56, 15, 8, 104] algorithms or solving an optimization problem [9]. Instead, image-agnostic UAP can be added to most benign images and then cause misclassification of DNNs with very high confidence [126, 137]. Due to the good generalization capability, UAPs, after learning from the dataset, can even attack unseen images. Besides, compared with image-dependent perturbations specific to each sample, image-agnostic UAPs,

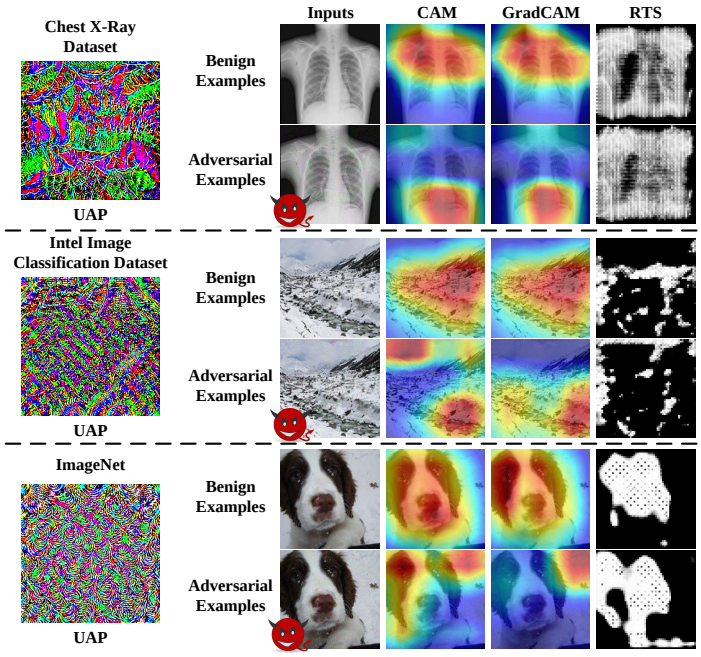


Figure 4.1: An illustration of interpretation discrepancy for benign and adversarial images (via adding universal adversarial perturbations) under various interpreters.

once learned and deployed, are more efficient, especially when facing large-scale data samples. Thus, image-agnostic universal adversarial perturbations pose a significant threat to the security of DNNs.

In recent years, in order to enhance the trustworthiness of DNNs models, increasing efforts have been made to help users understand the inner working mechanism of DNNs models by providing interpretations on how certain decisions are made [226, 156, 62, 134]. More specifically, the interpretations can be achieved by identifying the most influential parts (e.g., attribution maps) of the input with respect to its prediction [160, 31, 147, 53, 117, 45]. For example, as the first activation-based interpreter, CAM was proposed to compute a weighted sum of the activation of the last convolutional layer to generate attribution maps, where the weights are the parameters of the final dense layer [226]. Simonyan et al. [160] proposed to use the derivative as the attribution maps, considered as the gradient-based method. On the other hand,

recent works have found that DNNs models' interpretability can work as inspectors or detectors (i.e., model debugging) to diagnose models' errors and reveal models' vulnerability (i.e., adversarial perturbations) in computer vision domain [1] and text domain [96, 95, 180]. As a result, interpreters have the potential to detect adversarial perturbations for defending against adversarial attacks.

In order to verify such detection capabilities in the image domain under various DNNs' interpreters (i.e., CAM, GradCAM, and RTS) for UAP attacks [126], we first perform empirical studies to evaluate the interpretation discrepancy between interpretation maps on the benign and adversarial images. As shown in Fig. 4.1, the attribution maps on adversarial examples are significantly different from those of benign examples. Once such 'confused' predictions are inspected, humans (e.g., model designers) can develop some strategies to enhance the DNNs model's robustness against adversarial attacks. In other words, interpreters can be further exploited as anomaly detectors to improve the system's safety. Thus, our experimental results validate that *DNNs' interpretations provide great opportunities to detect universal adversarial perturbations for defending against adversarial attacks.*

In order to evade interpreters' detection with malicious desires, we further raise a new question, i.e., *whether there exist universal adversarial perturbations that can perform a joint attack to attack DNNs classifier and its interpretation simultaneously with malicious desires?* To be specific, by adding the universal adversarial perturbations to the benign images, the objectives of the joint attack are to maximize the change of DNNs' predictions while minimizing the interpretation discrepancy, leading to more severe safety issues for DNNs techniques. However, these two objectives are essentially contradictory to each other. That is because the added universal perturbations tend to be the most influential part of natural images for obtaining incorrect labels, while the goal of interpreters is to identify the most influential maps for explaining how do DNNs models make decisions. Therefore, to investigate such new questions, in this paper, we propose a novel attacking framework (**JUAP**) for generating universal adversarial

perturbations, which can simultaneously mislead the DNNs model's prediction and misguide the inspection from the interpreter.

Our major contributions are summarised as follows:

- We discover that DNNs modes' interpretability can be leveraged to understand and inspect the problematic outputs from adversarial examples caused by universal adversarial perturbations, which pave the way to enhance the safety of DNNs models.
- We study a novel problem of exploring universal adversarial perturbations to jointly attack DNNs models as well as their interpretations, which reveals more severe safety concerns. To the best of our knowledge, we are the first to investigate universal adversarial perturbations from the perspective of interpreters' detection.
- We propose a novel attacking framework to generate joint universal adversarial perturbations (**JUAP**), where a generative adversarial network is proposed to learn universal perturbations for fooling the DNNs model and misguide the inspection from interpreters simultaneously.
- Our comprehensive experiments on three real-world datasets demonstrate that the proposed attacker can achieve a high fooling ratio for attacking classifiers with a small interpretation discrepancy, which successfully achieves the joint attack on DNNs and their coupled interpreters.

The remainder of this paper is organized as follows. In Section 4.2, we briefly review some related works. In Section 4.3, we introduce the problem definition. Then, we present the details of our proposed method to generate joint universal adversarial perturbations in Section 4.4. In Section 4.5, we conduct massive experiments and demonstrate the effectiveness of our JUAP. At last, we conclude our work with future directions in Section 4.6.

4.2 Related works

In this section, we briefly review some related work about adversarial attacks and interpretation models.

4.2.1 Universal (Input-agnostic) Adversarial Perturbations

Deep neural networks are susceptible to deception by adversarial attacks. That is, the output of a DNNs model can be fooled by adding well-designed, imperceptible perturbations to the input [56, 121]. More specifically, by considering image-agnostic perturbations, adversaries can find universal adversarial perturbations (UAPs) that can deceive the neural network model when added to most images. Since only one perturbation is required to be generated, UAP has a significant advantage in computational efficiency over image-dependent perturbations [32]. Moosavi-Dezfooli et al. [126] presented the first data-driven iterative UAP generation algorithm that uses *Deep Fool* [127] for each unsuccessfully attacked image to identify the minimum perturbation that can deceive the model and add it to the UAP. Although the iterative approach can generate UAPs with a high attack success rate, its computational cost is quite expensive. Therefore, generative methods are applied to generate UAPs [142, 105, 128]. Hayes et al. [60] suggested using a universal adversarial network (UAN) to learn the overall distribution of UAPs instead of fixing one UAP. Thus the UAN can quickly generate different UAPs depending on the different input noises. Mopuri et al. [128] further introduced a new loss function to ensure that the generated perturbations are diverse enough and can fool the neural network model successfully.

4.2.2 DNNs Interpretability

Interpreters aim to reveal the underlying reasons why a deep neural network model makes a particular prediction in a human-understandable way. Several interpretation methods have been proposed for analyzing the discriminative regions of image classification models [161, 156, 217]. *CAM* [226] is the first proposed activation-based interpretation method, which performs a weighted sum of the feature maps of the convolutional layers, usually the last layer, for the explanation. To overcome the drawback that *CAM* needs to modify the network architecture, *Grad-CAM* [156] suggested taking the mean value of the gradient as the weight of the feature map and keeping only the regions that have a positive impact on the classification. *Grad-CAM++* [21] weights the gradients for the problem that there may be more than one class in the image, making the localization more accurate. Another category of interpretation methods are gradient-based methods, which calculate the gradient of the output over the input image to determine which pixels have a stronger impact on the prediction [163]. However, the saliency maps obtained by such methods are usually not clear enough. Instead of relying on gradients, the model-based interpreters train a masking model to predict saliency maps during forward propagation [31].

However, researchers have found that these interpreters can be manipulated arbitrarily. For instance, Heo et al. [67] fine-tuned the neural network model by adding the interpretation results to the penalty term of the loss function, which allowed manipulating the saliency map of the interpreter without compromising the accuracy of the model. Ghorbani et al. [55] introduced adversarial perturbations to interpreters, i.e., it is possible to generate interpretations that differ significantly from the original image by simply adding perturbations to the image that are imperceptible to the naked eye and that do not change the classification result. Dombrowski et al. [40] further manipulated the interpretation methods by using adversarial perturbations to generate arbitrary explanation maps and explained this phenomenon in terms of the geometric properties of neural networks.

4.2.3 Fooling Both Classifiers and Interpreters

Traditional adversarial attack methods change the explanation maps of neural network models while deceiving their predictions, and thus can be easily detected by users. A more threatening class of adversarial attacks is one that can mislead neural network models without changing the explanation maps. By considering *image-dependent perturbations*, Zhang et al. [218] proposed an attack method, ADV^2, to integrate the difference between the attribution maps of the generated adversarial examples and the original images into the loss function to ensure that the explanation maps are not changed when misleading the classifier. Zhang et al. also pointed out that a potential reason why classifiers and interpreters can be successfully attacked simultaneously is that interpreters usually explain only part of the classifier's behavior. Adversarial patches are a more practical class of adversarial attacks that can effectively mislead classifiers, but interpretation methods such as *Grad-CAM* [156] can highlight patch locations and thus be noticed. Subramanya et al. [170] optimize the patches while suppressing the heat map of the patch locations, making the adversarial patches unnoticeable to the interpreter when deceiving the classifier. However, most of these existing attacking methods focus on image-dependent attacks, i.e., they need to generate specific perturbations for each image. To the best of our knowledge, this is the first effort to investigate **universal adversarial perturbations** to jointly deceive the DNNs classifier and interpreter for a trustworthy DNNs model.

4.3 Problem Definition

Given a benign image $x \in \mathbb{R}^d$ with its ground truth $C_x \in \{1, 2, ..., k\}$, the goal of a DNN classifier is to learn a model for label prediction, which can be formulated as $f(x) : \mathbb{R}^d \rightarrow \{1, 2, ..., k\}$.

Adversarial Perturbations. For a benign image x, attackers can generate imperceptible perturbations $\hat{n}$ to obtain its corresponding adversarial example $\hat{x} = x + \hat{n}$ so as to manipulate the DNN classifier $f(\hat{x})$ to make the malicious prediction. Moreover, the perturbations are required to satisfy the imperceptibility property (i.e., perceptually indistinguishable from our human eye), which can be achieved by adding constraint: $\|\hat{n}\|_p \leq \zeta$, where ζ is a hyperparameter used to control the magnitude of the perturbation and p is the vector norm.

In general, adversarial perturbations can be categorized as image-dependent and universal (image-agnostic) perturbations. Given a well-trained DNNs classifier f and a dataset $D = \{x_i, C_{x_i} | i = 1, ..., N\}$ containing N samples with their labels C_{x_i}:

- **Image-dependent Adversarial Perturbations** can be defined as $\hat{n}_i$ for a specific image x_i. The image-dependent perturbations can mislead $f(x_i)$ to provide wrong prediction output, which can be mathematically formulated as:

$$f(x_i) \neq f(x_i + \hat{n}_i), i = 1, \cdots, N. \tag{4.1}$$

  Normally, image-dependent perturbations are devised for each sample, respectively, which is time-consuming and lacks generalization ability. The perturbations generated based on one specific sample may not function well when added to other benign images.

- **Universal Adversarial Perturbations (UAPs)** are defined as $\hat{n}$, aiming to fool classifer f to produce incorrect predictions for most samples in D as follows:

$$f(x_i) \neq f(x_i + \hat{n}), i = 1, \cdots, N. \tag{4.2}$$

  The universal adversarial perturbations are usually learned based on a set of training samples with great generalization properties. Thus, UAPs can also mislead the DNNs' predictions when added to unseen benign samples.

DNNs' Interpretations. Given a classifier f, its coupled interpreters are defined as $\mathcal{I} : \mathbb{R}^d \to \mathbb{R}^d$, aiming to produce an attribution map $\mathcal{I}(x)$, which highlights the most influential parts of the input with respect to its prediction. The attribution map can provide a great way to help users understand and explain the inner working mechanism of how certain decisions are made from the DNNs model. Moreover, recent works have found that DNNs' interpreters have the potential to work as anomaly detectors for improving the security of DNNs against adversarial attacks [1, 96, 95, 180], as demonstrated in our empirical studies (see the Fig. 4.1).

Problem Statement: Given an image x_i, the attacker aims to perform a joint attack by generating universal adversarial perturbations $\hat{n}$, which can fulfill the following two goals:

- The UAP $\hat{n}$ on any sample x_i can change the DNNs classifier prediction to incorrect label:

$$f(x_i) \neq f(x_i + \hat{n}), i = 1, \cdots, N; \tag{4.3}$$

- The UAP $\hat{n}$ on any sample x_i can manipulate the interpreter to produce the same attribution maps as that on that original input:

$$\mathcal{I}(x_i) = \mathcal{I}(x_i + \hat{n}), i = 1, \cdots, N. \tag{4.4}$$

4.4 Methodology

In this section, we introduce the details of our proposed method **JUAP** to achieve the joint attack.

4.4.1 An Overview of Our Proposed Method

The goal of this work is to conduct joint attacks by learning image-agnostic adversarial perturbations. To achieve this goal, we propose a novel universal adversarial pertur-

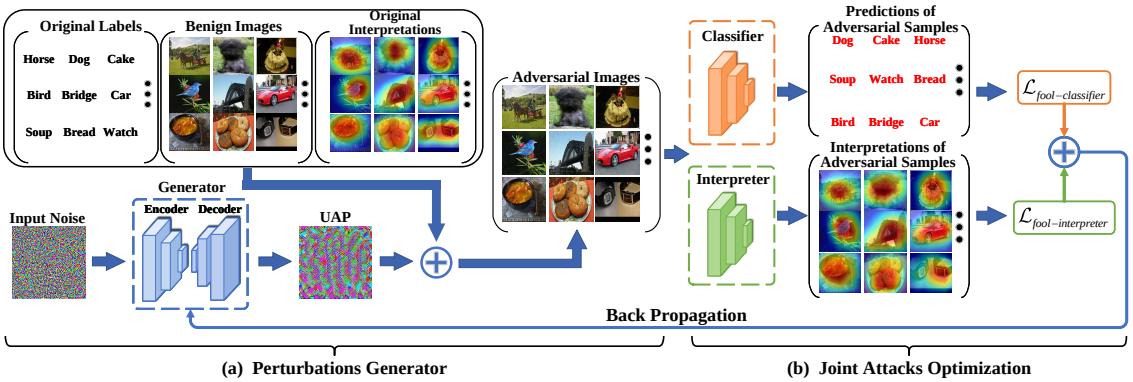


Figure 4.2: The overall framework of the proposed method (JUAP), consisting of perturbations generator and joint attacks optimization for fooling DNNs classifier and interpreter simultaneously.

bations framework (**JUAP**) to jointly attack the DNNs classifier and interpreter. More specifically, a generative adversarial network is proposed to learn universal perturbations that cause natural samples to be misclassified with high confidence while keeping their attribution maps unchanged. The overall framework of the proposed method JUAP is shown in Fig. 4.2, which consists of perturbations generator and joint attacks optimization for fooling DNNs classifier and interpreter simultaneously.

4.4.2 Attack DNNs Classifiers

A general solution to generate universal adversarial perturbations is to leverage an iterative gradient-based strategy [126, 32]. However, iteratively creating perturbations for each sample and fusing all perturbations to get the UAP is time-consuming. Moreover, UAP created by data-driven iterative gradient-based methods significantly depends on the training set. Thus, in this work, in order to attack DNNs classifiers, a generative adversarial framework is proposed to learn universal adversarial perturbations $\hat{n}$, which can be formulated as an image-to-image translation task by reconstructing different noise vectors, consisting of an encoder $e(\cdot)$ and a decoder

$d(\cdot)$. Given various initial input noise, the output perturbations are different from each other. Meanwhile, this generative model can generate infinite UAPs based on a specific training set. To accelerate the convergence speed, we randomly sample a noise vector from the uniform distribution $U(0,1)$, denoted as n, with the same size as the input images. More specifically, a generator $\mathcal{G}_\theta$ with weights θ is developed to map the random noise image to joint universal adversarial perturbations $\hat{n}$, which can be modeled as follows:

$$\hat{n} = \mathcal{G}_\theta(n) = d(e(n)), n \sim U(0,1), \tag{4.5}$$

where the encoder extracts the latent variables (or semantic features) of the inputs, while the decoder receives the encoders' output and reconstructs the latent variables to implement the image-to-image translation. More details about generator can be found in Section 4.5.2.

To make the JUAP imperceptible, the output of the generator is scaled to have a fixed magnitude. Mathematically, the JUAP generation process can be defined as follows:

$$\hat{n} = \min\left(1, \frac{\zeta}{\|\mathcal{G}_\theta(n)\|_p}\right) \cdot \mathcal{G}_\theta(n), n \sim U(0,1), \tag{4.6}$$

where ζ is a pre-defined threshold to control the magnitude of the JUAP. $\|\cdot\|_p$ is the p-norm of the vector. After that, the adversarial examples can be obtained by adding the joint universal perturbation to benign images, i.e., $\hat{x} = x + \hat{n}$.

To optimize the parameters θ of the generator $\mathcal{G}$ for attacking DNNs classifiers, we need to specify a classification objective to optimize. More specifically, with different attacking goals, we can define three types of objectives to learn generator $\mathcal{G}_\theta$'s parameters as follows:

- One of the representative attacking goals is to perform *non-targeted attack*. The most general practice for the image classification task is to minimize the cross-entropy loss between predictions and true labels to guide the parameters'

optimization for the exact classification. Inspired by this paradigm, we propose to maximize the cross-entropy loss for the non-targeted attack as follows:

$$\min_{\theta} \mathcal{L}_{fool-classifier} = \min_{\theta} \; -\log(\mathcal{L}\left(f\left(\hat{x}\right), C_x\right)), \tag{4.7}$$

where $\mathcal{L}$ is the cross-entropy loss.

- Instead of maximizing the cross-entropy loss, other alternatives can also be exploited to deceive classifiers for *non-targeted attack*. The most straightforward method is to assign a wrong label for samples at each iteration and minimize the cross-entropy loss to attack classifiers. To achieve a great performance, we set the least likely class as the target label as follows:

$$\min_{\theta} \mathcal{L}_{fool-classifier} = \min_{\theta} \; \log\left(\mathcal{L}\left(f\left(\hat{x}\right), C_{\min}\right)\right), \tag{4.8}$$

where C_{min} is the class with the minimum prediction probability, i.e., $C_{min} = \arg\min(f(x))$.

- Next, we can consider a more difficult task: *targeted attack*. In such a scenario, the goal of universal perturbations is to not only mislead the classifiers but also change the predictions to a specific target label C_t. To achieve the targeted attack, we can directly minimize the cross-entropy loss toward a target label C_t:

$$\min_{\theta} \mathcal{L}_{fool-classifier} = \min_{\theta} \; \log\left(\mathcal{L}\left(f\left(\hat{x}\right), C_t\right)\right). \tag{4.9}$$

4.4.3 Attack DNNs Interpreters

Fooling interpreters aims to minimize the discrepancy of interpretation maps between benign and adversarial examples. To demonstrate the effectiveness of the proposed methods, we focus on adversarial attacks on three representative interpretation methods, i.e., CAM [226], GradCAM [156], and RTS [31], where they have different underlying mechanisms. CAM [226] is the representative activation-based interpreter,

which has been widely used in various computer vision tasks. Using GradCAM [156] as the interpreter aims to demonstrate the effectiveness of JUAP in manipulating attribution maps based on gradients. Here typical gradient-based interpreters [163] are not taken into account since the attribution maps produced by these methods are not sufficiently clear/distinguishable. RTS [31] is employed as a representation of methods that do not rely on the intrinsic model parameters (e.g., activations or gradients) for generating the attribution maps. Mathematically, the aforementioned interpreter methods are briefly summarised as follows:

- CAM is the first activation-based method, which is easily embedded into most existing DNNs. Let A_{ij}^k denote the element in the i-th row and j-th column of the k-th channel of the last convolutional layer. The output of the following global average pooling layer is $A^k = \frac{1}{V}\sum_i \sum_j A_{ij}^k$, where V is the number of the latent variables in the last convolutional layer. ω_c^k denotes the weight in final dense layer between A^k and the logits of class c. The attribution map for class c is defined as $\mathcal{I}_c$, where each spatial element is given by

$$\mathcal{I}_c(i,j) = \sum_k \omega_c^k A_{ij}^k. \tag{4.10}$$

- GradCAM is one of the variants of CAM, which takes the gradient into consideration. The gradient is used to compute the weight ω_c^k, i.e.,

$$\omega_c^k = \frac{1}{V}\sum\nolimits_i \sum\nolimits_j \frac{\partial f_c}{\partial A_{ij}^k}, \tag{4.11}$$

  where f_c is the c-th output of the classifier. The attribution maps are computed by Eq (4.10).

