AdvDenoise: Fast Generation Framework of Universal and Robust Adversarial Patches Using Denoise

Jing Li[1], Zigan Wang[1,2], Jinliang Li[1*]
[1]School of Economics and Management, Tsinghua University, Beijing, China
[2]Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
lijing3@sem.tsinghua.edu.cn
wangzigan@sz.tsinghua.edu.cn, jll@tsinghua.edu.cn

Abstract

Adversarial patch attacks, which can mislead deep learning models and the human eye in both the digital and physical domains, have led to a trust crisis. Traditional approaches to generating powerful attack patches require extensive, multi-scenario data, but suffer from slow search speeds in adversarial gradient space, resulting in low global attack success rates and high costs. Especially high resource-consuming attack methods are not sufficient to pose sufficient threats, which leads to the vulnerability of defense. To address these challenges, we present a novel framework ***AdvDenoise*** *to generate universal adversarial patches fast and robustly using denoise. Concretely, we leverage the power of denoising diffusion probabilistic models to craft or optimize these patches, deviating from traditional pure gradient-based methods. We conduct comprehensive experiments on both pre-trained convolutional neural networks and vision transformer detectors, evaluating our method on standard benchmarks as well as in simulated real-world physical settings. The results demonstrate that our framework outperforms strong baselines, achieving higher attack success rates, better transferability across models, and improved robustness to transformations while maintaining visual realism and computational efficiency. When our method's performance approaches the state-of-the-art, the total time required to generate 100-shots adversarial patches is substantially lower than the state-of-the-art methods, with a remarkable 48.15% reduction in time complexity. The code and examples are publicly available at* https://github.com/advdenoise/advdenoise.

* Corresponding author.

1. Introduction

Adversarial Patches (APs) have emerged as potent tools to generate adversarial examples, significantly impacting deep learning models on a variety of computer vision tasks [2, 29, 39]. Recent developments in APs have used natural phenomena, such as lighting [46] and shadows [48], or employed mimetic camouflage strategies [22, 37, 40, 45] to deceive detectors in both Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), even often evading human eye detection. This concept harkens back to Szegedy *et al.* [38] seminal 2014 work, where they first defined adversarial examples, underscoring the evolving nature of attacks in the field. That means a perfect adversarial attack should fool not only the machine learning detector but also human eyes at the same time.

APs have significantly advanced adversarial Attack Success Rates (ASR) with the continuous efforts of the community in recent years, thereby undermining trust in deep learning. However, their creation is hampered by the vast search space required in traditional gradient-based [1, 20, 34, 39] or Generative Adversarial Networks (GANs) [7, 15, 26] training processes. Specifically, in real-world applications like autonomous driving and surveillance, the resource-intensive nature of generating APs often results in a lag behind the evolving defenses of machine learning systems. We know that in cyber attacks, according to MITRE ATT&CK [36], attack speed is often the key to success or failure. Therefore, the challenge of efficiently generating effective robust APs remains a critical area of inquiry.

Diffusion Models (DMs) [14] have recently emerged as a powerful class of likelihood-based generative models, posing a formidable challenge to the dominance of GANs. DMs exhibit several desirable properties, including increased training stability and improved coverage of the data distribution [6, 32]. At their core, DMs comprise two distinct processes: a forward diffusion process and a reverse generation process. The forward diffusion process

gradually corrupts the input data by introducing Gaussian noise, eventually transforming it into pure noise. On the contrary, the reverse generation process aims to recover the original data from this noise through a denoising-like technique. A well-trained diffusion model is remarkably capable of generating high-quality images from random noise inputs, showcasing its generative prowess.

Given that the demonstrated superiority of denoising diffusion probabilistic models [14] and their variants [6, 10, 21, 32] in image generation tasks suggests a viable approach to the challenges previously outlined. Recent works [3–5, 25, 27, 30, 44, 45, 47] have demonstrated that injecting adversarial noise using DMs to mislead deep learning models is becoming increasingly sophisticated. However, while these approaches normally focus on improving attack success rates, transferability, and resistance to purification, they often overlook the potential to leverage fast diffusion-based solvers for accelerated APs generation. In response, DMs have been observed to generate universal and robust APs, indicating a higher likelihood of successful attacks or transfers on CNNs and ViTs.

To bridge the gaps in existing adversarial patch generation methods, we introduce AdvDenoise, a novel framework that harnesses the power of denoising diffusion models to enable fast and effective adversarial patch creation. Unlike typical end-to-end approaches [5, 22, 40], AdvDenoise employs an innovative strategy that generates adversarial patches through a diffusion process that starts from intermediate patch stages, similar to a fission process.

This fission process offers several distinct advantages over traditional methods. First, it promotes a more uniform framework for adversarial patch generation, mitigating issues such as the collapse of the optimization process due to mismatched attack responses across different models or datasets. By starting from intermediate patch stages and gradually refining patches through a diffusion process, AdvDenoise can effectively navigate the complex adversarial landscape and generate robust and transferable patches.

And, the fission-like diffusion process employed by AdvDenoise also allows for greater flexibility and control over the patch generation process. By starting from intermediate stages, researchers and practitioners can selectively guide the diffusion process towards desired patch characteristics, such as visual inconspicuousness, transferability across models, or robustness to various transformations. This level of control is particularly valuable in scenarios where specific attack requirements or constraints need to be addressed.

Furthermore, AdvDenoise opens up new avenues for incorporating additional prior knowledge or constraints into the patch generation process. By conditioning the diffusion model on relevant information, such as the target model's architecture, dataset characteristics, or deployment environments, AdvDenoise can generate tailored adversarial patches that are optimized for specific use cases or threat models.

Through this innovative fusion of diffusion models and adversarial patch generation, AdvDenoise aims to push the boundaries of adversarial machine learning research, enabling the rapid and efficient crafting of robust adversarial patches. By addressing the limitations of existing methods and leveraging the powerful generative capabilities of diffusion models, AdvDenoise has the potential to advance our understanding of model vulnerabilities and drive the development of more secure and trustworthy computer vision systems.

In a nutshell, our contributions are threefold:

• We introduce the novel AdvDenoise framework, which utilizes a fast solver for the denoising of diffusion models to outperform current methods in terms of speed and cost in multiple attack vectors.

• We propose a fast fission method based on denoise for the generation of adversarial patches, employing the diffusion of adversarial characteristics and selective patch retention to achieve effective results without the complexity of adversarial composite examples.

• We validate our approach through a series of robustness and efficiency experiments on a dedicated simulator.

The code and examples of AdvDenoise will be available at https://github.com/advdenoise/advdenoise.

2. Related Work

In this section, we briefly introduce some recent developments in adversarial patches and diffusion models.

2.1. Adversarial Patches

For physical domain adversarial attacks, universal perturbations in the digital domain [38] often prove ineffective due to real-world factors such as rotation, lighting variations [46], and shadows [48]. Adversarial patches, which involve adding or sticking carefully crafted patches onto the target objects, have emerged as a common and effective technique for causing misclassifications in machine learning systems [2, 15, 29, 39].

Early work on adversarial patches relied on optimization-based approaches, iteratively updating the patch to maximize the loss of the target model [2]. However, these methods are computationally expensive and often require a large number of iterations to converge. More recent work has explored data-driven approaches, training generative models to synthesize adversarial patches directly [22, 29, 40]. Although these methods can generate patches more efficiently, they typically require a significant amount of training data and computational resources.