- RTS is the representative model-based method, which trains a model to create attribution maps. Let R_{ϖ} denote the interpreter with weights ϖ. The attribution maps are created by the interpreter, i.e., $\mathcal{I} = R_{\varpi}(x)$. To optimize the

parameters of RTS interpreter, the objective function is formulated as:

$$\begin{aligned}\mathcal{L}_{rts} =& \lambda_1 \mathcal{L}_{tv}(\mathcal{I}) + \lambda_2 \mathcal{L}_{av}(\mathcal{I}) \\ & - \log\left(f_c(\phi(x, \mathcal{I}))\right) + \lambda_3 f_c(\phi(x, 1-\mathcal{I}))^{\lambda_4},\end{aligned} \tag{4.12}$$

where $\mathcal{L}_{tv}(\mathcal{I})$ and $\mathcal{L}_{av}(\mathcal{I})$ are the total variation and the average value of the attribution map. These two items aim to reduce the noise and retain the most informative part of the attribution map. ϕ uses the interpretation $\mathcal{I}$ as the mask and blends the benign image x with noise. The last two items encourage the interpreter to capture the most informative part of the attribution map.

In order to deceive DNNs' interpreters, we aim to minimize the interpretation shift of interpretation maps between benign and adversarial examples. There are multiple alternatives, e.g., Mean Square Error, Kullback-Leibler Divergence, Maximum Mean Discrepancy, etc. In this paper, we use the L_2 measure, which is easy to implement without additional computation cost. The objective is shown as follows:

$$\min_{\theta} \mathcal{L}_{fool-interpreter} = \min_{\theta} \left\| \mathcal{I}\left(\hat{x}\right) - \mathcal{I}\left(x\right) \right\|_2 , \tag{4.13}$$

$$\mathcal{I}(i,j)\left(\cdot\right) = \begin{cases} \sum\limits_k \omega_c^k \cdot A_{ij}^k, \text{ if CAM} , \\ \sum\limits_k \left(\frac{1}{V} \sum_i \sum_j \frac{\partial f_c}{\partial A_{ij}^k} \right) \cdot A_{ij}^k, \text{ if GradCAM} , \\ R_{\varpi}\left(\cdot\right), \text{ if RTS} . \end{cases} \tag{4.14}$$

Optimization. During deceiving interpreters, it is necessary to ensure that the second-order derivative of generators is not all-zero, especially for the interpreters using gradients. The ReLU activation function is widely used in various DNNs, which consists of two linear components. The second-order derivative of ReLU is all-zero and brings difficulties in performing back-propagation during gradient descent. To tackle this optimization issue, we propose using a smoothing mechanism on the gradient of ReLU for minimizing Eq. (4.13) as follows:

$$a\left(s\right) = \mathbb{1}\left(s < 0\right) \cdot \left(1 + s/\sqrt{s^2 + \tau}\right) + \mathbb{1}\left(s \geq 0\right) \cdot s/\sqrt{s^2 + \tau}, \tag{4.15}$$

where $\mathbb{1}(\cdot)$ is the indicator function, and $\tau = 10^{-4}$ is a small constant to ensure that the output of the function is bounded. s is the input of the ReLU activation layer.

4.4.4 Universal Adversarial Perturbations for Joint Attack

After introducing the DNNs classifier attack and interpreters attack, we finally design the joint attack to generate UAPs for attacking the classifier and its interpreter simultaneously. The most straightforward approach for the joint attack is to minimize the aforementioned loss functions simultaneously for optimizing generator's parameters. However, deceiving both classifiers and interpreters still face a tremendous challenge. That is, the objective of changing predictions usually contradicts the objective of keeping interpretations unchanged. That is because to change the classifier output more significantly, the perturbations tend to be added to the most informative parts in an image, which are also the regions easily detected by the interpreter.

To address the challenge, we propose an iterative optimization strategy for optimizing the objectives of classifier and interpreter attacks. More specifically, at the beginning of the generator training, we conduct joint optimization on the two objectives. When the perturbations produced by the generator are able to deceive the classifier, we focus on fooling its coupled interpreter without deceiving classifiers. The overall objective is defined as follows:

$$\mathcal{L}_{JUAP} = \max\{\mathcal{L}_{fool-classifier}, \delta\} + \lambda \cdot \mathcal{L}_{fool-interpreter}, \tag{4.16}$$

where λ is a hyperparameter to control the attacking effect between the classifier and interpreter, and δ is exploited to decide whether to fool classifiers. In the following section, we conduct comprehensive experiments and demonstrate that the proposed method is robust to different loss functions defined in Eq. (4.7)-(4.9). The overall training process of the proposed method JUAP is shown in Algorithm 2.

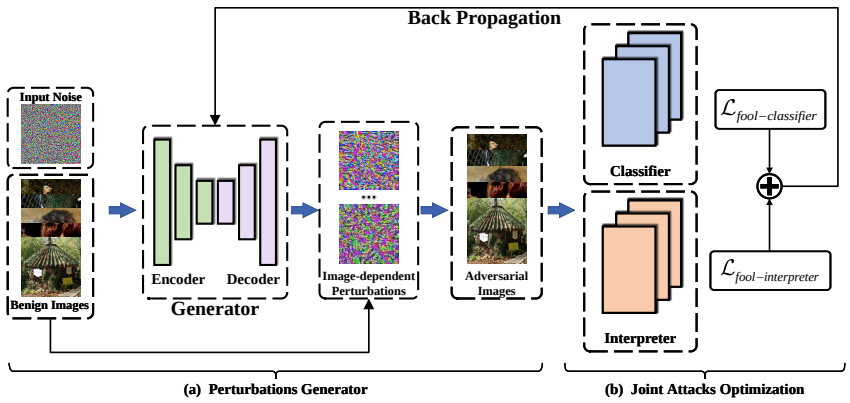


Figure 4.3: The overall framework of the proposed method on for generating joint image-dependent perturbations.

4.4.5 JUAP for Image-dependent Perturbations

Although our proposed method is designed for generating image-agnostic perturbations, the proposed framework has strong scalability and can be easily generalized to produce image-dependent perturbations. To achieve the goal, instead of using multiple samples to learn universal adversarial perturbations, image-dependent perturbations are devised for each sample. That is, for each sample, the parameters of the generator vary and are updated by backpropagation for learning image-dependent perturbations. The production of image-dependent perturbations is defined by:

$$\hat{n}_i = \min\left(1, \frac{\zeta}{\|\mathcal{G}_{\theta_i}(n)\|_p}\right) \cdot \mathcal{G}_{\theta_i}(n), n \in [0,1]^d. \tag{4.17}$$

The overall framework for generating image-dependent perturbations can be found in Fig. 4.3. Note that we follow the same optimization objective and strategy as JUAP in the image-dependent scenario. We also conduct comprehensive experiments to evaluate the effectiveness of our proposed method for generating image-dependent perturbations in Section 4.5.5. It should be noted that JUAP and the related study called ADV^2 [218] have different focuses, with JUAP being primarily used for generating *universal adversarial perturbations (UAP)* while ADV^2 for *generating image-*

Algorithm 2: Our proposed method **JUAP**

Input:

Benign samples x, generator $\mathcal{G}_\theta$, classifier f, interpreter $\mathcal{I}$, vector norm p, threshold ζ, random noise n, iteration T.

Output: Joint universal adversarial perturbation $\hat{n}$.

Procedure:

for *t in 1:T* **do**

  Reconstruct and scale the random noise n to get JUAP $\hat{n}$: $\hat{n} = \min\left(1, \frac{\zeta}{\|\mathcal{G}_\theta(n)\|_p}\right) \cdot \mathcal{G}_\theta(n)$;

  Create adversarial examples $\hat{x}$: $\hat{x} = x + \hat{n}$;

  Compute the loss function by using one of objectives for attacking classifiers in Eqs. (4.7)-(4.9) ;

  Compute the interpretation shift by Eq. (4.13) ;

  Compute the overall objective $\mathcal{L}_{JUAP}$ by Eq. (4.16);

  Update parameters θ of the generator: $\theta \leftarrow \theta - \gamma \cdot \nabla_\theta \mathcal{L}_{JUAP}$;

end for

dependent perturbations (IDP). Therefore, we only verify the potential of JUAP in generating IDPs to deceive both classifiers and interpreters without comparing JAP with ADV^2, which is unfair and meaningless.

4.5 Experiments

4.5.1 Datasets

In our experiment, three representative image classification datasets (i.e., ImageNet, Chest X-Ray dataset, and Intel Image Classification dataset) are used to validate the effectiveness of the proposed method. The details are shown as follows:

- **ImageNet** [36] is the most widely used image classification dataset, which contains more than 12 million training images, 50,000 validation images, and 100,000 test images with 1,000 classes. We randomly sample 20% validation images (10,000 images) as training and test sets, respectively.

- **Chest X-Ray Dataset** [84] contains 5,863 chest X-ray images from children with two classes (i.e., Pneumonia/Normal). We use the training and test sets to evaluate the performance of the proposed framework.

- **Intel Image Classification Dataset** [1] is initially published by Intel to host an Image classification Challenge. It contains around 25,000 images under 6 classes. We randomly choose 20% training images as the training and test sets, respectively.

4.5.2 Implementation details

Classifiers

Two representative deep neural networks are used as the classifiers, i.e., ResNet18 [61] and DenseNet121 [72]. These two DNNs vary in capacities and architectures, which factors out the influence of different classifiers and demonstrates the robustness of the proposed method. We directly use the pre-trained models as the classifiers when constructing experiments on the ImageNet dataset. For other datasets, we fine-tune the pre-trained models as the target classifiers. The Top-1 accuracy attained on different datasets is summarised in Tab. 4.1.

[1] https://www.kaggle.com/datasets/puneet6060/intel-image-classification

Table 4.1: Top-1 accuracy on different datasets.

Datasets		**ImageNet**	**X-Ray**	**Intel**
DNNs Models	**ResNet18**	67.340	87.821	90.100
	DenseNet121	72.100	91.506	88.604

Universal Adversarial Perturbations Generators

In order to evaluate the robustness of our proposed method JUAP, in this paper, we adopt two representative architectures to generate universal adversarial perturbations.

- **ResNet generator** [79] consists of several residual blocks proposed in [61] for reconstructing the input noise. *Residual connections* make the output image share structure with the input image, which is suitable for the image-to-image translation task.
- **U-Net generator** [150] is comprised of five convolutional layers and symmetrical upsampling operators. The skip connections between the convolutional layer and its symmetrical upsampling operator increase the resolution of the output, which is widely used in image translation and semantic segmentation tasks.

Baselines

To better demonstrate the superiority of the proposed method, we use several representative methods as baselines for comparison. The details about these baselines and variants of the proposed method are summarised as follows:

- UAP [126]: This is the first iterative method for generating universal adversarial perturbations.
- GUAP [142]: This method is a typical generative method for creating universal

adversarial perturbations.

- GUAP-C_t [142]: This method is used to generate UAPs for the targeted attack.
- PGD [121]: This method is used as the representative method to create image-dependent perturbations.
- R-JUAP: This is the proposed method that uses the ResNet generator to create JUAPs.
- U-JUAP: This method is proposed to create JUAPs based on the U-Net generator.

Evaluation Metrics

Three evaluation metrics are used to quantitatively evaluate the performance of the proposed method from two aspects, i.e., attacking DNNs classifiers and interpreters. The details are summarised as follows:

- **Fooling Ratio (FR)** [126]. This metric is used to evaluate the effectiveness of fooling classifiers as follows:

$$\text{Fooling Ratio (FR)} = \left(\sum_{i=1}^{N} \mathbb{1} \left(f\left(\hat{x}_i\right) \neq f\left(x_i\right) \right) \right) / N, \tag{4.18}$$

  where $\mathbb{1}(\cdot)$ is the indicator function. The higher FR values indicate better performance for attacking the DNNs classifier.

- $\mathcal{L}_\infty$ **metric (L1)**. As for attacking interpreter, this metric is exploited to assess the discrepancy of attribution maps between benign and adversarial examples, which is computed as follows:

$$\mathcal{L}_1 = \left\| \mathcal{I}\left(\hat{x}\right) - \mathcal{I}\left(x\right) \right\|_1. \tag{4.19}$$

  The lower $\mathcal{L}_\infty$ values indicate better performance for attacking DNNs interpreters.

- **Intersection-Over-Union (IOU) score**. This metric is widely used in the target detection domain. We adopt it to measure the similarity of attribution maps:

$$\mathrm{IOU}\left(\mathcal{I}\left(x\right),\mathcal{I}\left(\hat{x}\right)\right)=\frac{\left|O\left(\mathcal{I}\left(x\right)\right)\cap O\left(\mathcal{I}\left(\hat{x}\right)\right)\right|}{\left|O\left(\mathcal{I}\left(x\right)\right)\cup O\left(\mathcal{I}\left(\hat{x}\right)\right)\right|}, \tag{4.20}$$

  where $O(\cdot)$ is the binarization function. The higher IOU values indicate better performance for attacking interpreters.

Parameter Settings

All experiments are implemented on *PyTorch*, and all baselines are conducted by their open-source implementations. For JUAP, we use Adam the optimizer with a learning rate of 0.0002, and the batch size is set to 30. All images are resized to 224 $\times$ 224 and preprocessed by widely used data augmentation methods, e.g., Random Crop and Random Horizontal Flip. During the inference process, all test images are resized to 224 $\times$ 224 without further preprocessing. In Eq. (4.6), $p = 2$ and $\zeta = 2000$ are set as the default. In the following experiments, we also set $p = \infty$ and $\zeta = 10$ to demonstrate the robustness of the proposed framework. The reason for selecting these two sets of parameters is that the generated UAPs can effectively carry out attacks with imperceptible perturbations [126, 142]. In Eq. (4.16), δ is set to -0.8. Due to the differences of characteristics between multiple interpreters, λ is fine-tuned dynamically. We recommend setting $\lambda \in [0.0001, 0.003]$. During the training process, we mainly use Eq (4.8) as the objective. When deceiving RTS, since the encoder plays a crucial role in creating attribution maps, solely using the aforementioned loss function $\mathcal{L}_{fool-interpreter}$ is not sufficient to manipulate the interpretation. We add an additional loss function to deceive RTS, i.e., $\mathcal{L}_{rts} = \left\|e\left(\hat{x}\right) - e\left(x\right)\right\|_2$, where $e(\cdot)$ is the output of the encoder.

Table 4.2: Fooling ratio on different datasets.

Datasets			ImageNet	X-Ray	Intel
Method	ResNet18	**UAP**	77.64	75.80	51.46
		GUAP	79.91	83.97	50.75
		R-JUAP-CAM	80.35	84.62	55.627
		R-JUAP-GradCAM	80.21	**84.94**	**51.71**
		R-JUAP-RTS	83.88	79.49	50.78
		U-JUAP-CAM	77.35	80.29	54.63
		U-JUAP-GradCAM	84.27	82.53	46.58
		U-JUAP-RTS	**85.00**	83.01	50.32
	DenseNet121	**UAP**	**78.37**	75.16	46.83
		GUAP	77.01	79.17	53.10
		R-JUAP-CAM	76.60	**83.81**	47.54
		R-JUAP-GradCAM	77.54	81.09	50.57
		R-JUAP-RTS	77.31	76.12	52.99
		U-JUAP-CAM	77.17	79.81	45.01
		U-JUAP-GradCAM	78.10	78.53	51.78
		U-JUAP-RTS	76.12	79.81	**54.06**

4.5.3 Effectiveness of Attacking Classifiers

We first evaluate the effectiveness of JUAP for attacking target classifiers. We summarise the fooling ratio for different target classifiers on three benchmark datasets, and the results are shown in Tab. 4.2. We only report one result for UAP and GUAP since interpreters do not affect their fooling ratios. We use R-JUAP-CAM (or R-JUAP-GradCAM, and R-JUAP-RTS) and U-JUAP-CAM (or U-JUAP-GradCAM, and U-JUAP-RTS) to represent the JUAP generated by the ResNet generator and U-Net generator, respectively, where the interpreter is CAM (or GradCAM, and RTS). From the results, we have the following observations:

- Compared with other baselines, it is observed that both R-JUAP and U-JUAP achieve a high fooling ratio across different target classifiers, which demonstrates

Table 4.3: Interpretation discrepancy between adversarial and benign attribution maps (Model: DenseNet121).

Datasets		**ImageNet**		**X-Ray**		**Intel**	
Metrics		**L1**	**IOU**	**L1**	**IOU**	**L1**	**IOU**
Method	**UAP-CAM**	0.23	0.52	0.28	0.45	0.25	0.51
	GUAP-CAM	0.18	0.59	0.25	0.46	0.20	0.53
	U-JUAP-CAM	0.17	0.63	0.22	0.53	0.18	0.59
	R-JUAP-CAM	**0.17**	**0.63**	**0.21**	**0.54**	**0.18**	**0.61**
	UAP-GradCAM	0.21	0.36	0.28	0.44	0.23	0.55
	GUAP-GradCAM	0.18	0.46	0.25	0.47	0.20	0.53
	U-JUAP-GradCAM	0.18	0.49	0.21	0.54	0.20	0.55
	R-JUAP-GradCAM	**0.17**	**0.51**	**0.21**	**0.54**	**0.20**	**0.57**
	UAP-RTS	0.10	0.68	0.13	0.72	0.13	0.61
	GUAP-RTS	0.14	0.60	0.14	0.71	0.20	0.54
	U-JUAP-RTS	0.08	0.72	0.08	0.81	0.12	0.64
	R-JUAP-RTS	**0.07**	**0.74**	**0.06**	**0.85**	**0.08**	**0.71**

the effectiveness of JUAP for fooling classifiers. Although the optimization objective is complicated, it is still possible to find the optimal solution.

- R-JUAP and U-JUAP achieve a similar fooling ratio, which indicates that the proposed framework is robust to the architecture of the generator. Meanwhile, in most cases, R-JUAP and U-JUAP with different interpreters outperform the other baselines in the fooling ratio, which indicates that the proposed framework successfully resolves the dilemma between attacking the classifier and its coupled interpreters.

Table 4.4: Detection Rate against an interpreter-based detector.

Target Model	Method	X-Ensemble
ResNet18	**UAP**	71.65
	GUAP	76.53
	U-JUAP	18.57
	R-JUAP	**17.14**
DenseNet121	**UAP**	72.54
	GUAP	83.60
	U-JUAP	25.72
	R-JUAP	**20.33**

4.5.4 Effectiveness of Attacking Interpreters

In this section, we evaluate the effectiveness of JUAP for deceiving interpreters, i.e., make the attributions maps of benign and adversarial examples similar. We summarise the $\mathcal{L}_\infty$ measure and IOU score of adversarial attribution maps with respect to benign maps on three datasets. The results are shown in Tab. 4.3, Fig. 4.4, and Fig. 4.5, respectively. We get the following insightful observations from the experimental results.

- We first visualize some examples for qualitative comparison. Fig. 4.4 shows some attributions maps of benign and adversarial images (produced by UAP, GUAP, U-JUAP, and R-JUAP) with respect to different interpreters (i.e., CAM, Grad-CAM, and RTS). We observe that attribution maps of U-JUAP and R-JUAP are similar with that of the benign images with respect to various interpreters. As for the other baselines, the differences of attribution maps are large, which makes the attack perceptible.

- Next we use $\mathcal{L}_\infty$ measure and IOU score for quantitative comparison. As shown in Tab. 4.3 and Fig. 4.5, both R-JUAP and U-JUAP achieve high IOU scores

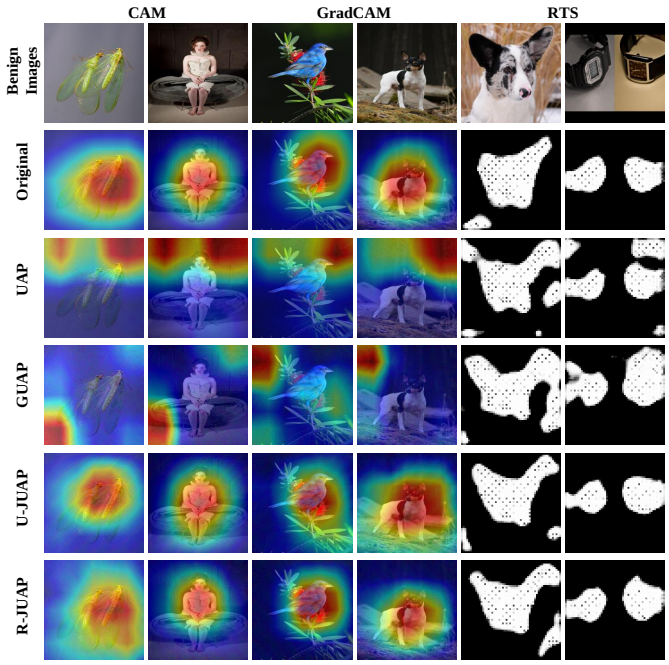
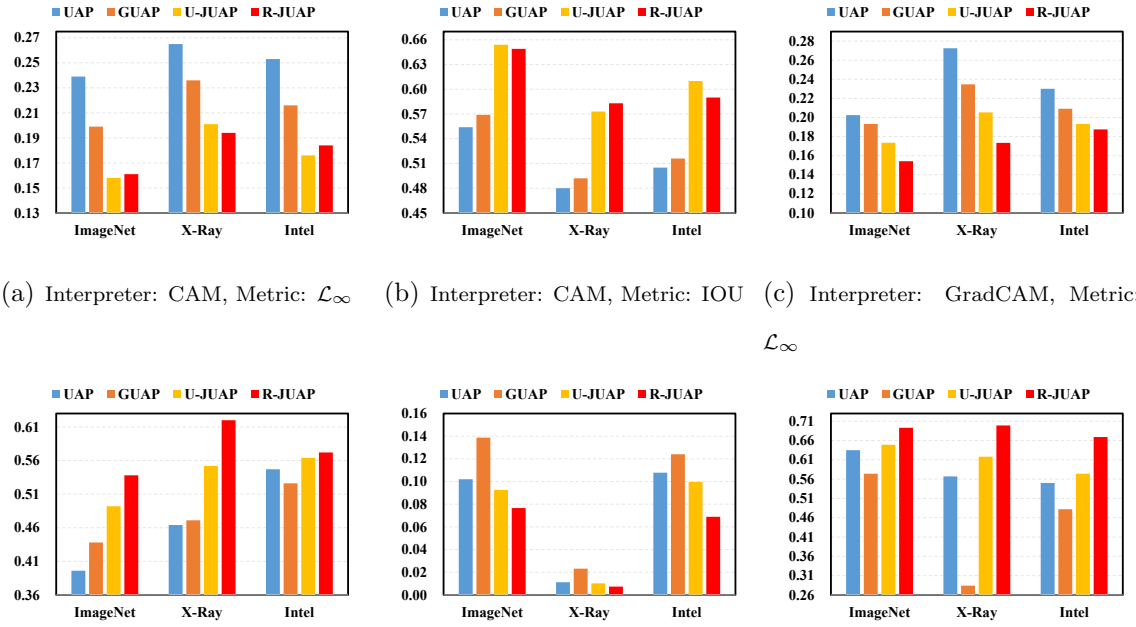


Figure 4.4: Attribution maps of benign images and adversarial images with different interpreters on ResNet18.

with small $\mathcal{L}_\infty$ measure, indicating that the adversarial attribution maps of JUAP are perceptually indistinguishable from their benign counterparts. From these comprehensive experiments, we sufficiently demonstrate the effectiveness of JUAP in deceiving both classifiers and interpreters.

- In addition, we test our R-JUAP and U-JUAP against an interpreter-based detector called X-Ensemble [183]. X-Ensemble utilizes an ensemble of interpreters to generate a couple of saliency maps for an image, and then trains CNN-based binary classifiers accordingly and an random forest to detect adversarial examples. The results are shown in Tab. 4.4. We observe that the detection rate of JUAP against is significantly lower than that of UAP and GUAP, which indicates that JUAP can effectively evade the interpreter-based detector.



(a) Interpreter: CAM, Metric: $\mathcal{L}_\infty$ (b) Interpreter: CAM, Metric: IOU (c) Interpreter: GradCAM, Metric: $\mathcal{L}_\infty$

(d) Interpreter: GradCAM, Metric: IOU (e) Interpreter: RTS, Metric: $\mathcal{L}_\infty$ (f) Interpreter: RTS, Metric: IOU

Figure 4.5: Interpretation discrepancy of attribution maps between benign and adversarial examples (Model: ResNet18).

4.5.5 Effectiveness of Generating IDPs

As we mentioned before, the proposed framework is equipped with the ability to produce image-dependent perturbations for malicious attacks. In this subsection, we use the most representative attacker, PGD [121], as the baseline for comparison, and GradCAM is exploited as the basic interpreter. We first illustrate some examples for qualitative comparison. As shown in Fig. 4.6, attribution maps of samples added joint adversarial perturbations almost remain unchanged. The quantitative results are summarized in Tab. 4.5. We can observe that JAP achieves the high fooling ratio and IOU with the small $\mathcal{L}_\infty$ measure, which further quantitatively demonstrates the above conclusion.

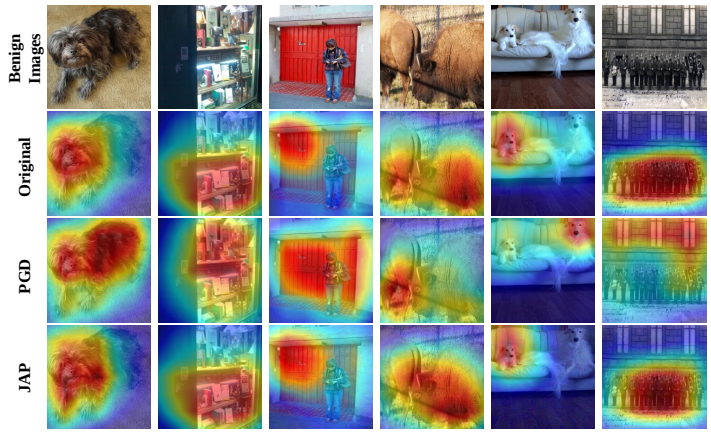


Figure 4.6: Attribution maps of benign images and adversarial images based on the GradCAM interpreter (Model: ResNet18).

4.5.6 Model Analysis

In this section, we aim to investigate the impact of model components and model hyper-parameters in our proposed framework.

L_p Norm Attacks

In addition to $p = 2$ with threshold $\zeta = 2000$ in Eq. (4.6), here we investigate how $p = \infty$ with threshold $\zeta = 10$ affects the attacking performance. The results are summarized in Tab. 4.6. We observe that the three metrics (i.e., fooling ratio, $\mathcal{L}_\infty$

Table 4.5: Results of PGD and JAP on ImageNet.

Target Model	Method	FR	L1	IOU
ResNet18	**PGD**	100.00	36.70	0.55
	JAP	100.00	**27.15**	**0.65**
DenseNet121	**PGD**	100.00	34.54	0.57
	JAP	100.00	**26.42**	**0.66**

Table 4.6: Fooling ratio on different datasets.

Models		**ResNet18**			**DenseNet121**		
Method		**FR**	**L1**	**IOU**	**FR**	**L1**	**IOU**
L_∞**-10**	**R-JUAP-CAM**	85.32	0.12	0.64	75.50	0.13	0.61
	R-JUAP-GradCAM	78.39	0.15	0.66	78.88	0.17	0.61
	R-JUAP-RTS	80.38	**0.08**	**0.69**	76.69	**0.07**	**0.73**
	U-JUAP-CAM	81.69	0.16	0.64	78.37	0.17	0.62
	U-JUAP-GradCAM	**86.30**	0.18	0.49	**79.15**	0.18	0.47
	U-JUAP-RTS	85.44	0.09	0.64	79.06	0.09	0.68
L_2**-2000**	**R-JUAP-CAM**	80.35	0.16	0.65	76.60	0.17	0.63
	R-JUAP-GradCAM	80.21	0.15	0.54	77.54	0.17	0.51
	R-JUAP-RTS	83.88	**0.08**	**0.69**	77.31	**0.07**	**0.74**
	U-JUAP-CAM	77.35	0.15	0.65	77.17	0.17	0.63
	U-JUAP-GradCAM	84.27	0.17	0.49	**78.10**	0.18	0.49
	U-JUAP-RTS	**85.00**	0.09	0.65	76.12	0.08	0.72

measure, and IOU score) fluctuate in a small range, which indicates that our JUAP creation is robust under different threat models.

Table 4.7: Results of JUAP with different loss functions.

Method		**FR**	**L1**	**IOU**
Non-targeted	**R-JUAP-**C_{min}	80.35	0.16	0.65
	R-JUAP-CE	**89.63**	0.16	**0.65**
Targeted	**GUAP-**C_t	73.44	0.24	0.49
	R-JUAP-C_t	**77.46**	**0.22**	**0.53**

Objective on Attacking DNNs Classifier

In Section 4.4.2, we introduce three kinds of objectives to optimize the parameters of the generator. Here, we study the impact of these different objectives on the proposed framework. The coupled interpreter is CAM. We use R-JUAP-CE, R-JUAP-C_{min},

and R-JUAP-C_t to denote the JUAP optimized by Eqs. (4.7)-(4.9), respectively. The results are shown in Tab. 4.7. For the non-targeted attack, we can see that using Eq. (4.7) and Eq. (4.8) as optimization objectives can achieve similar results. For the targeted attack, GUAP-C_t is used as the baseline since it can also produce targeted universal perturbations. As shown in Tab. 4.7, R-JUAP-C_t outperforms GUAP in $\mathcal{L}_\infty$ and IOU scores with a similar fooling ratio.

4.6 Conclusion

In this work, we first find that DNNs models' interpretations can be used to inspect the problematic outputs from adversarial examples caused by universal adversarial perturbations. This finding motivates us to further raise a new research problem: *whether there exist universal adversarial perturbations that can perform a joint attack to attack DNNs classifier and its interpretation simultaneously with malicious desires?* To address this problem, we propose a generative framework to create the joint universal adversarial perturbations to jointly attack the DNNs classifier and its coupled interpreters. Our comprehensive experiments validate the effectiveness of the proposed JUAP on three real-world datasets. Considering the vulnerability of DNNs interpreters, we would like to investigate a new interpretation method to defend against adversarial attacks for trustworthy DNNs in the future.

Chapter 5

SemDiff: Generating Natural Unrestricted Adversarial Examples via Semantic Attributes Optimization in Diffusion Models

Unrestricted adversarial examples (UAEs), allow the attacker to create non-constrained adversarial examples without given clean samples, posing a severe threat to the safety of deep learning models. Recent works utilize diffusion models to generate UAEs. However, these UAEs often lack naturalness and imperceptibility due to simply optimizing in intermediate latent noises. In light of this, we propose SemDiff, a novel unrestricted adversarial attack that explores the semantic latent space of diffusion models for meaningful attributes, and devises a multi-attributes optimization approach to ensure attack success while maintaining the naturalness and imperceptibility of generated UAEs. We perform extensive experiments on four tasks on three high-resolution datasets, including CelebA-HQ, AFHQ and ImageNet. The results demonstrate that SemDiff outperforms state-of-the-art methods in terms of attack

success rate and imperceptibility. The generated UAEs are natural and exhibit semantically meaningful changes, in accord with the attributes' weights. In addition, SemDiff is found capable of evading different defenses, which further validates its effectiveness and threatening. We believe our work can shed light on further exploration of the vulnerability of deep neural networks.

5.1 Introduction

Deep Neural Networks (DNNs) have achieved significant success in wide range of applications, such as image classification [36], face recognition [141], social recommendation [193], and machine translation [6]. However, a lot of research finds that deep learning models are vulnerable to adversarial attacks [173, 56, 34, 132, 133, 194, 110, 166, 85, 216, 185, 196, 33, 24]. Traditional adversarial attacks try to generate adversarial examples to fool DNNs into wrong predictions by injecting imperceptible perturbations into clean samples [173, 56, 34, 132]. This type of adversarial examples is called as perturbation-based adversarial examples [166], which pose a constraint on the perturbation magnitude [56, 34, 132].

Different from perturbation-based adversarial examples, unrestricted adversarial examples (UAEs) have no common perturbation constraints and do not depend on any given clean images, hence are more threatening to the safety of deep learning models [166, 85, 216, 185, 196, 33, 24]. Specifically, common unrestricted adversarial attacks use generative models to generate UAEs by optimizing the random noise (e.g., the input noise vector of Generative Adversarial Networks (GANs) [166, 85, 76] or the intermediate latent noise of diffusion models [33, 24]). Since the random noise could refer to unseen samples in the dataset and there is no limitation in the L_p norm spaces from existing data samples, UAEs are more flexible and aggressive than perturbation-based adversarial examples [196]. Moreover, attackers have no need to collect clean samples as the inputs in advance, and can generate an infinite number

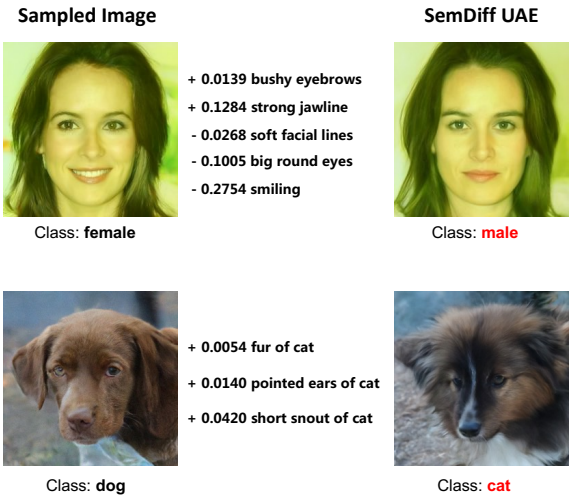


Figure 5.1: **Some UAEs generated by the proposed SemDiff.** Note that the sampled images are not real-world images, but are generated by diffusion models with randomly sampled noise images. Our SemDiff optimizes the weights of multiple meaningful attributes to craft adversarial examples based on the sampled images. It is shown that the weights are in accord with the semantic changes in images, even the negative weight can lead to an opposite but meaningful change.

of UAEs with a trained generative model [33]. This is particularly important in some data-expensive attack scenarios, such as medical images [74], where collecting clean samples is difficult and expensive.

Previous UAE works utilize GANs to generate UAEs [166, 85, 216, 185, 196], but they show unsatisfactory visual quality on high-resolution datasets due to GAN's bad interpretability and unstable training [33]. Instead, recent works [33, 24] adopt diffusion models, which outperform GAN-based models in terms of image synthesis quality and diversity in recent years [37, 148, 122]. However, the generated UAEs of these works are still visually unnatural, exhibiting local visible perturbations (see the examples of AdvDiff [33] and AdvDiffuser [24] in Fig. 5.5 and Fig. 5.6). The root cause is that they directly optimize in the intermediate latent noise space of diffusion models [33, 24]. Unlike GANs, whose input latent space encodes high-level

semantic information, the intermediate latent noise of diffusion models are dominated by stochastic noise [70] and hard to directly capture such semantics [86, 90]. This noise-centric nature forces perturbations to focus on low-level pixel variations rather than semantic attributes. variations rather than semantic attributes. Furthermore, diffusion models are found to construct the high-level context in the early stage and refine details in the later stage of the denoising process [27]. However, because the intermediate latent images in the early steps are too noisy to obtain a accurate gradient, these works [33, 24] primarily perturb in the later steps, which only affect local details. As a result, the generated UAEs fail to achieve semantic coherence and exhibit unnatural local patterns similar to traditional perturbation-based adversarial examples.

In order to address these issues, we propose **SemDiff**, a novel unrestricted adversarial attack that explores the semantic latent space of diffusion models to generate natural and realistic UAEs with semantic attribute changes. Specifically, leveraging the disentanglement property of diffusion models [90], we propose to learn some meaningful but adversarial attributes in the semantic latent space of diffusion models. The semantic latent space lies in the deepest feature maps in the UNet of diffusion models. In contrast to the intermediate latent noise, these bottleneck features encode high-level semantics with minimal spatial redundancy [90]. Then, we create a set of meaningful and class-sensitive attributes for each task, such as *bushy eyebrows* and *strong jawline* for gender classification task on CelebA-HQ dataset. Changing the weight of such attribute in the semantic latent space can lead to the corresponding attribute change in images.

Furthermore, considering a single attribute needs a large weight to successfully fool target models that often results in excessive semantic shifts, we devise a multi-attributes optimization approach. The weights of multiple attributes are jointly optimized together with a weight penalty term, hence slight changes of multiple attributes are sufficient to cause misclassification while keeping UAEs realistic and close to the

sampled images, as exemplified in Fig. 5.1. Compared with current diffusion-based UAE works [33, 24], our SemDiff updates images along the direction of semantic attributes, generating more natural UAEs with imperceptible difference from real-world images.

We summarize our contributions as follows:

- To the best of our knowledge, our work is the first to utilize the semantic latent space of diffusion models to generate UAEs. Previous works only optimize in the intermediate latent noise space, which hinders the naturalness and imperceptibility of UAEs.

- We propose SemDiff, a novel unrestricted adversarial attack that optimizes the weights of multiple meaningful attributes in the semantic latent space of diffusion models, achieving attack success while maintaining the naturalness of generated UAEs.

- We conduct extensive experiments across four tasks on three high-resolution datasets (CelebA-HQ, AFHQ and ImageNet). The results demonstrate that SemDiff outperforms state-of-the-art methods in terms of attack success rate and imperceptibility. The generated UAEs are more natural with semantic attribute changes.

- We empirically show that SemDiff can bypass various defenses (detection-based defenses, preprocessing-based defenses and adversarial training), which further validates the threatening of our attack, and unveils the vulnerability of current deep neural networks and defenses.

5.2 Related Work and Preliminary

In this section, we first review the traditional perturbation-based adversarial examples. Then we discuss the related work on unrestricted adversarial examples, including GAN-based UAEs and diffusion-based UAEs. Finally, we introduce diffusion models and their disentanglement capability.

5.2.1 Perturbation-based Adversarial Examples

Early adversarial attack works focus on perturbation-based adversarial examples, which add a small perturbation into clean samples to fool the target models into incorrect predictions [173, 56, 34, 132, 121, 140, 73, 181]. The perturbation is usually constrained in a L_p norm space to keep imperceptible to human eyes. Perturbation-based adversarial examples can be defined as [166]:

$$\mathcal{A}_p \triangleq \{\tilde{\boldsymbol{x}} \in \mathcal{O} \mid \exists \boldsymbol{x} \in \mathcal{T}, ||\tilde{\boldsymbol{x}} - \boldsymbol{x}||_p \leq \epsilon \wedge \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{o}(\boldsymbol{x}) = \boldsymbol{o}(\tilde{\boldsymbol{x}}) \neq \boldsymbol{f}(\tilde{\boldsymbol{x}})\}, \tag{5.1}$$

where $\mathcal{O}$ is the set of all images that look realistic to humans, $\mathcal{T}$ is a selected test dataset $\mathcal{T} \subset \mathcal{O}$, $||\cdot||_p$ is a L_p norm, such as L_0, L_2 or L_∞. ϵ is a small constraint, $\boldsymbol{f}$ is the target model to be attacked, and $\boldsymbol{o}$ represents an oracle model that provides ground-truth results. A lot of methods have been proposed to generate perturbation-based adversarial examples, such as the fast gradient-sign method (FGSM) [173] with a L_∞ norm constraint, projected gradient descent (PGD) [121] with a L_2 norm constraint, and L_0 norm Jacobian-based Saliency Map Attack (JSMA) [140].

5.2.2 Unrestricted Adversarial Examples

Song et al. [166] first proposed the concept of unrestricted adversarial examples (UAEs), which get rid of the dependence of input images, as long as the generated

UAEs do not change the semantics. The definition of UAEs can be formulated as:

$$\mathcal{A}_u \triangleq \{\tilde{\boldsymbol{x}} \in \mathcal{O} \mid \boldsymbol{o}(\tilde{\boldsymbol{x}}) \neq \boldsymbol{f}(\tilde{\boldsymbol{x}})\}. \tag{5.2}$$

It is clear that the set of UAEs is a superset of that of traditional perturbation-based adversarial examples, namely $\mathcal{A}_p \subset \mathcal{A}_u$. To craft UAEs, current researchers utilize generative models, including GANs and diffusion models.

GAN-based UAEs

One line of works first build a generator of GANs to generate realistic images, and then optimize the input noise vector to update the output to fool the target model [166, 85, 216]. Specifically, Song et al. [166] proposed to use a well-trained AC-GAN [135] to generate UAEs. Starting from a randomly sampled noise vector z^0, the method searches a z by minimizing an adversarial loss to achieve misclassification. In addition, Khoshpasand and Ghorbani [85] trained a generator for each class to improve the transferability of UAEs. Zhang et al. [216] designed a novel augmented triple-GAN and train a MLP to produce the noise vector instead of optimizing it directly. Another line of GAN-based UAEs is end-to-end method, which trains generators to directly output realistic UAEs that can deceive the target model [185, 196] without the optimization of noise vector. However, all of these methods only focus on small datasets, and their UAEs are found low-quality and unnatural on high-resolution datasets such as ImageNet [33].

Diffusion-based UAEs

Inspired by recent advances of diffusion models in image synthesis, some recent works turn to utilize diffusion models to generate UAEs [33, 24]. Chen et al. [24] first proposed AdvDiffuser to generate UAEs with a pre-trained conditional DDPM [37]. Starting from an initial random noise x_T, AdvDiffuser performs PGD to add perturbation into each intermediate latent noise in the reverse process, and obtains the final

output x_0 as the UAE. Furthermore, Dai et al. [33] proposed AdvDiff and added another noise sampling guidance to update the initial noise x_T in each iteration, achieving a higher attack success rate. Although theses diffusion-based methods achieve better performance than GAN-based methods, we find that the generated UAEs are still unnatural with visible and meaningless perturbations as they directly perturb in the intermediate latent noises of diffusion models. In contrast, our proposed SemDiff explores the semantic latent space of diffusion models to generate UAEs with natural and meaningful attribute changes.

5.2.3 Diffusion Models and Disentanglement

Dinh et al. [164] first proposed the concept of diffusion-based generation. After that, Ho et al. [70] proposed denoising diffusion probabilistic models (DDPMs), which significantly improved both the quality and diversity of the generated images, making diffusion models a competitive alternative to traditional GANs. Specifically, the forward process of DDPMs starts from an real image $\boldsymbol{x}_0$ and progressively diffuses it in a sequence of steps $\boldsymbol{x}_{1:T}$ through Gaussian transitions, which is defined as a Makov chain:

$$q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}\right)=\mathcal{N}\left(\boldsymbol{x}_t ; \sqrt{1-\beta_t} \boldsymbol{x}_{t-1}, \beta_t \mathbf{I}\right), \tag{5.3}$$

where $\{\beta\}_{t=1}^{T}$ is the variance schedule. As $T \rightarrow \infty$, $\boldsymbol{x}_T$ approaches an isotropic Gaussian distribution. Then the reverse process gradually denoises the noise and finally recovers the real image:

$$p_\theta\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right)=\mathcal{N}\left(\boldsymbol{x}_{t-1} ; \boldsymbol{\mu}_\theta\left(\boldsymbol{x}_t, t\right), \sigma_t^2 \mathbf{I}\right), \tag{5.4}$$

where σ_t^2 is a variance of the reverse process and is set to β_t. The mean $\boldsymbol{\mu}_\theta$ can be modeled with a noise predictor $\boldsymbol{\epsilon}_t^\theta$:

$$\boldsymbol{x}_{t-1}=\frac{1}{\sqrt{1-\beta_t}}\left(\boldsymbol{x}_t-\frac{\beta_t}{\sqrt{1-\alpha_t}} \boldsymbol{\epsilon}_t^\theta\left(\boldsymbol{x}_t\right)\right)+\sigma_t \boldsymbol{z}_t, \tag{5.5}$$

where $\alpha_t = \prod_{i=1}^{t}(1-\beta_i)$ and $\boldsymbol{z}_t \sim \mathcal{N}(0, \mathbf{I})$. Subsequently, DDPMs learn the data distribution by training $\boldsymbol{\epsilon}_t^\theta$ using the variational lower-bound (VLB).

In addition, Song et al. [165] proposed denoising diffusion implicit model (DDIM), which defines a non-Markovian process and achieves much faster sampling speed with fewer steps:

$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}}\ \mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta(\boldsymbol{x}_t)) + \mathbf{D}_t(\boldsymbol{\epsilon}_t^\theta(\boldsymbol{x}_t)) + \sigma_t \boldsymbol{z}_t, \tag{5.6}$$

where $\mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta(\boldsymbol{x}_t)) = \left(\frac{\boldsymbol{x}_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t^\theta(\boldsymbol{x}_t)}{\sqrt{\alpha_t}}\right)$ denotes the predicted $\boldsymbol{x}_0$, and $\mathbf{D}_t(\boldsymbol{\epsilon}_t^\theta(\boldsymbol{x}_t)) = \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\boldsymbol{\epsilon}_t^\theta(\boldsymbol{x}_t)$ denotes the direction pointing to $\boldsymbol{x}_t$, $\sigma_t \boldsymbol{z}_t$ is a random noise. We further abbreviate $\mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta(\boldsymbol{x}_t))$ as $\mathbf{P}_t$ and $\mathbf{D}_t(\boldsymbol{\epsilon}_t^\theta(\boldsymbol{x}_t))$ as $\mathbf{D}_t$ for simplicity when the context clearly specifies the arguments. $\sigma_t = \eta\sqrt{(1-\alpha_{t-1})/(1-\alpha_t)}\sqrt{1-\alpha_t/\alpha_{t-1}}$, where η is a hyperparameter that controls the variance of the noise, the process becomes DDPM when $\eta = 1$ and DDIM when $\eta = 0$.

Disentanglement capability allows image generative models to disentangle different attributes of the generated images, such as semantic contents and styles, which is crucial for image editing tasks. Different from GANs inherently endowed with strong disentangled semantics in their latent space, diffusion models show degraded performance by simply editing in the intermediate latent noise space [86]. Recent works have explored ways to achieve disentanglement in diffusion models. For instance, Preechakul et al. [143] introduced an additional input to the reverse sampling process. The input is a semantic latent vector from an extra encoder, that is jointly trained with diffusion models. Wu et al. [195] adopted stable diffusion [149] to achieve disentanglement by optimizing the mixing weights of two text embeddings. In addition, Kwon et al. [90] proposed to leverage the deepest feature maps in UNet as the semantic latent space to edit images.

5.3 Methodology

In this section, we first introduce the motivations and overview of the proposed method. Then, we present the proposed adversarial semantic attributes and the multi-attributes optimization approach in detail.

5.3.1 Motivations and Overview

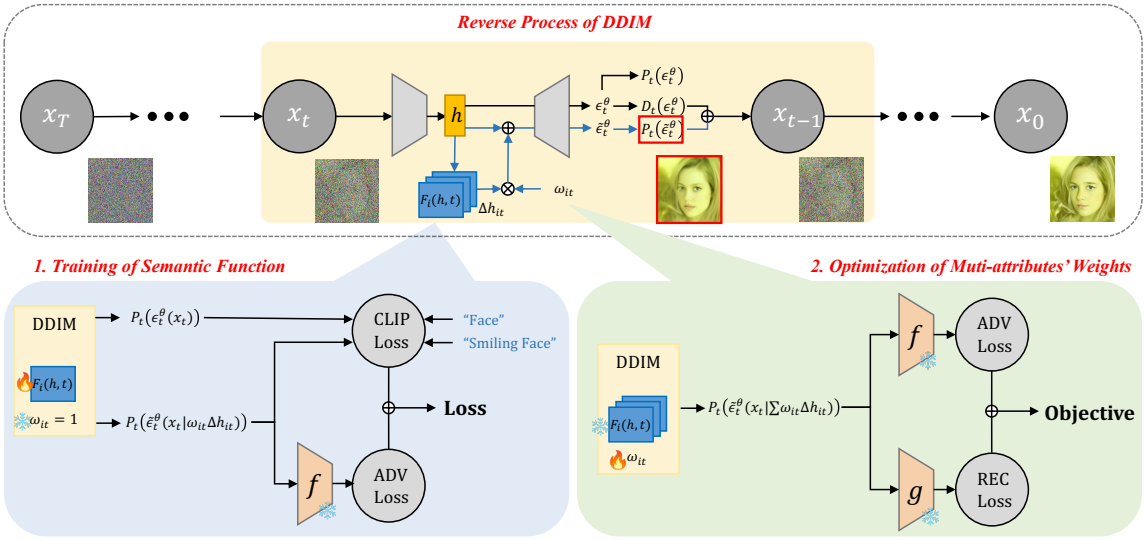


Figure 5.2: **Overview of the proposed SemDiff.** The yellow box illustrates that SemDiff only modifies $\mathbf{P}_t$ (blue lines) while preserving $\mathbf{D}_t$ (black lines) in the reverse process of DDIM to alter the semantics of the generated UAEs. SemDiff first train a semantic function $\mathbf{F}_i(\boldsymbol{h}_t, t)$ to learn each adversarial semantic attribute with the loss shown in the blue box. Then, the weights of multiple attributes $\boldsymbol{w}_{it}$ are optimized with the objective exhibited in the green box. $\boldsymbol{f}$ is the target classifier to be attacked, $\boldsymbol{g}$ is another auxiliary classifier to maintain the class appearance. Notice that we use clearer $\mathbf{P}_t$ (within the red border) instead of $\boldsymbol{x}_t$ at each timestep in both function training and weight optimization.

We aim to design an unrestricted adversarial attack that can generate natural and

realistic adversarial examples to deceive target models without given clean samples. Current state-of-the-art methods [33, 24] utilize diffusion models to generate UAEs, which are still visually unnatural. We find two issues that may degrade the naturalness of generated UAEs. First, these diffusion-based methods directly add perturbations in the intermediate latent noise space, that is noisy and lack high-level semantic information like the input latent space of GANs. Therefore, compared with GAN-based methods and our proposed method, existing diffusion-based methods fail to alter semantic attributes instead introduce perceptible perturbations into sampled images (See Fig. 5.5 and Fig. 5.6), which hinders the naturalness and imperceptibility performance. Our method is motivated to explore a high-level semantic latent space within diffusion models to generate natural UAEs with meaningful attribute changes, achieving better imperceptibility than existing methods.

Second, these diffusion-based methods [33, 24] restrict perturbations to the later stages of the reverse denoising process, as the early-stage intermediate latent images are deemed too noisy for stable gradient computation [33, 24]. This design choice fundamentally conflicts with the hierarchical nature of diffusion models—prior work [27] finds that high-level semantics are established in early denoising steps, while later steps refine low-level details. By perturbing only the later stages, these methods limit their impact to superficial details rather than semantic content, causing generated UAEs to retain unnatural local perturbations similar to traditional perturbation-based adversarial examples (see Fig. 5.5 and Fig. 5.6). To avoid this issue, we propose to optimize in the semantic latent space of diffusion models, and utilize $\mathbf{P}_t$ of DDIM (see Section 5.2.3) instead of the intermediate latent noise $\boldsymbol{x}_t$ to obtain the adversarial gradient. Compared to $\boldsymbol{x}_t$, $\mathbf{P}_t$ denotes the predicted $\boldsymbol{x}_0$ and removes the noise term, thus better approximating the final output $\boldsymbol{x}_0$ and clearer in early stage (see Fig. **??** for example). We use $\mathbf{P}_t$ both in the training of semantic function and the optimization of multi-attributes' weights, making editing image semantics in the early stage of the denoising process possible.

To accomplish our goal, we propose **SemDiff**, a novel unrestricted adversarial attack, with overall framework depicted in Fig. 5.2. In order to generate a natural UAE, our method starts from a randomly sampled noise $\boldsymbol{x}_T$ and iteratively denoises it until the final output $\boldsymbol{x}_0$, adding perturbations in the semantic latent space $\boldsymbol{h}$ of diffusion models. The perturbation is composed of multiple $\boldsymbol{\Delta h}_{it}$ that refer to different adversarial semantic attributes $\boldsymbol{a}_i$ at timestep t, and their corresponding weights $\boldsymbol{w}_{it}$. Therefore, we firstly design a semantic function to learn different semantic attributes but easily deceive target models. Furthermore, we devise a multi-attributes optimization approach to optimize the weights of adversarial semantic attributes, so as to further boost the attack success rate while maintaining the naturalness of generated UAEs.

5.3.2 Adversarial Semantic Attributes

We first introduce adversarial semantic attributes that can generate natural UAEs with semantically meaningful attribute changes but can deceive target models.

Semantic Image Editing with Diffusion Models

For semantic latent manipulation of images $\boldsymbol{x}_0$ given a *pretrained* and *frozen* diffusion model, we choose to shift the predicted noise $\boldsymbol{\epsilon}_t^\theta\left(\boldsymbol{x}_t\right)$ instead of the intermediate latent noise $\boldsymbol{x}_t$ at each sampling step. However, it is found that the intermediate changes in $\mathbf{P}_t$ and $\mathbf{D}_t$ will destruct each other causing the same $p_\theta(\boldsymbol{x}_{0:T})$ [90]. Therefore, we only modify $\mathbf{P}_t$ by shifting $\boldsymbol{\epsilon}_t^\theta(\boldsymbol{x}_t)$ to $\tilde{\boldsymbol{\epsilon}}_t^\theta(\boldsymbol{x}_t)$ while preserving $\mathbf{D}_t$:

$$\boldsymbol{x}_{t-1}=\sqrt{\alpha_{t-1}}\,\mathbf{P}_t(\tilde{\boldsymbol{\epsilon}}_t^\theta\left(\boldsymbol{x}_t\right))+\mathbf{D}_t(\boldsymbol{\epsilon}_t^\theta\left(\boldsymbol{x}_t\right))+\sigma_t\boldsymbol{z}_t. \tag{5.7}$$

Considering that $\boldsymbol{\epsilon}_t^\theta$ is usually implemented as UNet in diffusion models, we follow [90] to modify its deepest feature maps $\boldsymbol{h}_t$ to control $\boldsymbol{\epsilon}_t^\theta$, because $\boldsymbol{h}_t$ has smaller spatial resolutions and high-level semantics than $\boldsymbol{\epsilon}_t^\theta$. The sampling equation accordingly

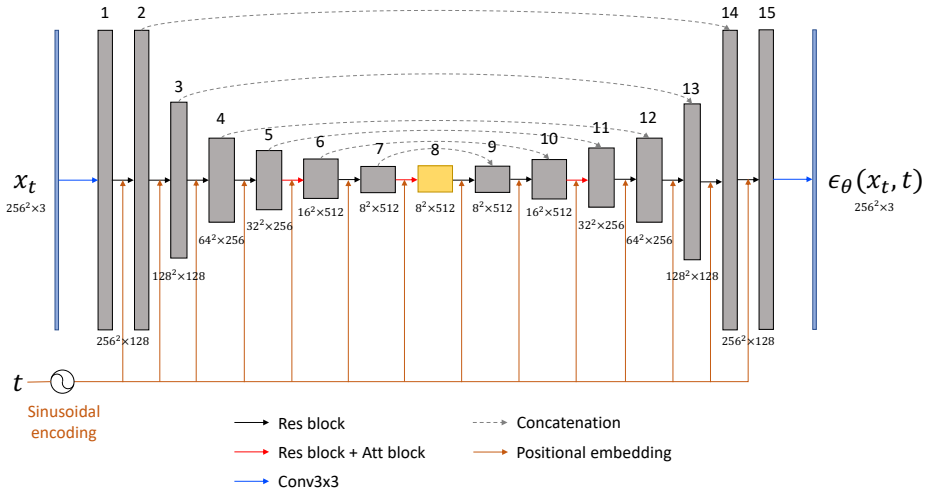


Figure 5.3: Location of the semantic latent space $\boldsymbol{h}_t$. The UNet architecture of diffusion models outputs 256 × 256 images. Each layer is indexed with a number along the operating sequence of the model. The 8th layer is our h-space which is not directly influenced by a skip connection.

becomes:

$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}}\, \mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta\,(\boldsymbol{x}_t|\Delta\boldsymbol{h}_t)) + \mathbf{D}_t(\boldsymbol{\epsilon}_t^\theta\,(\boldsymbol{x}_t)) + \sigma_t \boldsymbol{z}_t, \tag{5.8}$$

where $\boldsymbol{\epsilon}_t^\theta\,(\boldsymbol{x}_t|\Delta\boldsymbol{h}_t)$ adds $\Delta\boldsymbol{h}_t$ to the original feature maps $\boldsymbol{h}_t$. Since directly optimizing $\Delta\boldsymbol{h}_t$ on multiple timesteps is computationally expensive and needs carefully chosen hyperparamenters, we instead use a semantic function $\mathbf{F}(\boldsymbol{h}_t, t)$ to generate $\Delta\boldsymbol{h}_t$ for given $\boldsymbol{h}_t$ and t. $\mathbf{F}$ is implemented as a small neural network with two 1x1 convolutions concatenated with timestep t. See Fig. 5.3 and Fig. 5.4 for the details of $\boldsymbol{h}_t$ and $\mathbf{F}$, respectively.

CLIP Guidance

For training a semantic function $\mathbf{F}$ to learn a specific semantic attribute, we use directional CLIP loss [54] with a pre-trained image encoder E_I and text encoder E_T

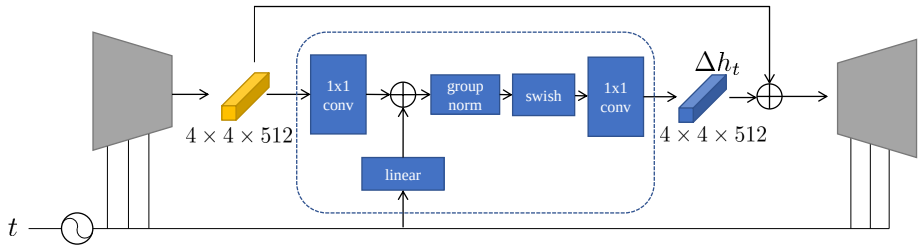


Figure 5.4: Illustration of $\boldsymbol{F}$. It has only two 1x1 convolution layers with 512 channels. We use group norm and swish following DDPM.

of CLIP [144]:

$$\begin{aligned} \mathcal{L}_{\text{direction}}\left(\boldsymbol{x}^{\text{edit}}, c^{\text{target}}; \boldsymbol{x}^{\text{source}}, c^{\text{source}}\right) &= 1 - \frac{\Delta I \cdot \Delta T}{\|\Delta I\|\|\Delta T\|}, \\ \Delta T &= E_T\left(c^{\text{target}}\right) - E_T\left(c^{\text{source}}\right), \\ \Delta I &= E_I\left(\boldsymbol{x}^{\text{edit}}\right) - E_I\left(\boldsymbol{x}^{\text{source}}\right), \end{aligned} \tag{5.9}$$

where $\boldsymbol{x}^{\text{edit}}$ and $\boldsymbol{x}^{\text{source}}$ are the edited and source images, c^{target} and c^{source} are the target and source textual descriptions. Compared to directly minimizing the cosine distance between the edited image $E_I\left(\boldsymbol{x}^{\text{edit}}\right)$ and the target textual description $E_T\left(c^{\text{target}}\right)$, the directional CLIP loss aligns the direction of image transformation with the direction of textual transformation, which preserves image structural consistency and enables more precise attribute editing. In our work, we use the modified $\mathbf{P}_t^{\text{edit}} = \mathbf{P}_t(\tilde{\boldsymbol{\epsilon}}_t^\theta)$ and the original $\mathbf{P}_t^{\text{source}} = \mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta)$ as the edited and source images, and the textual prompts such as 'smiling face' and 'face' for attribute *smiling* as the target and source textual descriptions. We also add a regularization term to keep $\mathbf{P}_t^{\text{edit}}$ close to $\mathbf{P}_t^{\text{source}}$ and the original $\mathbf{P}_t^{\text{source}}$:

$$\mathcal{L}_{\text{CLIP}} = \mathcal{L}_{\text{direction}}\left(\mathbf{P}_t^{\text{edit}}, c^{\text{target}}; \mathbf{P}_t^{\text{source}}, c^{\text{source}}\right) + \lambda_{\text{reg}}|\mathbf{P}_t^{\text{edit}} - \mathbf{P}_t^{\text{source}}|. \tag{5.10}$$

Adversarial Guidance

We also introduce an adversarial guidance to the semantic function $\mathbf{F}$ to learn a semantic attribute but is adversarial to the target model $\boldsymbol{f}$. We focus on targeted

unrestricted adversarial attacks, where the attacker tries to generate an image $\boldsymbol{x}$ that looks like the ground-truth label y_{source} but is misclassified as the target label y_{target} by $\boldsymbol{f}$. Different from previous works [33, 24] feeding the intermediate latent noise $\boldsymbol{x}_t$ to the target model to obtain the adversarial guidance at each timestep, we adopt the modified $\mathbf{P}_t(\tilde{\boldsymbol{\epsilon}}_t^\theta)$ for better approximation of the final output $\boldsymbol{x}_0$:

$$\mathcal{L}_{\text{adv}} = \mathcal{L}_{\text{CE}}\left(\boldsymbol{f}(\mathbf{P}_t(\tilde{\boldsymbol{\epsilon}}_t^\theta(\boldsymbol{x}_t))), y_{\text{target}}\right), \tag{5.11}$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss of the target model.

Hence, the overall loss of the semantic function $\mathbf{F}$ is:

$$\mathcal{L} = \mathcal{L}_{\text{CLIP}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}. \tag{5.12}$$

For each attribute $\boldsymbol{a}_i$, we train a specific semantic function $\mathbf{F}_i$ in the same way, as depicted in Fig. 5.2. The training process is detailed in Algorithm 3.

5.3.3 Multi-attributes Optimization Approach

Given a trained semantic function $\mathbf{F}$, we can generate $\Delta\boldsymbol{h}_t = \boldsymbol{w}_t\mathbf{F}(\boldsymbol{h}_t, t)$ to derive the UAE $\boldsymbol{x}_0$. However, a single attribute usually needs a large weight $\boldsymbol{w}_t$ to successfully deceive target models, which results in unnatural UAEs with excessive semantic shifts. To address this issue, we further propose a multi-attributes optimization approach to optimize the weights of multiple adversarial semantic attributes with a weight penalty term, so that slight changes of multiple attributes are sufficient to cause misclassification while keeping UAEs natural and close to the sampled images.

With a set of attributes $\{\boldsymbol{a}_i\}_{i=1}^M$ and their corresponding semantic functions $\{\mathbf{F}_i\}_{i=1}^M$, we optimize the weights $\boldsymbol{w}_{it}$ of different attributes to minimize the following objective:

$$\begin{aligned}\min_{\boldsymbol{w}_{it}} \mathcal{L}_{\text{CE}}(\boldsymbol{f}(\mathbf{P}_t^{edit}),\ y_{\text{target}}) + \lambda_1\mathcal{L}_{\text{CE}}(\boldsymbol{g}(\mathbf{P}_t^{edit}),\ y_{\text{source}}) \\ + \lambda_2 \sum |\boldsymbol{w}_{it}|,\end{aligned} \tag{5.13}$$

Algorithm 3: Training Semantic Function $F(h_t, t)$

Input: ϵ_θ(pretrained model), $\{x_0^{(i)}\}_{i=1}^N$(images to precompute), F(Semantic function of an attribute), c^{src}(source text), c^{tar}(target text), S_{inv}(# of inversion steps), K(# of training epochs), f(target model), y_{tar}(target label)

Output: F_t (trained semantic function)

`// Step 1: Precompute latents`

Define $\{\tau_s\}_{s=1}^{S_{inv}}$ s.t $\tau_1 = 0, \tau_{S_{inv}} = T$;

for $i = 1, 2, \ldots, N$ **do**

  for $s = 1, 2, \ldots, S_{inv} - 1$ **do**

    $\epsilon \leftarrow \epsilon_\theta(x_{\tau_s}^{(i)}, \tau_s)$;

    $x_{\tau_{s+1}}^{(i)} = \sqrt{\alpha_{\tau_s}} x_{\tau_s}^{(i)} + \sqrt{1 - \alpha_{\tau_s}}\epsilon$;

  Save the latent $x_{\tau_{S_{inv}}}^{(i)}$;

`// Step 2: Update` F

for *epoch* $= 1, 2, \ldots, K$ **do**

  for $i = 1, 2, \ldots, N$ **do**

    Clone the latent $\tilde{x}_{\tau_{S_{inv}}}^{(i)} = x_{\tau_{S_{inv}}}^{(i)}$;

    for $s = S_{inv}, S_{inv} - 1, \ldots, 1$ **do**

      Extract feature map h_{τ_s} from $\epsilon_\theta(\tilde{x}_{\tau_s}^{(i)})$;

      $\Delta h_{\tau_s} = F_{\tau_s}(h_{\tau_s})$;

      $P = \frac{\tilde{x}_{\tau_s}^{(i)} - \sqrt{1-\alpha_{\tau_s}}\epsilon_\theta(\tilde{x}_{\tau_s}^{(i)}, \Delta h_{\tau_s})}{\sqrt{\alpha_{\tau_s}}}$; $P_{src} = \frac{x_{\tau_s}^{(i)} - \sqrt{1-\alpha_{\tau_s}}\epsilon_\theta(x_{\tau_s}^{(i)})}{\sqrt{\alpha_{\tau_s}}}$;

      $\tilde{x}_{\tau_{s-1}}^{(i)} = \sqrt{\alpha_{\tau_{s-1}}} P + \sqrt{1 - \alpha_{\tau_{s-1}}}\epsilon_\theta(\tilde{x}_{\tau_s}^{(i)})$;

      $x_{\tau_{s-1}}^{(i)} = \sqrt{\alpha_{\tau_{s-1}}} P_{src} + \sqrt{1 - \alpha_{\tau_{s-1}}}\epsilon_\theta(x_{\tau_s}^{(i)})$;

    $L \leftarrow L_{direction}(P, c^{tar}, P_{src}, c^{src}) + \lambda_{recon}\|P - P_{src}\| + \lambda_{adv} L_{CE}(f(P), y_{tar})$;

    Take a gradient step on L and update F;

where $\mathbf{P}_t^{edit} = \mathbf{P}_t\left(\boldsymbol{\epsilon}_t^\theta\left(\boldsymbol{x}_t \mid \sum \boldsymbol{w}_{it}\mathbf{F}_i(\boldsymbol{h}_{it}, t)\right)\right)$, the first cross-entropy loss encourages $\boldsymbol{f}$ to predict y_{target}, the second cross-entropy loss encourages another auxiliary classifier $\boldsymbol{g}$ to make correct prediction y_{source}, and the third term penalizes $\boldsymbol{w}_{it}$ to prevent excessive semantic shifts. We hypothesize that $\boldsymbol{g}$ is relatively uncorrelated with $\boldsymbol{f}$, that can possibly promote the generated UAEs to reside in class y_{source} while crossing the decision boundary of $\boldsymbol{f}$.

Considering diffusion models generate high-level context in the early stage and refine details in the later stage [27], we only modify the generative process in the early stage to achieve adversarial semantic changes with the editing interval $[T, t_{edit}]$, and inject stochastic noise in the later stage to boost image quality with the interval $[t_{boost}, 0]$. The modified generative process of DDIM $p_\theta^{(t)}\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right)$ becomes:

$$\begin{cases} \sqrt{\alpha_{t-1}}\mathbf{P}_t\left(\boldsymbol{\epsilon}_t^\theta\left(\boldsymbol{x}_t \mid \Delta\boldsymbol{h}_t\right)\right) + \mathbf{D}_t & \text{if } T \geq t \geq t_{\text{edit}} \\ \sqrt{\alpha_{t-1}}\mathbf{P}_t\left(\boldsymbol{\epsilon}_t^\theta\left(\boldsymbol{x}_t\right)\right) + \mathbf{D}_t & \text{if } t_{\text{edit}} > t \geq t_{\text{boost}} \\ \mathcal{N}\left(\sqrt{\alpha_{t-1}}\mathbf{P}_t\left(\boldsymbol{\epsilon}_t^\theta\left(\boldsymbol{x}_t\right)\right) + \mathbf{D}_t, \sigma_t^2\boldsymbol{I}\right) & \text{if } t_{\text{boost}} > t \end{cases} \tag{5.14}$$

where $\eta = 1$ only in the interval $[t_{\text{boost}}, 0]$. We follow Kwon et al. [90] to find t_{edit} with $\text{LPIPS}(\boldsymbol{x}, \mathbf{P}_{t_{\text{edit}}}) = 0.33 - \delta$ which represents the shortest editing interval from T to t when $\mathbf{P}_t$ approaches the original image $\boldsymbol{x}$. δ is adaptive to the cosine distance of CLIP text embeddings for different attributes, namely $\delta = 0.33d(E_T\left(c^{\text{source}}\right), E_T\left(c^{\text{target}}\right))$. Similarly, we find t_{boost} with $\text{LPIPS}(\boldsymbol{x}, \boldsymbol{x_t}) = 1.2$ which achieves quality boosting with minimal content change.

Starting from a sampled noise $\boldsymbol{x}_T$, we progressively denoise it through the above three phases to obtain the final output $\boldsymbol{x}_0$ as the potential adversarial example. The SemDiff unrestricted adversarial attack is detailed in Algorithm 4.

Algorithm 4: The algorithm of SemDiff

Input: ϵ_θ(frozen pretrained model), f(target classifier), g(auxiliary classifier), $\{F_i\}_{i=1}^{M}$(M Semantic functions for M attributes), S_{gen}(# of inference steps), K(# of iterations), y_{tar}(target label), y_{src}(source label), t_{edit}(timestep for semantic editing), t_{boost}(timestep for quality boosting)

Output: x_0 (adversarial example)

Define $\{\tau_s\}_{s=1}^{S_{gen}}$ s.t $\tau_1 = 0, \tau_{S_{edit}} = t_{edit}, \tau_{S_{noise}} = t_{\text{boost}}$ and $\tau_{S_{gen}} = T$;

$x_{\tau_{S_{gen}}} \sim \mathcal{N}(0,1)$; $w_i \sim \mathcal{N}(0,1)$;

for $iter = 1, 2, \ldots, K$ **do**

  for $s = S_{gen}, S_{gen} - 1, \ldots, 2$ **do**

    if $s \geq S_{edit}$ **then**

      Extract feature map h_{τ_s} from $\epsilon_\theta(x_{\tau_s})$;

      $\Delta h_{\tau_s} = \sum_{i=1}^{M} w_{i\tau_s} F_i(h_{\tau_s}, \tau_s)$;

      $\tilde{\epsilon} = \epsilon_\theta(x_{\tau_s} | \Delta h_{\tau_s})$; $\epsilon = \epsilon_\theta(x_{\tau_s})$;

      $\sigma_{\tau_s} = 0$;

      $\tilde{P}_{\tau_s} = \frac{x_{\tau_s} - \sqrt{1-\alpha_{\tau_s}}\tilde{\epsilon}}{\sqrt{\alpha_{\tau_s}}}$;

      $L \leftarrow L_{\text{CE}}(f(\tilde{P}_{\tau_s}),\ y_{\text{tar}}) + \lambda_1 L_{\text{CE}}(g(\tilde{P}_{\tau_s}),\ y_{\text{src}}) + \lambda_2 \sum |\boldsymbol{w}_{i\tau_s}|$;

      Take a gradient step on L and update $w_{i\tau_s}$;

    else if $s \geq S_{noise}$ **then**

      $\tilde{\epsilon} = \epsilon = \epsilon_\theta(x_{\tau_s})$; $\sigma_{\tau_s} = 0$;

    else

      $\tilde{\epsilon} = \epsilon = \epsilon_\theta(x_{\tau_s})$;

      $\sigma_{\tau_s} = \sqrt{\frac{1-\alpha_{\tau_s-1}}{1-\alpha_{\tau_s}}}\sqrt{1 - \frac{\alpha_{\tau_s}}{\alpha_{\tau_s-1}}}$;

    $x_{\tau_s-1} = \sqrt{\alpha_{\tau_s-1}}(\frac{x_{\tau_s} - \sqrt{1-\alpha_{\tau_s}}\tilde{\epsilon}}{\sqrt{\alpha_{\tau_s}}}) + \sqrt{1 - \alpha_{\tau_s-1} - \sigma_{\tau_s}^2}\epsilon + \sigma_{\tau_s} z,\ z \sim \mathcal{N}(0,1)$;

  if $f(x_0) = y_{tar}$ **then**

    return x_0

Table 5.1: Accuracy of the classifiers

Classifier	CelebA-HQ	AFHQ	ImageNet
ResNet50	98.4%	99.2%	76.1%
ViT	97.4%	99.7%	81.1%
VGG16	98.6%	99.8%	71.6%

5.4 Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of the proposed SemDiff. We first introduce the experiment settings. Then we evaluate SemDiff in unrestricted adversarial attack on four tasks and three high-resolution datasets. Following, we further investigate the efficacy and imperceptibility of the generated UAEs by testing SemDiff against different defense methods. Finally, we provide the ablation study of SemDiff and report its hyperparameter sensitivity.

5.4.1 Experiment Settings

Datasets and Target Models

We evaluate unrestricted adversarial attacks on four tasks on three high-resolution datasets, including gender classification on CelebA-HQ [80], animal classification on AFHQ [28], church classification and any-class classification on ImageNet [36]. For the targeted classifiers to attack, we use the pretrained ResNet50 [61] and ViT [44] for ImageNet dataset, and retrain them for the tasks on CelebA-HQ and AFHQ. We also adopt VGG16 [162] as the auxiliary classifier for our SemDiff attack and the baseline UGAN [166] attack. The performance of the classifiers are reported in Tab. 5.1.

Baselines

We compare our proposed SemDiff with the state-of-the-art unrestricted adversarial attacks, including GAN-based UGAN [166], diffusion-based AdvDiff [33] and AdvDiffuser [24]. For a fair comparison, we adopt the same pretrained diffusion models for diffusion-based attacks, namely SDEdit [122] on CelebA-HQ and AFHQ, ADM [37] on ImageNet. For GAN-based UGAN, we use StyleGAN2 [81] on CelebA-HQ and AFHQ, BigGAN [17] on ImageNet.

Evaluation Metrics

We use the Attack Success Rate (ASR) to evaluate the attack performance. In terms of the imperceptibility and naturalness of the generated UAEs, we utilize the Fréchet Inception Distance (FID) [68], Kernel Inception Distance (KID) [11] and BRISQUE [123]. Among them, FID and KID measure the similarity between generated UAEs and real-world images, while BRISQUE is a no-reference image quality assessment metric that can evaluate image naturalness similar to human perception.

Implementation Details

We perform targeted attacks in all experiments. For each attack, we generate 1000 UAEs with the same sampled noise images for diffusion models. Since GANs use different input noise vector, we reconstruct the sampled noise images of diffusion models, and then project them to the input latent space of GANs as the input vectors for a fair comparison. For our method, we train $\boldsymbol{F}$ for 1 epoch using 1,000 samples with $S = 40$ on CelebA-HQ and AFHQ, $S = 80$ on ImageNet. We set $\lambda_{adv} = 1$, and use $\lambda_{reg} = \text{CLIP simlarity} * 3$ when reducing L1 loss for different attributes. For multi-attributes optimization, we use $\lambda_1 = 1$ and $\lambda_2 = 10$ on CelebA-HQ and AFHQ, $\lambda_1 = 2$, $\lambda_2 = 15$ on ImageNet. We set $K = 40$ on each attack for a fair

comparison. For the hyperparameter λ_{adv} to control the adversarial guidance when training semantic functions, we empirically set it as 1 for all attributes and did not extensively explore it, since the attack objective is mainly realized by the subsequent multi-attributes optimization, and meanwhile we observe that the semantic function usually converges in less than one epoch, and excessive training tends to generate overly exaggerated attribute changes. We use the directional CLIP simlarity($\mathcal{S}_{\text{dir}}$) following [90, 54] to measure the similarity between the source text and the target text:

$$\mathcal{S}_{\text{dir}}\left(\boldsymbol{x}^{\text{edit}}, c^{\text{target}}; \boldsymbol{x}^{\text{source}}, c^{\text{source}}\right) = \frac{\Delta I \cdot \Delta T}{\|\Delta I\|\|\Delta T\|}. \tag{5.15}$$

We devise a set of attributes with source/target text pairs for each task, more details about the hyperparameters t_{edit}, t_{boost} and CLIP simlarity can be found in Tab. 5.2. Some attributes are neutral such as *smiling* and *young*, while some attributes correspond to the target classes, such as *strong jawline* and *busy eyebrows* for male class in gender classification task. The code is available at: https://github.com/Daizy97/SemDiff.

5.4.2 Unrestricted Adversarial Attack

Gender Classification on CelebA-HQ

In this task, the attack aims to generate UAEs that look like females but are misclassified as males. Our proposed SemDiff uses {*smiling*, *young*, *tanned*, *strong jawline*, *busy eyebrows*} as the attribute set. The detailed results are presented in Tab. 5.3. It is observed that all three diffusion-based attack can achieve 100% ASR, except for UGAN-StyleGAN2, which demonstrates the effectiveness of diffusion models in generating UAEs. Moreover, SemDiff shows the superior performance in terms of BRISQUE, FID and KID, indicating the generated UAEs of SemDiff are more natural and realistic.

Table 5.2: Attributes with corresponding hyperparamenters of different tasks

Task	attribute	source text	target text	CLIP similarity	λ_{reg}	t_{edit}	t_{boost}
Gender	smiling	face	smiling face	0.899	0.899*3	513	167
	young	person	young person	0.905	0.905*3	515	167
	tanned	face	tanned face	0.886	0.886*3	512	167
	short hair	person	person with short hair	0.837	0.837*3	450	167
	strong jawline	person	person with strong jawline	0.821	0.821*3	496	167
	busy eyebrows	person	person with busy eyebrows	0.833	0.833*3	497	167
	big round eyes	person	person with big round eyes	0.787	0.787*3	483	167
	soft facial lines	person	person with soft facial lines	0.739	0.739*3	465	167
Animal	dog happy	dog	happy dog	0.883	0.883*3	430	167
	dog cat fur	dog	dog with fur of cat	0.841	0.841*3	411	167
	dog cat nose	dog	dog with short snout of cat	0.826	0.826*3	403	167
	dog cat ears	dog	dog with pointed ears of cat	0.829	0.829*3	403	167
Church	church night	church	church at night	0.785	0.785*3	341	252
	church old	church	old church	0.796	0.796*3	346	252
	church wooden	church	wooden church	0.793	0.793*3	345	252
Any-class	sunny	photo	photo of a sunny day	0.851	0.851*3	359	252
	rainy	photo	photo of a rainy day	0.850	0.850*3	359	252
	foggy	photo	photo of a foggy day	0.841	0.841*3	357	252
	motion blur	photo	photo with motion blur	0.812	0.812*3	348	252
	defocus blur	photo	photo with defocus blur	0.808	0.808*3	346	252
	frosted glass blur	photo	photo with frosted glass blur	0.762	0.762*3	328	252

We also provide some visual examples in Fig. 5.5, where the first column shows the sampled images reconstructed with the randomly sampled noise images by diffusion models. Notice that the sampled images are just the output images if we do not modify the generative models for attack. We can find that the UAEs generated by AdvDiff and AdvDiffuser contain perceptible perturbations and basically do not change any semantic contents compared with the sampled images. This illustrates that current diffusion-based attacks that perturb in the intermediate latent noise space perform like traditional perturbation-based attacks with superficial perturbations, resulting in inadequate imperceptibility and naturalness. Instead, our SemDiff generate natural UAEs with meaningful semantic changes consistent with the attribute set, such as becoming younger for the females in the images. Such slight and meaningful semantic changes accord with our cognition thus more imperceptible to human eyes. Although UGAN-StyleGAN2 also reveals some semantic changes in their UAEs, their semantic directions are unknown and cannot guarantee naturalness because UGAN attack just optimizes the input latent vector in GANs. In comparison, SemDiff optimizes the weights of different semantic attributes, making the generated UAEs more controllable and realistic.

Animal Classification on AFHQ

This task have three classes, including dog, cat and wildlife. The attack aims to generate UAEs that look like dogs but are misclassified as cats. We use {*dog happy*, *dog cat fur*, *dog cat nose*, *dog cat ears*} as the attribute set for SemDiff. From Tab. 5.3, we can observe that our SemDiff outperforms all the baselines in terms of BRISQUE, FID and KID, confirming the imperceptibility and naturalness of SemDiff. AdvDiff obtains slightly higher ASR than other attacks for ResNet50 as the target classifier. However, when faced with stronger ViT, the ASR of all attacks decreases, while SemDiff achieves the highest ASR, which demonstrates the attack effectiveness of SemDiff.

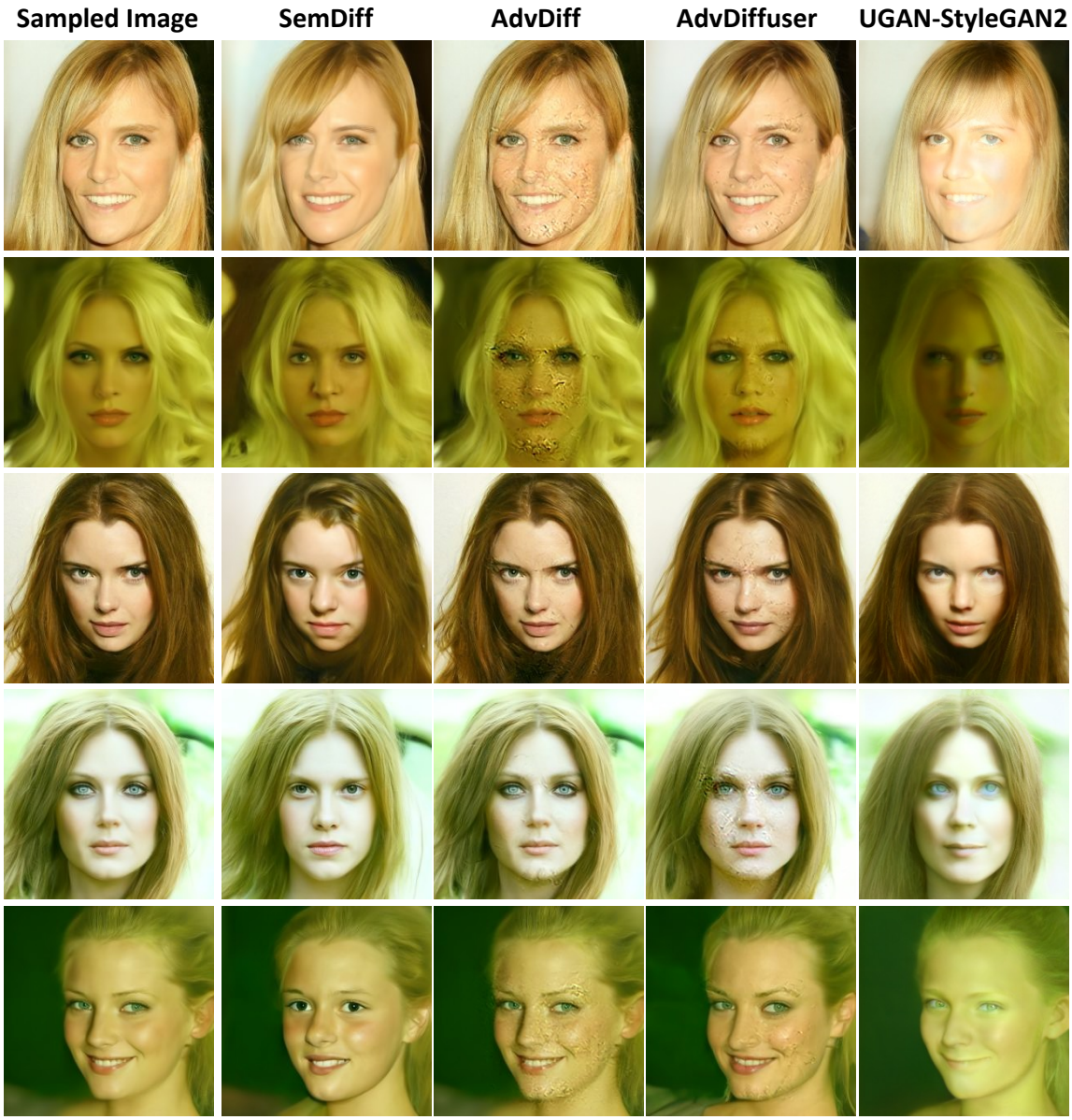


Figure 5.5: Visual examples of the UAEs generated by different attacks for gender classification task on CelebA-HQ. The first column shows the sampled images reconstructed with the randomly sampled noise images by diffusion models.

Table 5.3: Comparison results of the proposed SemDiff attack with baselines in unrestricted adversarial attack

Task	Method	ResNet50				ViT			
		ASR↑	BRISQUE↓	FID↓	KID↓	ASR↑	BRISQUE↓	FID↓	KID↓
Gender classification on CelebA-HQ	AdvDiff	**100%**	27.28	28.81	0.009	**100%**	33.64	22.70	0.008
	AdvDiffuser	**100%**	27.27	28.33	0.009	**100%**	33.64	22.21	0.008
	UGAN-StyleGAN2	93.2%	31.79	38.75	0.014	95.0%	**30.79**	39.61	0.015
	SemDiff (ours)	**100%**	**26.59**	**27.60**	**0.007**	**100%**	31.37	**19.77**	**0.007**
Animal classification on AFHQ	AdvDiff	**96.9%**	24.93	30.46	0.015	93.7%	27.68	39.59	0.018
	AdvDiffuser	96.3%	25.22	30.37	0.015	93.9%	27.88	39.38	0.018
	UGAN-StyleGAN2	91.7%	26.03	38.40	0.024	88.9%	28.00	47.08	0.036
	SemDiff (ours)	95.4%	**23.98**	**26.40**	**0.010**	**94.6%**	**24.59**	**36.36**	**0.012**
Church classification on ImageNet	AdvDiff	**98.9%**	28.09	31.70	0.015	97.6%	31.21	**36.31**	**0.016**
	AdvDiffuser	98.2%	28.23	31.76	0.015	96.9%	31.03	36.38	**0.016**
	UGAN-BigGAN	94.3%	29.01	38.40	0.021	92.7%	31.23	45.08	0.028
	SemDiff (ours)	98.4%	**26.14**	**28.55**	**0.014**	**98.2%**	**26.02**	38.31	0.017
Any-class classification on ImageNet	AdvDiff	**99.1%**	27.47	33.57	0.020	**98.3%**	30.21	35.31	0.019
	AdvDiffuser	97.9%	27.03	34.86	0.020	97.2%	30.31	34.23	0.019
	UGAN-BigGAN	94.7%	33.02	46.97	0.037	83.4%	33.55	41.30	0.032
	SemDiff (ours)	98.5%	**25.04**	**29.62**	**0.017**	97.4%	**27.74**	**28.05**	**0.017**

For the visual examples in Fig. 5.6, we notice the similar phenomenon in CelebA-HQ that the UAEs generated by AdvDiff and AdvDiffuser show limited naturalness with visible perturbations, while SemDiff and UGAN-StyleGAN2 modify the semantics of UAEs to cause misclassification. Furthermore, SemDiff exhibits more natural and meaningful semantic changes compared with UGAN-StyleGAN2, leading to UAEs of higher quality. For example, some dogs have denser hair and their ears become shorter and more pointed, resembling those of cats. These semantic changes also accord with *dog cat fur* and *dog cat ears* attributes.

Church classification on ImageNet

ImageNet contains 1000 classes, and this task aims to generate UAEs that look like churches but are misclassified as other classes. We use {*church night*, *church old*, *church wooden*} as the attribute set for SemDiff. According to the results in Tab. 5.3, SemDiff obtains comparable performance with AdvDiff in terms of ASR and bet-

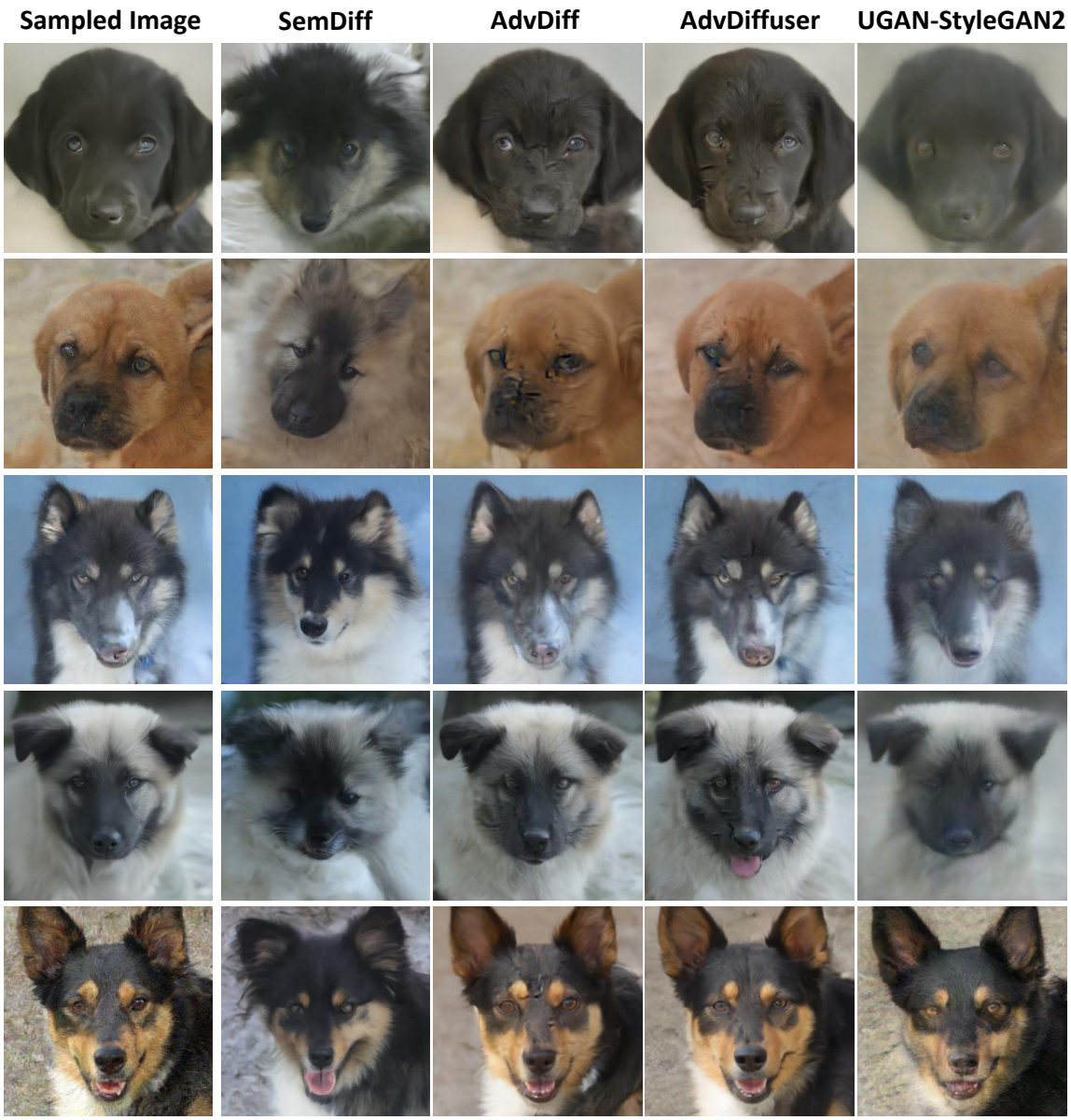


Figure 5.6: Visual examples of the UAEs generated by different attacks for animal classification task on AFHQ. The first column shows the sampled images reconstructed with the randomly sampled noise images by diffusion models.

ter performance in most of the naturalness metrics, which demonstrates the better naturalness of SemDiff once again. We also observe that there is a relatively large gap between UGAN-BigGAN and diffusion-based attacks in terms of ASR, FID and KID, which suggests that the performance of unrestricted adversarial attacks is heavily affected by the generation quality of generative models. Th visual examples are provided in Fig. 5.7.

Any-class Classification on ImageNet

In previous tasks, the attributes set is designed specific to the source class, which may limit the generalization of the proposed SemDiff to other classes. In this task, we try to create some universal attributes that can be applied to any class for modification. Inspired by the common visual corruptions used in ImageNet-C dataset [65], we design a set of attributes related to some common weather or blurs that usually exist in a photo: {*sunny*, *foggy*, *motion blur*, *defocus blur*, *frosted glass blur*}. Therefore, any class object can be modified with these universal attributes to achieve attack. The results are reported in Tab. 5.3. It is found that SemDiff shows comparable performance with other diffusion-based attacks, and outperforms UGAN-BigGAN with a large margin in terms of ASR. In addition, SemDiff still dominates other attacks in all three imperceptibility and naturalness metrics, which confirms the superiority of our attack. Th visual examples are provided in Fig. 5.8.

5.4.3 Adversarial Defenses

To further validate the efficacy of the proposed SemDiff attack, we investigate whether our SemDiff can evade four different defense methods, including JPEG compression [115], feature squeezing [206], SRNet [14] and adversarial training model [14]. JPEG compression is a preprocessing-based defense that aims to eliminate adversarial perturbations by compressing input images [115]. Feature squeezing (FS) [206]

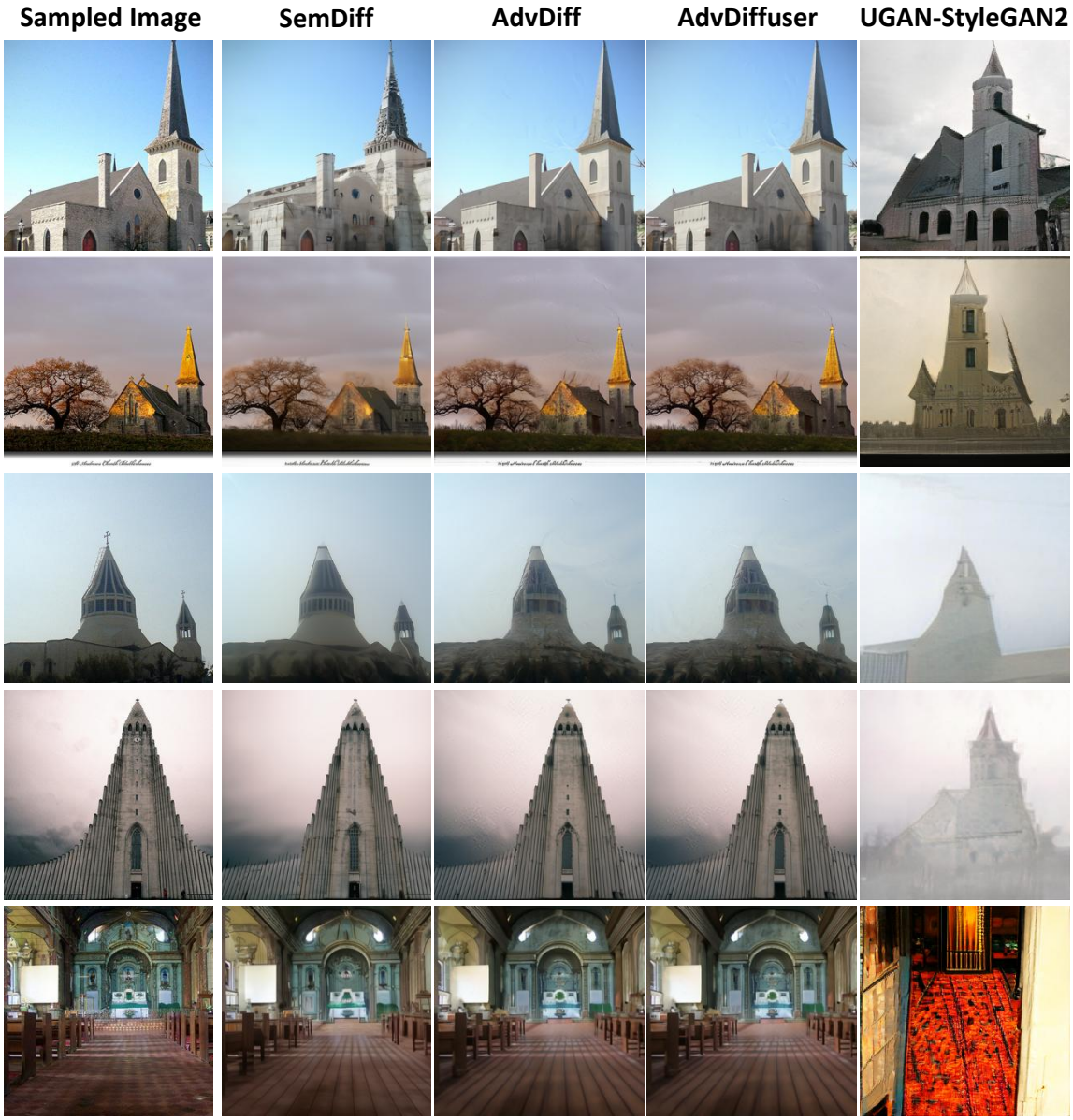


Figure 5.7: Visual examples of the UAEs generated by different attacks for church classification task on ImageNet. The first column shows the sampled images reconstructed with the randomly sampled noise images by diffusion models.

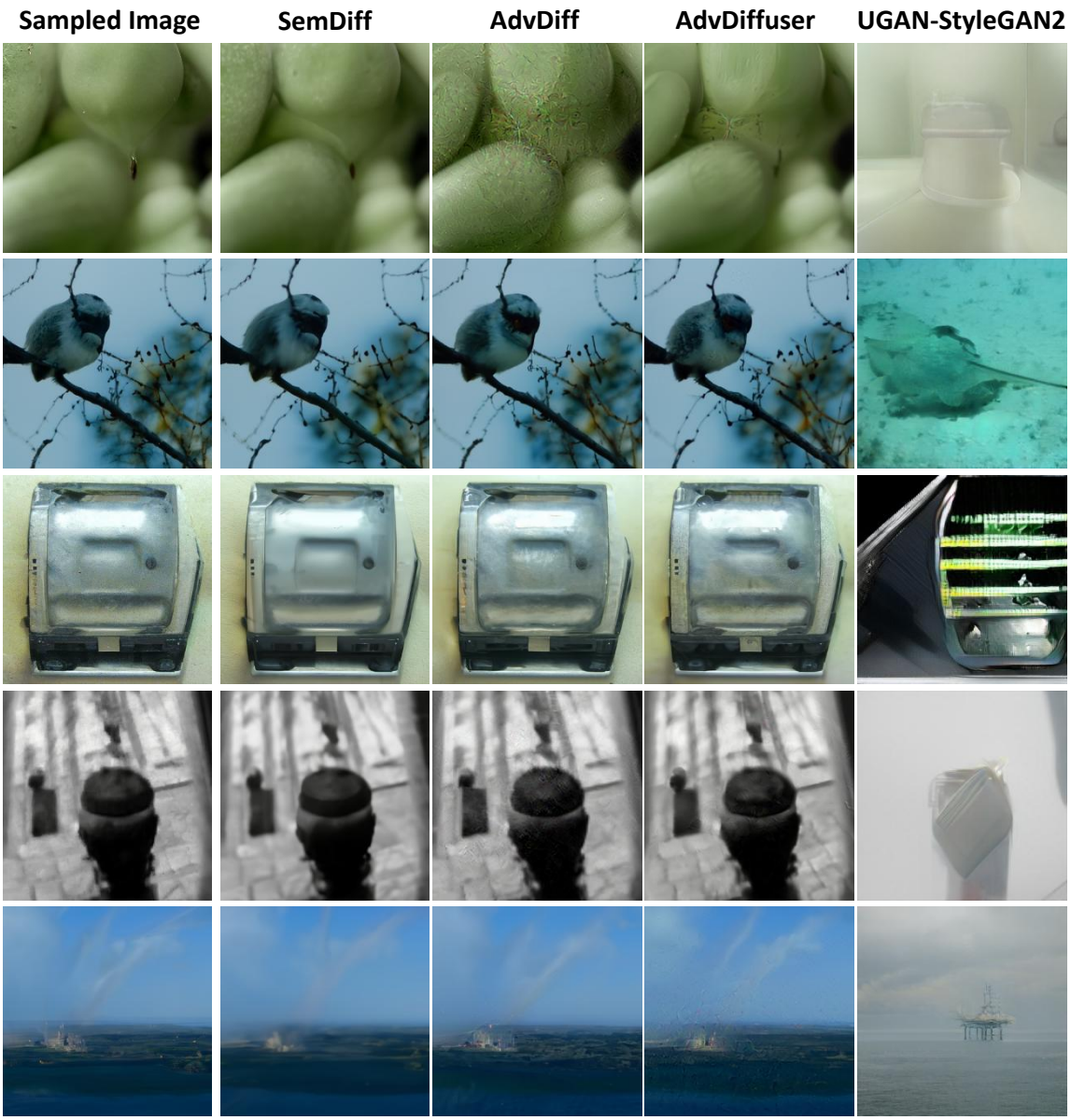


Figure 5.8: Visual examples of the UAEs generated by different attacks for church classification task on ImageNet. The first column shows the sampled images reconstructed with the randomly sampled noise images by diffusion models.

Table 5.4: Comparison results of the proposed SemDiff attack with baselines against four defenses

Task	Method	None	JPEG	FS	SRNet	AT
ResNet50	AdvDiff	99.1%	50.1%	60.5%	55.1%	58.7%
	AdvDiffuser	97.9%	52.4%	58.3%	54.8%	59.4%
	UGAN	94.7%	65.8%	70.6%	52.3%	55.1%
	SemDiff	98.8%	**71.3%**	**81.4%**	**56.2%**	**64.9%**
ViT	AdvDiff	98.3%	56.2%	55.1%	53.7%	56.4%
	AdvDiffuser	97.2%	58.7%	52.4%	**58.4%**	60.2%
	UGAN	83.4%	66.1%	68.4%	49.6%	58.3%
	SemDiff	97.4%	**78.3%**	**73.9%**	54.7%	**63.8%**

and SRNet [14] are two detection-based defenses that try to detect adversarial examples before feeding them into the classifier, where FR detects adversarial examples based on the uncertainty of model output before and after squeezing, SRNet is a CNN-based binary classifier to distinguish adversarial examples from clean ones. Adversarial training (AT) [77] is a robustness-based defense that trains the classifier with adversarial examples to improve the robustness of the model against adversarial attacks. We set the quality factor as 75 for JPEG compression, $(5-bit, 2\times 2, 11-3-4)$ for FS, finetune SRNet with the UAEs generated by all attacks, and use an adversarially training model IncRes-v2-adv-ens [14].

The evaluation results are presented in Tab. 5.4, where the None column denotes the ASR without any defense while the other columns represent the ASR under different defense methods. It is found that the proposed SemDiff is more robust than the baseline attacks against most of the defenses. Specifically, the SemDiff attack achieves the highest ASR under JPEG compression, feature squeezing, and adversarial training with a large margin. Among them, JPEG compression and feature squeezing both

preprocess input images with compression, smoothing or color bit squeezing, which can remove high-frequency noise and perturbations in images. The adversarial examples used for adversarial training are generated by perturbation-based attacks, such as FGSM [173]. As discussed before, the generated UAEs of AdvDiff and AdvDiffuser perform like traditional perturbation-based attacks with superficial perturbations, thus are more sensitive to these defenses in comparison with SemDiff and UGAN. For the binary classifier detector SRNet which is trained with the UAEs generated by all test attacks, the ASR degrades most significantly, but SemDiff can still obtain a competitive performance. In summary, the proposed SemDiff attack is more robust against various defenses, which further confirms its attack effectiveness and imperceptibility.

5.4.4 Ablation Study

In this subsection, we carry out the ablation study on the proposed SemDiff method, including the adversarial semantic attribute and the multi-attributes optimization approach. To verify the effectiveness of the adversarial semantic attribute, we additionally train the semantic function $\boldsymbol{F}$ without adversarial guidance to obtain clean semantic attributes. As for the multi-attributes optimization approach, we implement a single-attribute optimization using line search to gradually increase the attribute weight until misclassification.

We adopt the SemDiff on CelebA-HQ with the attribute set {*smiling*, *young*, *tanned*, *strong jawline*, *busy eyebrows*} for the ablation study. We use "adv_attr" and "attr" to distinguish whether the method employs adversarial semantic attributes or clean semantic attributes, and use "multi" and "single" to distinguish if the method employs multi-attributes optimization or single-attribute optimization. The results are provided in Tab. 5.5. We report an extra metric weight to measure the mean absolute weight of attributes when the attack succeeds. The weight is expected to be close

to 0 so that the generated UAEs resemble the sampled images with slight semantic changes but can successfully mislead the target classifier.

Effectiveness of Adversarial Semantic Attribute

From Tab. 5.5, we can find that all "adv_attr" methods gain higher ASR than the corresponding "attr" methods, which demonstrates the adversarial semantic attributes indeed enhance the attack capability compared with clean semantic attributes. Moreover, the weight of attributes of all "adv_attr" methods is also much smaller than that of the corresponding "attr" methods, indicating that the SemDiff equipped adversarial semantic attributes can achieve attack success with slighter semantic changes and higher efficiency. On the other hand, two types of methods have similar BRISQUE performance, but "adv_attr" methods achieve better FID and KID. This suggests that although clean semantic attributes can preserve the naturalness of UAEs to some extent, they generally underperform compared to adversarial semantic attributes, as the UAEs that are difficult to attack successfully will exhibit more significant and even distorted semantic changes. Therefore, the results validate the efficacy of adversarial semantic attributes in the SemDiff attack.

Effectiveness of Multi-Attributes Optimization Approach

We can observe that in Tab. 5.5, "adv_attr & multi" method performs better than any "single" method, especially in terms of the attribute weight. Due to the limited optimization space of the single-attribute optimization method, it is hard to achieve the same level of attack efficiency as the multi-attributes optimization method. Excessive optimization in a single attribute may lead to a large weight of the attribute, and cause the generated UAEs to deviate from the sampled images with unnatural semantic changes. Additionally, an intriguing finding is that some neutral attributes such as *smiling* tend to perform better than gender-related attributes like *strong jaw-*

Table 5.5: Ablation study of SemDiff attack against ResNet50. Weight denotes the mean absolute weight of attributes when the attack succeeds.

Method	ASR↑	BRISQUE↓	FID↓	KID↓	weight↓
adv_attr & multi	**100%**	26.59	**27.60**	**0.007**	**0.174**
attr & multi	76.6%	**24.03**	36.81	0.014	0.414
adv_strong_jawline & single	99.2%	26.67	29.59	0.009	0.503
strong_jawline & single	45.6%	28.51	33.48	0.012	1.270
adv_busy_eyebrows & single	99.9%	31.71	31.80	0.010	0.729
busy_eyebrows & single	69.3%	29.69	47.76	0.024	1.442
adv_smiling & single	**100%**	25.24	28.86	0.008	0.433
smiling & single	90.6%	27.58	40.92	0.022	1.293

line and *busy eyebrows* when attacking the gender classification task. We speculate that the male facial images in the dataset may appear more serious and lack smiles compared to female facial images, causing a negative weight of the *smiling* attribute tends to lead the UAEs to be misclassified as a male.

5.4.5 Hyperparameter Analysis

We analyze and discuss the impact of the important hyperparameters of SemDiff in the subsection. Note that our proposed method does not require retraining the diffusion models, and the attack is performed in the reverse process.

Influence of λ_1 and λ_2

λ_1 and λ_2 are two hyperparameters used in the multi-attributes optimization approach. λ_1 is the weight of reconstruction loss to keep UAEs resembling the source class y_{source}, and λ_2 is the weight of the penalty term to control the weight size

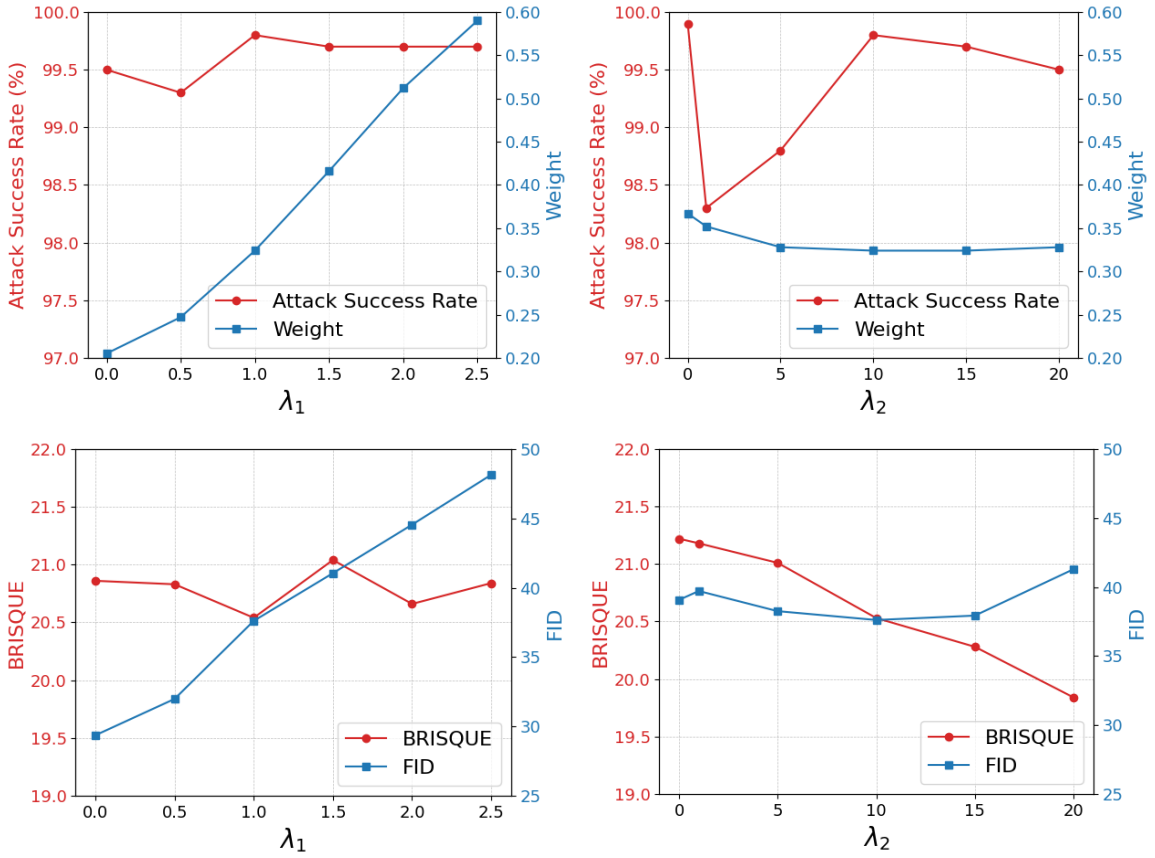


Figure 5.9: Hyperparameter analysis of λ_1 and λ_2 in SemDiff. The results are generated on CelebA-HQ dataset against ResNet50 model. We adopt the ASR, Weight, BRISQUE, FID and KID to measure the impact on attack effectiveness and generation quality.

and prevent excessive semantic shifts. Thus, both λ_1 and λ_2 affect the balance between the attack effectiveness and the imperceptibility of the generated UAEs. From Fig. 5.9, we can observe that as λ_1 increases, the ASR and BRISQUE show small variation, while the Weight and FID also increase. This indicates a certain level of conflict between the objectives of causing the misclassification of target classifiers and maintaining the imperceptibility of UAEs to be consistent with y_{source}. Although achieving both goals requires larger attribute weights, λ_1 is indispensable to ensure the generated UAEs resemble the source class. As for λ_2, as it increases, both the Weight and BRISQUE exhibit a gradual decline, which demonstrates that the weight penalty term effectively reduces semantic changes in the UAE, resulting in more natural UAEs close to the sampled images. However, too large λ_2 will also influence the ASR and FID performance. Thus, we set a moderate λ_1 and λ_2 to better balance the attack effectiveness and the imperceptibility of the generated UAEs.

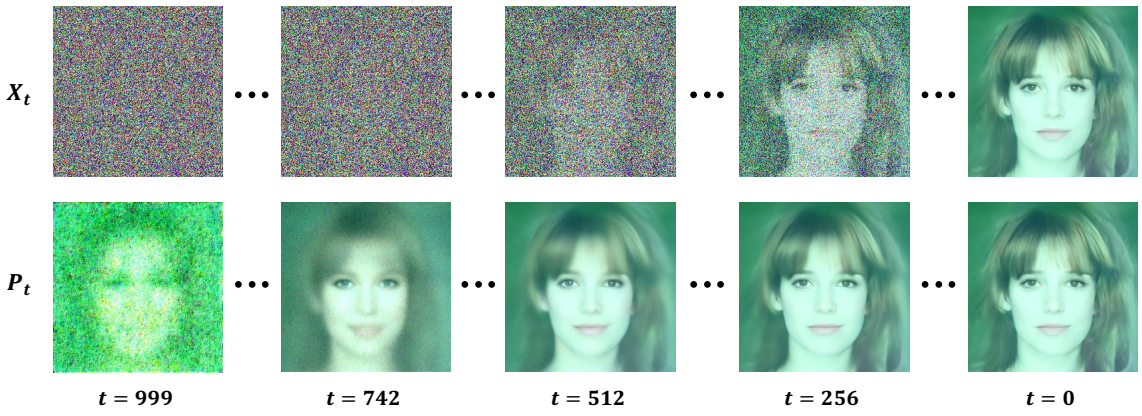


Figure 5.10: Visual examples of $\mathbf{P}_t$ or $\boldsymbol{x}_t$ at different timesteps in the denoising process of DDIM. The examples are generated by SemDiff on CelebA-HQ dataset.

Choosing $\mathbf{P}_t$ or $\boldsymbol{x}_t$

One main difference between our proposed SemDiff and previous diffusion-based attacks [33, 24] is that we modify $\mathbf{P}_t$ and obtain the adversarial gradient through $\mathbf{P}_t$ instead of $\boldsymbol{x}_t$ in DDIM. $\mathbf{P}_t$ is the predicted $\boldsymbol{x}_0$, and constitutes $\boldsymbol{x}_t$ together with $\mathbf{D}_t$

Table 5.6: Ablation study of SemDiff attack against ResNet50. Weight denotes the mean absolute weight of attributes when the attack succeeds.

Method	ASR↑	BRISQUE↓	FID↓	KID↓	Weight↓
SemDiff with $\mathbf{P}_t$	**100%**	**26.59**	**27.60**	**0.007**	**0.174**
SemDiff with X_t	65.7%	29.82	31.52	0.012	0.352

and random noise according to Eq. (5.6). Compared to $\boldsymbol{x}_t$, $\mathbf{P}_t$ better approximates the final output $\boldsymbol{x}_0$, and already contains clear semantic contents in the early stage of the denoising process, as shown in Fig. 5.10. To investigate the effect of choosing $\mathbf{P}_t$ or $\boldsymbol{x}_t$, we test our SemDiff equipped with $\mathbf{P}_t$ and $\boldsymbol{x}_t$, respectively. From the results reported in Tab. 5.6, we can observe that SemDiff with $\mathbf{P}_t$ outperforms SemDiff with $\boldsymbol{x}_t$ in all metrics, particularly achieving a significant lead in ASR. This suggests that SemDiff with $\boldsymbol{x}_t$ can not obtain accurate adversarial gradients through noisy $\boldsymbol{x}_t$ in the early stage of the denoising process, leading to unsuccessful attacks and also affecting the naturalness and imperceptibility of the generated UAEs. In comparison, the clear semantic contents in $\mathbf{P}_t$ in the early stage are enough to guide SemDiff to generate effective UAEs with adversarial semantic attribute shifts.

5.5 Conclusion

In this paper, we present a novel unrestricted adversarial attack, SemDiff, to generate natural and imperceptible unrestricted adversarial examples (UAEs) with diffusion models. Different from existing diffusion-based methods that simply optimize in the intermediate latent space, we first propose to utilize the semantic latent space of diffusion models to obtain adversarial but meaningful attributes for attack. Specifically, we create a set of adversarial semantic attributes and train semantic functions to learn such attributes in order to guide the generation of diffusion models. We further devise a multi-attributes optimization approach to optimize the weights of

these attributes, so that the generated UAEs can achieve attack success while maintaining naturalness with semantic attributes changes. Extensive experiments on four tasks on three high-resolution datasets, including CelebA-HQ, AFHQ, and ImageNet, demonstrate that SemDiff outperforms the state-of-the-art methods in both attack effectiveness and the imperceptibility of generated UAEs. Moreover, we test SemDiff against various adversarial defenses, and the results show that SemDiff is capable of evading different defenses, which further validates its threatening. Considering the effectiveness and realism of the generated UAEs of SemDiff, we encourage researchers to further explore the vulnerability of deep neural networks and develop more robust methods to resist unrestricted adversarial attacks.

Chapter 6

Conclusion and Perspectives

In this last chapter, we first summarize our contributions made within this thesis and then outline potential research questions that remain unanswered but interesting to explore in the future.

6.1 Conclusion

This thesis systematically investigates and enhances imperceptible adversarial attacks in images. Considering the ambiguity of imperceptibility in different types of adversarial attacks, we first propose a taxonomy of adversarial examples, categorizing them into *input-specific*, *input-agnostic*, and *input-free* adversarial examples based on their mathematical definitions and generation mechanisms. This taxonomy provides a unified framework for understanding the distinct imperceptibility requirements and attack objectives across different types of adversarial examples. Next, we provide a comprehensive review about the imperceptibility of adversarial examples from dual perspectives: *human imperceptibility* (naturalness and realism to human observers) and *machine imperceptibility* (capability of evading machine detectors).

Then, we focus on three research questions about the imperceptibility for different

types of adversarial examples, and propose novel attack methods accordingly to address them:

- **Saliency Attack for Input-Specific Adversarial Examples:** Considering the query-sensitive nature and poor imperceptibility problem in black-box input-specific attacks, we introduce Saliency Attack to restrict the perturbations to a small but classifier-sensitive region, and further refine the perturbations to generate efficient and imperceptible input-specific adversarial examples in the black-box setting.

- **JUAP for Input-Agnostic Adversarial Examples:** Targeting the interpretation discrepancy phenomenon of adversarial examples, we aim to explore whether a universal adversarial perturbation can achieve both human and machine imperceptibility. To this end, we propose a novel framework called JUAP (Joint Universal Adversarial Perturbation) that jointly optimizes the adversarial loss and the interpretation discrepancy.

- **Diffusion-based SemDiff for Input-Free Adversarial Examples:** Facing the unnatural adversarial examples generated by input-free adversarial attacks, we propose SemDiff, leveraging the semantic latent space of diffusion models to generate adversarial examples. By optimizing the weights of multiple semantic attributes, the generated adversarial examples become natural with semantically meaningful attribute shifts.

Extensive experiments validate the superiority of our methods in generating imperceptible yet potent adversarial examples across multiple datasets (e.g., ImageNet, AFHQ and CelebA-HQ). These advancements not only expose critical vulnerabilities in DNNs but also provide actionable insights for developing robust DNN-based systems.

6.2 Future Perspectives

Building on the contributions of this thesis, several promising directions emerge to advance imperceptible adversarial attacks and their broader implications.

First, the inherent conflict between maximizing attack success and minimizing imperceptibility remains a core challenge. While our methods mitigate this attack-imperceptibility trade-off, fundamental limitations persist. It is possible formalize the problem as a multi-objective optimization framework, dynamically balancing adversarial strength and imperceptibility across different attack scenarios. Techniques such as Pareto optimization, reinforcement learning, or evolutionary algorithms could help identify optimal solutions along the attack-imperceptibility frontier.

In addition, the thesis focuses the imperceptibility of adversarial attacks primarily on images. It is valuable to extend the concepts of imperceptibility and methods to other domains and modalities, such as video, audio, text, and multimodal systems. This extension will require rethinking the definition of imperceptibility and developing new metrics and methodologies tailored to the unique characteristics of these domains. The target model should also not be limited to image classifiers, but also include other types of DNN-based models, especially large language models (LLMs) and multimodal systems.

Last but not least, our third work, SemDiff, explores the semantic latent space of diffusion models for generating natural adversarial examples with meaningful attribute shifts. It is suggested that current image processing systems are still vulnerable to imperceptible adversarial attacks, new adversarial defenses and more robust DNN-based models should be developed to mitigate these vulnerabilities. On the other hand, the generation scheme of SemDiff can be applied to controllable image generation and editing, which can be further explored in future research.

In conclusion, this thesis advances the field of adversarial machine learning by sys-

tematically addressing the imperceptibility challenge across diverse attack paradigms. The proposed methodologies, insights, and open questions lay a foundation for future research toward secure, reliable, and human-aligned DNN systems.

References

[1] Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 700–712, 2020.

[2] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 2006.

[3] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430, 2018.

[4] Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, Huan Zhang, Cho-Jui Hsieh, and Mani B. Srivastava. Genattack: practical black-box attacks with gradient-free optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1111–1119, Prague, Czech Republic, July 2019.

[5] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *Proceedings of the 16th European Conference on Computer Vision, Part XXIII*, volume 12368, pages 484–501, Glasgow, UK, August 2020.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, May 2015.

[7] Vincent Ballet, Xavier Renard, Jonathan Aigrain, Thibault Laugel, Pascal Frossard, and Marcin Detyniecki. Imperceptible adversarial attacks on tabular data. *arXiv preprint arXiv:1911.03274*, 2019.

[8] Hubert Baniecki and Przemyslaw Biecek. Adversarial attacks and defenses in explainable artificial intelligence: A survey. *Information Fusion*, page 102303, 2024.

[9] Osbert Bastani, Yani Ioannou, Leonidas Lampropoulos, Dimitrios Vytiniotis, Aditya Nori, and Antonio Criminisi. Measuring neural net robustness with constraints. *Advances in Neural Information Processing Systems*, 29, 2016.

[10] Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and DA Forsyth. Unrestricted adversarial examples via semantic manipulation. In *8th International Conference on Learning Representations*, 2020.

[11] Mikolaj Binkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD gans. In *International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada, April 2018.

[12] Akhilan Boopathy, Sijia Liu, Gaoyuan Zhang, Cynthia Liu, Pin-Yu Chen, Shiyu Chang, and Luca Daniel. Proper network interpretability helps adversarial robustness in classification. In *International Conference on Machine Learning*, pages 1014–1023. PMLR, 2020.

[13] Mehdi Boroumand, Mo Chen, and Jessica J. Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, 2019.

[14] Mehdi Boroumand, Mo Chen, and Jessica J. Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, 2019.

[15] Luca Bortolussi, Ginevra Carbone, Luca Laurenti, Andrea Patane, Guido Sanguinetti, and Matthew Wicker. On the robustness of bayesian neural networks to adversarial attacks. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[16] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, BC, Canada, April 2018.

[17] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*, New Orleans, LA, May 2019.

[18] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

[19] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 3–14, 2017.

[20] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy*, pages 39–57, San Jose, CA, May 2017.

[21] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847. IEEE, 2018.

[22] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *Proceedings of the 2020 IEEE*

Symposium on Security and Privacy, pages 1277–1294, an Francisco, CA, May 2020.

[23] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26, Dallas, TX, November 2017.

[24] Xinquan Chen, Xitong Gao, Juanjuan Zhao, Kejiang Ye, and Cheng-Zhong Xu. Advdiffuser: Natural adversarial example synthesis with diffusion models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4539–4549, Paris, France, October 2023.

[25] Minhao Cheng, Simranjit Singh, Patrick Chen, Pin Yu Chen, Sijia Liu, and Cho Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. In *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[26] Minhao Cheng, Huan Zhang, Cho Jui Hsieh, Thong Le, Pin Yu Chen, and Jinfeng Yi. Query-efficient hard-label black-box attack: An optimization-based approach. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[27] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2416–2425, New Orleans, LA, June 2022.

[28] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8185–8194, Seattle, WA, June 2020.

[29] Edward Chou, Florian Tramer, and Giancarlo Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 48–54. IEEE, 2020.

[30] Francesco Croce and Matthias Hein. Sparse and imperceivable adversarial attacks. In *Proceedings of the 2019 IEEE International Conference on Computer Vision*, pages 4723–4731, Seoul, Korea (South), October 2019.

[31] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. *Advances in Neural Information Processing Systems*, 30, 2017.

[32] Jiazhu Dai and Le Shu. Fast-uap: An algorithm for speeding up universal adversarial perturbation generation with orientation of perturbation vectors. *arXiv preprint arXiv:1911.01172*, 2019.

[33] Xuelong Dai, Kaisheng Liang, and Bin Xiao. Advdiff: Generating unrestricted adversarial examples using diffusion models. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *European Conference on Computer Vision (ECCV)*, volume 15104, pages 93–109, Milan, Italy, September 2024.

[34] Zeyu Dai, Shengcai Liu, Qing Li, and Ke Tang. Saliency attack: towards imperceptible black-box adversarial attack. *ACM Transactions on Intelligent Systems and Technology*, 14(3):1–20, 2023.

[35] Stanislas Dehaene. The neural basis of the weber–fechner law: a logarithmic mental number line. *Trends in cognitive sciences*, 7(4):145–147, 2003.

[36] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 248–255, Miami, FL, June 2009.

[37] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 8780–8794, Virtual, December 2021.

[38] Keyan Ding, Ronggang Wang, and Shiqi Wang. Social media popularity prediction: A multiple feature fusion approach with deep neural networks. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2682–2686, 2019.

[39] Yujuan Ding, Wenqi Fan, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on rag meets llms: Towards retrieval-augmented large language models. *arXiv preprint arXiv:2405.06211*, 2024.

[40] Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. *Advances in Neural Information Processing Systems*, 32, 2019.

[41] Xiaoyi Dong, Jiangfan Han, Dongdong Chen, Jiayang Liu, Huanyu Bian, Zehua Ma, Hongsheng Li, Xiaogang Wang, Weiming Zhang, and Nenghai Yu. Robust superpixel-guided attentional adversarial attack. In *Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition*, pages 12892–12901, Seattle, WA, June 2020.

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

[43] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7714–7722, 2019.

[44] 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. In *International Conference on Learning Representations (ICLR)*, Virtual, May 2021.

[45] Rudresh Dwivedi, Devam Dave, Het Naik, Smiti Singhal, Rana Omer, Pankesh Patel, Bin Qian, Zhenyu Wen, Tejal Shah, Graham Morgan, et al. Explainable ai (xai): Core ideas, techniques, and solutions. *ACM Computing Surveys*, 55(9):1–33, 2023.

[46] Logan Engstrom, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling cnns with simple transformations. *CoRR*, 2017.

[47] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018.

[48] Wenqi Fan, Xiangyu Zhao, Xiao Chen, Jingran Su, Jingtong Gao, Lin Wang, Qidong Liu, Yiqi Wang, Han Xu, Lei Chen, et al. A comprehensive survey on trustworthy recommender systems. *arXiv preprint arXiv:2209.10117*, 2022.

[49] Wenqi Fan, Xiangyu Zhao, Qing Li, Tyler Derr, Yao Ma, Hui Liu, Jianping Wang, and Jiliang Tang. Adversarial attacks for black-box recommender sys-

tems via copying transferable cross-domain user profiles. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[50] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

[51] Sid Ahmed Fezza, Yassine Bakhti, Wassim Hamidouche, and Olivier Déforges. Perceptual evaluation of adversarial attacks for cnn-based image classification. In *Proceedings of the 11th International Conference on Quality of Multimedia Experience*, pages 1–6, Berlin, Germany, June 2019.

[52] Samuel G Finlayson, John D Bowers, Joichi Ito, Jonathan L Zittrain, Andrew L Beam, and Isaac S Kohane. Adversarial attacks on medical machine learning. *Science*, 363(6433):1287–1289, 2019.

[53] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437, 2017.

[54] Rinon Gal, Or Patashnik, Haggai Maron, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):141:1–141:13, 2022.

[55] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3681–3688, 2019.

[56] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, May 2015.

[57] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

[58] Chuan Guo, Jared S. Frank, and Kilian Q. Weinberger. Low frequency adversarial perturbation. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*, volume 115, pages 1127–1137, Tel Aviv, Israel, July 2019.

[59] Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Jian Wang, Bing Yu, Wei Feng, and Yang Liu. Watch out! motion is blurring the vision of your deep neural networks. *Advances in Neural Information Processing Systems*, 33:975–985, 2020.

[60] Jamie Hayes and George Danezis. Learning universal adversarial perturbations with generative models. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 43–49. IEEE, 2018.

[61] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, Las Vegas, NV, June 2016.

[62] Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *The Twelfth International Conference on Learning Representations*, 2023.

[63] Christopher E Heil and David F Walnut. Continuous and discrete wavelet transforms. *SIAM review*, 31(4):628–666, 1989.

[64] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmenta-

tion. In *Proceedings of the IEEE international conference on computer vision*, pages 2755–2764, 2017.

[65] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, May 2019.

[66] Dan Hendrycks and Kevin Gimpel. Early methods for detecting adversarial images. 2017.

[67] Juyeon Heo, Sunghwan Joo, and Taesup Moon. Fooling neural network interpretations via adversarial model manipulation. *Advances in Neural Information Processing Systems*, 32, 2019.

[68] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 6626–6637, Long Beach, CA, December 2017.

[69] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[70] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, Virtual, December 2020.

[71] Hossein Hosseini and Radha Poovendran. Semantic adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1614–1619, 2018.

[72] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[73] Lifeng Huang, Chengying Gao, and Ning Liu. Erosion attack: Harnessing corruption to improve adversarial examples. *IEEE Transactions on Image Processing*, 32:4828–4841, 2023.

[74] Aminul Huq and Mst Tasnim Pervin. Analysis of adversarial attacks on skin cancer recognition. In *International Conference on Data Science and Its Applications (ICoDSA)*, pages 1–4, 2020.

[75] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 2142–2151, Stockholmsmässan, Stockholm, Sweden, July 2018.

[76] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, LA, May 2019.

[77] Xiaojun Jia, Yong Zhang, Baoyuan Wu, Jue Wang, and Xiaochun Cao. Boosting fast adversarial training with learnable adversarial initialization. *IEEE Transactions on Image Processing*, 31:4417–4430, 2022.

[78] Linxi Jiang, Xingjun Ma, Shaoxiang Chen, James Bailey, and Yu-Gang Jiang. Black-box adversarial attacks on video recognition models. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 864–872, 2019.

[79] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[80] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada, April 2018.

[81] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, Virtual, December 2020.

[82] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[83] Andre Kassis and Urs Hengartner. Unmarker: A universal attack on defensive image watermarking. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 2602–2620, 2025.

[84] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131, 2018.

[85] Mehrgan Khoshpasand and Ali A. Ghorbani. On the generation of unrestricted adversarial examples. In *50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops*, pages 9–15, Valencia, Spain, June 2020.

[86] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *IEEE/CVF Conference*

on Computer Vision and Pattern Recognition (CVPR), pages 2416–2425, New Orleans, LA, June 2022.

[87] Hyungsub Kim, Rwitam Bandyopadhyay, Muslum Ozgur Ozmen, Z Berkay Celik, Antonio Bianchi, Yongdae Kim, and Dongyan Xu. A systematic study of physical sensor attack hardness. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 2328–2347. IEEE, 2024.

[88] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[89] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial Intelligence Safety and Security*, pages 99–112. Chapman and Hall/CRC, 2018.

[90] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have A semantic latent space. In *International Conference on Learning Representations (ICLR)*, Kigali, Rwanda, May 2023.

[91] Cassidy Laidlaw and Soheil Feizi. Functional adversarial attacks. *Advances in neural information processing systems*, 32, 2019.

[92] Eric C. Larson and Damon M. Chandler. Most apparent distortion: full-reference image quality assessment and the role of strategy. *Journal of Electronic Imaging*, 19(1):011006, 2010.

[93] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[94] Min Seok Lee, WooSeok Shin, and Sung Won Han. TRACER: extreme attention guided salient object tracing network (student abstract). In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, pages 12993–12994, Virtual Event, February 2022.

[95] Piyawat Lertvittayakumjorn, Lucia Specia, and Francesca Toni. Find: Human-in-the-loop debugging deep text classifiers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 332–348, 2020.

[96] Piyawat Lertvittayakumjorn and Francesca Toni. Explanation-based human debugging of nlp models: A survey. *Transactions of the Association for Computational Linguistics*, 9:1508–1528, 2021.

[97] Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. Trustworthy ai: From principles to practices. *ACM Computing Surveys*, 55(9):1–46, 2023.

[98] Juncheng Li, Frank Schmidt, and Zico Kolter. Adversarial camera stickers: A physical camera-based attack on deep learning systems. In *International conference on machine learning*, pages 3896–3904. PMLR, 2019.

[99] Lin Li. Towards robust visual classification through adversarial training. 2024.

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

[101] Nannan Li and Zhenzhong Chen. Toward visual distortion in black-box attacks. *IEEE Transactions on Image Processing*, 30:6156–6167, 2021.

[102] Xi Li, David J Miller, Zhen Xiang, and George Kesidis. Bic-based mixture model defense against data poisoning attacks on classifiers: A comprehensive study. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[103] Xurong Li, Shouling Ji, Meng Han, Juntao Ji, Zhenyu Ren, Yushan Liu, and Chunming Wu. Adversarial examples versus cloud-based detectors: A black-box empirical study. *IEEE Transactions on Dependable and Secure Computing*, 18(4):1933–1949, 2021.

[104] Yanjie Li, Bin Xie, Songtao Guo, Yuanyuan Yang, and Bin Xiao. A survey of robustness and safety of 2d and 3d deep learning models against adversarial attacks. *ACM Computing Surveys*, 56(6):1–37, 2024.

[105] Aishan Liu, Xianglong Liu, Jiaxin Fan, Yuqing Ma, Anlan Zhang, Huiyuan Xie, and Dacheng Tao. Perceptual-sensitive gan for generating adversarial patches. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1028–1035, 2019.

[106] Anmin Liu, Weisi Lin, Manoranjan Paul, Chenwei Deng, and Fan Zhang. Just noticeable difference for images with decomposition model for separating edge and textured regions. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(11):1648–1652, 2010.

[107] Haochen Liu, Yiqi Wang, Wenqi Fan, Xiaorui Liu, Yaxin Li, Shaili Jain, Yunhao Liu, Anil Jain, and Jiliang Tang. Trustworthy ai: A computational perspective. *ACM Transactions on Intelligent Systems and Technology*, 14(1):1–59, 2022.

[108] Hui Liu, Bo Zhao, Minzhi Ji, Mengchen Li, and Peng Liu. Greedyfool: Multi-factor imperceptibility and its application to designing a black-box adversarial attack. *Information Sciences*, 613:717–730, 2022.

[109] Shengcai Liu, Ning Lu, Cheng Chen, Chao Qian, and Ke Tang. Hydratext: Multi-objective optimization for adversarial textual attack. *arXiv preprint arXiv:2111.01528*, abs/2111.01528, 2021.

[110] Shengcai Liu, Ning Lu, Cheng Chen, and Ke Tang. Efficient combinatorial optimization for word-level adversarial textual attack. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:98–111, 2021.

[111] Shengcai Liu, Ning Lu, Cheng Chen, and Ke Tang. Efficient combinatorial optimization for word-level adversarial textual attack. *IEEE ACM Transactions on Audio, Speech and Language Processing*, 30:98–111, 2022.

[112] Xinwei Liu, Jian Liu, Yang Bai, Jindong Gu, Tao Chen, Xiaojun Jia, and Xiaochun Cao. Watermark vaccine: Adversarial attacks to prevent watermark removal. In *European conference on computer vision*, pages 1–17, 2022.

[113] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, April 2017.

[114] Zhun-Ga Liu, Liang-Bo Ning, and Zuo-Wei Zhang. A new progressive multi-source domain adaptation network with weighted decision fusion. *IEEE Transactions on neural networks and learning systems*, 35(1):1062–1072, 2022.

[115] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 860–868, Long Beach, CA, June 2019.

[116] Teng Long, Qi Gao, Lili Xu, and Zhangbing Zhou. A survey on adversarial attacks in computer vision: Taxonomy, visualization and future directions. *Computers & Security*, 121:102847, 2022.

[117] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[118] Bo Luo, Yannan Liu, Lingxiao Wei, and Qiang Xu. Towards imperceptible and robust adversarial example attacks against neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[119] Cheng Luo, Qinliang Lin, Weicheng Xie, Bizhu Wu, Jinheng Xie, and Linlin Shen. Frequency-driven imperceptible adversarial attack on semantic similarity. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15315–15324, 2022.

[120] M Ronnier Luo, Guihua Cui, and Bryan Rigg. The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, 26(5):340–350, 2001.

[121] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, BC, Canada, April 2018.

[122] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, Virtual, April 2022.

[123] Anish Mittal, Anush K. Moorthy, and Alan C. Bovik. Blind/referenceless image spatial quality evaluator. In Michael B. Matthews, editor, *Asilomar Conference on Signals, Systems and Computers (ACSCC)*, pages 723–727, Pacific Grove, CA, November 2011.

[124] Hadi Mohaghegh Dolatabadi, Sarah Erfani, and Christopher Leckie. Advflow: Inconspicuous black-box adversarial attacks using normalizing flows. *Advances in Neural Information Processing Systems*, 33:15871–15884, 2020.

[125] Seungyong Moon, Gaon An, and Hyun Oh Song. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 4636–4645, Long Beach, California, June 2019.

[126] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1765–1773, 2017.

[127] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, Las Vegas, NV, June 2016.

[128] Konda Reddy Mopuri, Utkarsh Ojha, Utsav Garg, and R Venkatesh Babu. Nag: Network for adversary generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 742–751, 2018.

[129] Dongbin Na, Sangwoo Ji, and Jong Kim. Unrestricted black-box adversarial attack using gan with limited queries. In *European Conference on Computer Vision*, pages 467–482. Springer, 2022.

[130] Hanieh Naderi, Leili Goli, and Shohreh Kasaei. Generating unrestricted adversarial examples via three parameteres. *Multimedia Tools and Applications*, 81(15):21919–21938, 2022.

[131] Nina Narodytska and Shiva Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1310–1318, Honolulu, HI, July 2017.

[132] Liang-bo Ning, Zeyu Dai, Wenqi Fan, Jingran Su, Chao Pan, Luning Wang, and Qing Li. Joint universal adversarial perturbations with interpretations. *arXiv preprint arXiv:2408.01715*, 2024.

[133] Liang-bo Ning, Zeyu Dai, Jingran Su, Chao Pan, Luning Wang, Wenqi Fan, and Qing Li. Interpretation-empowered neural cleanse for backdoor attacks. In

Companion Proceedings of the ACM on Web Conference 2024, pages 951–954, 2024.

[134] Gherman Novakovsky, Nick Dexter, Maxwell W Libbrecht, Wyeth W Wasserman, and Sara Mostafavi. Obtaining genetics insights from deep learning via explainable artificial intelligence. *Nature Reviews Genetics*, 24(2):125–137, 2023.

[135] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning (ICML)*, volume 70, pages 2642–2651, Sydney, NSW, Australia, August 2017. PMLR.

[136] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018.

[137] Chao Pan, Qing Li, and Xin Yao. Adversarial initialization with universal adversarial perturbation: A new approach to fast adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21501–21509, 2024.

[138] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[139] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519, Abu Dhabi, United Arab Emirates, April 2017.

[140] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning

in adversarial settings. In *Proceedings of the IEEE European Symposium on Security and Privacy*, pages 372–387, Saarbrücken, Germany, March 2016.

[141] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *Proceedings of the 26th British Machine Vision Conference*, pages 41.1–41.12, Swansea, UK, September 2015.

[142] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4422–4431, 2018.

[143] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10609–10619, New Orleans, LA, June 2022.

[144] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *International Conference on Machine Learning (ICML)*, volume 139, pages 8748–8763, Virtual, July 2021. PMLR.

[145] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, Las Vegas, NV, June 2016.

[146] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[147] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.

[148] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, New Orleans, LA, June 2022.

[149] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, New Orleans, LA, June 2022.

[150] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015.

[151] Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4322–4330, 2019.

[152] Daniel L Ruderman, Thomas W Cronin, and Chuan-Chin Chiao. Statistics of cone responses to natural images: implications for visual coding. *Journal of the Optical Society of America A*, 15(8):2036–2045, 1998.

[153] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[154] JL Schnapf, TW Kraft, and DA Baylor. Spectral sensitivity of human cone photoreceptors. *Nature*, 325(6103):439–441, 1987.

[155] Alexander Schrijver. *Theory of linear and integer programming*. Wiley, 1999.

[156] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the 2017 IEEE International Conference on Computer Vision*, pages 618–626, Venice, Italy, October 2017.

[157] Ali Shahin Shamsabadi, Ricardo Sanchez-Matilla, and Andrea Cavallaro. Colorfool: Semantic adversarial colorization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1151–1160, 2020.

[158] Mahmood Sharif, Lujo Bauer, and Michael K. Reiter. On the suitability of lp-norms for creating and preventing adversarial examples. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1605–1613, Salt Lake City, UT, June 2018.

[159] 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 *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016.

[160] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[161] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proceedings of the 2nd International Conference on Learning Representations*, Banff, AB, Canada, April 2014.

[162] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, May 2015.

[163] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[164] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*, volume 37, pages 2256–2265, Lille, France, July 2015.

[165] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, Virtual, May 2021.

[166] Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 8322–8333, Montréal, Canada, December 2018.

[167] 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.

[168] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012.

[169] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

[170] Akshayvarun Subramanya, Vipin Pillai, and Hamed Pirsiavash. Fooling network interpretation in image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2020–2029, 2019.

[171] Qi Sun, Arjun Ashok Rao, Xufeng Yao, Bei Yu, and Shiyan Hu. Counteracting adversarial attacks in autonomous driving. In *Proceedings of the IEEE/ACM International Conference On Computer Aided Design*, pages 83:1–83:7, San Diego, CA, November 2020.

[172] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, Las Vegas, NV, June 2016.

[173] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proceedings of the 2nd International Conference on Learning Representations*, Banff, AB, Canada, April 2014.

[174] Guanhong Tao, Shiqing Ma, Yingqi Liu, and Xiangyu Zhang. Attacks meet interpretability: Attribute-steered detection of adversarial samples. *Advances in neural information processing systems*, 31, 2018.

[175] Chun-Chen Tu, Pai-Shun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based ze-

roth order optimization method for attacking black-box neural networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 742–749, Honolulu, Hawaii, January 2019.

[176] Fatemeh Vakhshiteh, Ahmad Nickabadi, and Raghavendra Ramachandra. Adversarial attacks against face recognition: A comprehensive study. *IEEE Access*, 9:92735–92756, 2021.

[177] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[178] Astha Verma, A Venkata Subramanyam, and Rajiv Ratn Shah. Wasserstein metric attack on person re-identification. In *2022 IEEE 5th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 234–239. IEEE, 2022.

[179] Viet Quoc Vo, Ehsan Abbasnejad, and Damith C. Ranasinghe. Query efficient decision based sparse attacks against black-box machine learning models. 2022.

[180] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 856–865, 2018.

[181] Donghua Wang, Wen Yao, Tingsong Jiang, and Xiaoqian Chen. Improving transferability of universal adversarial perturbation with feature disruption. *IEEE Transactions on Image Processing*, 33:722–737, 2023.

[182] Huan Wang, Ziwen Cui, Ruigang Liu, Lei Fang, and Ying Sha. A multi-type transferable method for missing link prediction in heterogeneous social networks. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[183] Jingyuan Wang, Yufan Wu, Mingxuan Li, Xin Lin, Junjie Wu, and Chao Li. Interpretability is a kind of safety: An interpreter-based ensemble for adversary

defense. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 15–24, 2020.

[184] Run Wang, Felix Juefei-Xu, Qing Guo, Yihao Huang, Xiaofei Xie, Lei Ma, and Yang Liu. Amora: Black-box adversarial morphing attack. In *Proceedings of the 28th ACM international conference on multimedia*, pages 1376–1385, 2020.

[185] Xiaosen Wang, Kun He, Chuanbiao Song, Liwei Wang, and John E Hopcroft. At-gan: An adversarial generator model for non-constrained adversarial examples. *arXiv preprint arXiv:1904.07793*, 2019.

[186] Yajie Wang, Shangbo Wu, Wenyi Jiang, Shengang Hao, Yu-an Tan, and Quanxin Zhang. Demiguise attack: Crafting invisible semantic adversarial perturbations with perceptual similarity. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pages 3125–3133, Virtual Event / Montreal, Canada, August 2021.

[187] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[188] Zihan Wang, Olivia Byrnes, Hu Wang, Ruoxi Sun, Congbo Ma, Huaming Chen, Qi Wu, and Minhui Xue. Data hiding with deep learning: A survey unifying digital watermarking and steganography. *IEEE Transactions on Computational Social Systems*, 10(6):2985–2999, 2023.

[189] Wenqi Wei and Ling Liu. Trustworthy distributed ai systems: Robustness, privacy, and governance. *ACM Computing Surveys*, 2024.

[190] Zhipeng Wei, Jingjing Chen, Xingxing Wei, Linxi Jiang, Tat-Seng Chua, Fengfeng Zhou, and Yu-Gang Jiang. Heuristic black-box adversarial attacks on video recognition models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12338–12345, 2020.

[191] Michael Werman, Shmuel Peleg, and Hillel Rom. A unified approach to the change of resolution: Space and grey level. In *Image Processing, Analysis, Measurement, and Quality*, volume 901, pages 91–104. SPIE, 1988.

[192] Eric Wong, Frank Schmidt, and Zico Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. In *International conference on machine learning*, pages 6808–6817. PMLR, 2019.

[193] Jiahao Wu, Wenqi Fan, Jingfan Chen, Shengcai Liu, Qing Li, and Ke Tang. Disentangled contrastive learning for social recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM)*, pages 4570–4574, 2022.

[194] Jiahao Wu, Ning Lu, Zeiyu Dai, Wenqi Fan, Shengcai Liu, Qing Li, and Ke Tang. Backdoor graph condensation. *arXiv preprint arXiv:2407.11025*, 2024.

[195] Qiucheng Wu, Yujian Liu, Handong Zhao, Ajinkya Kale, Trung Bui, Tong Yu, Zhe Lin, Yang Zhang, and Shiyu Chang. Uncovering the disentanglement capability in text-to-image diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1900–1910, Vancouver, BC, Canada, June 2023.

[196] Tao Xiang, Hangcheng Liu, Shangwei Guo, Yan Gan, and Xiaofeng Liao. EGM: an efficient generative model for unrestricted adversarial examples. *ACM Trans. Sens. Networks*, 18(4):51:1–51:25, 2022.

[197] Tao Xiang, Hangcheng Liu, Shangwei Guo, Tianwei Zhang, and Xiaofeng Liao. Local black-box adversarial attacks: A query efficient approach. *arXiv preprint arXiv:2101.01032*, abs/2101.01032, 2021.

[198] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *International Conference on Learning Representations*, 2018.

[199] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1369–1378, 2017.

[200] 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 *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, Long Beach, CA, June 2019.

[201] Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K Jain. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 17:151–178, 2020.

[202] Kaidi Xu, Sijia Liu, Gaoyuan Zhang, Mengshu Sun, Pu Zhao, Quanfu Fan, Chuang Gan, and Xue Lin. Interpreting adversarial examples by activation promotion and suppression. *arXiv preprint arXiv:1904.02057*, 2019.

[203] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In *Computer vision–ECCV 2020: 16th European conference, glasgow, UK, August 23–28, 2020, proceedings, part v 16*, pages 665–681. Springer, 2020.

[204] Shige Xu, Lei Zhang, Yin Tang, Chaolei Han, Hao Wu, and Aiguo Song. Channel attention for sensor-based activity recognition: embedding features into all frequencies in dct domain. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[205] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Proceedings of the 25th Annual Network and Distributed System Security Symposium*, San Diego, California, February 2018.

[206] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, California, February 2018.

[207] Karren Yang, Wan-Yi Lin, Manash Barman, Filipe Condessa, and Zico Kolter. Defending multimodal fusion models against single-source adversaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3340–3349, 2021.

[208] Xiaokang Yang, WS Ling, ZK Lu, Ee Ping Ong, and SS Yao. Just noticeable distortion model and its applications in video coding. *Signal processing: Image communication*, 20(7):662–680, 2005.

[209] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, page 100211, 2024.

[210] Jia-Li Yin, Menghao Chen, Jin Han, Bo-Hao Chen, and Ximeng Liu. Adversarial example quality assessment: A large-scale dataset and strong baseline. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 4786–4794, 2024.

[211] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019.

[212] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proceedings of the 13th European Conference on Computer*

Vision, Part I, volume 8689, pages 818–833, Zurich, Switzerland, September 2014.

[213] Hanwei Zhang. *Deep Learning in Adversarial Context*. PhD thesis, École normale supérieure de Rennes, 2021.

[214] Hanwei Zhang, Yannis Avrithis, Teddy Furon, and Laurent Amsaleg. Smooth adversarial examples. *EURASIP Journal on Information Security*, 2020:15, 2020.

[215] Hanwei Zhang, Yannis Avrithis, Teddy Furon, and Laurent Amsaleg. Walking on the edge: Fast, low-distortion adversarial examples. *IEEE Transactions on Information Forensics and Security*, 16:701–713, 2020.

[216] Lilin Zhang, Ning Yang, Yanchao Sun, and Philip S. Yu. Provable unrestricted adversarial training without compromise with generalizability. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12):8302–8319, 2024.

[217] Qinglong Zhang, Lu Rao, and Yubin Yang. A novel visual interpretability for deep neural networks by optimizing activation maps with perturbation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3377–3384, 2021.

[218] Xinyang Zhang, Ningfei Wang, Hua Shen, Shouling Ji, Xiapu Luo, and Ting Wang. Interpretable deep learning under fire. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020.

[219] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742, 2021.

[220] Zuowei Zhang, Zhunga Liu, Liangbo Ning, Arnaud Martin, and Jiexuan Xiong. Representation of imprecision in deep neural networks for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[221] Zuowei Zhang, Liangbo Ning, Zechao Liu, Qingyu Yang, and Weiping Ding. Mining and reasoning of data uncertainty-induced imprecision in deep image classification. *Information Fusion*, 96:202–213, 2023.

[222] Ting Zhao and Xiangqian Wu. Pyramid feature attention network for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3085–3094, Long Beach, CA, June 2019.

[223] Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Chongxuan Li, Ngai-Man Man Cheung, and Min Lin. On evaluating adversarial robustness of large vision-language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[224] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. Towards large yet imperceptible adversarial image perturbations with perceptual color distance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1039–1048, 2020.

[225] Zhengyu Zhao, Zhuoran Liu, and Martha A. Larson. Towards large yet imperceptible adversarial image perturbations with perceptual color distance. In *Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1036–1045, Seattle, June 2020.

[226] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, Las Vegas, NV, June 2016.

[227] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16, 2003.

[228] Peifei Zhu, Tsubasa Takahashi, and Hirokatsu Kataoka. Watermark-embedded adversarial examples for copyright protection against diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24420–24430, 2024.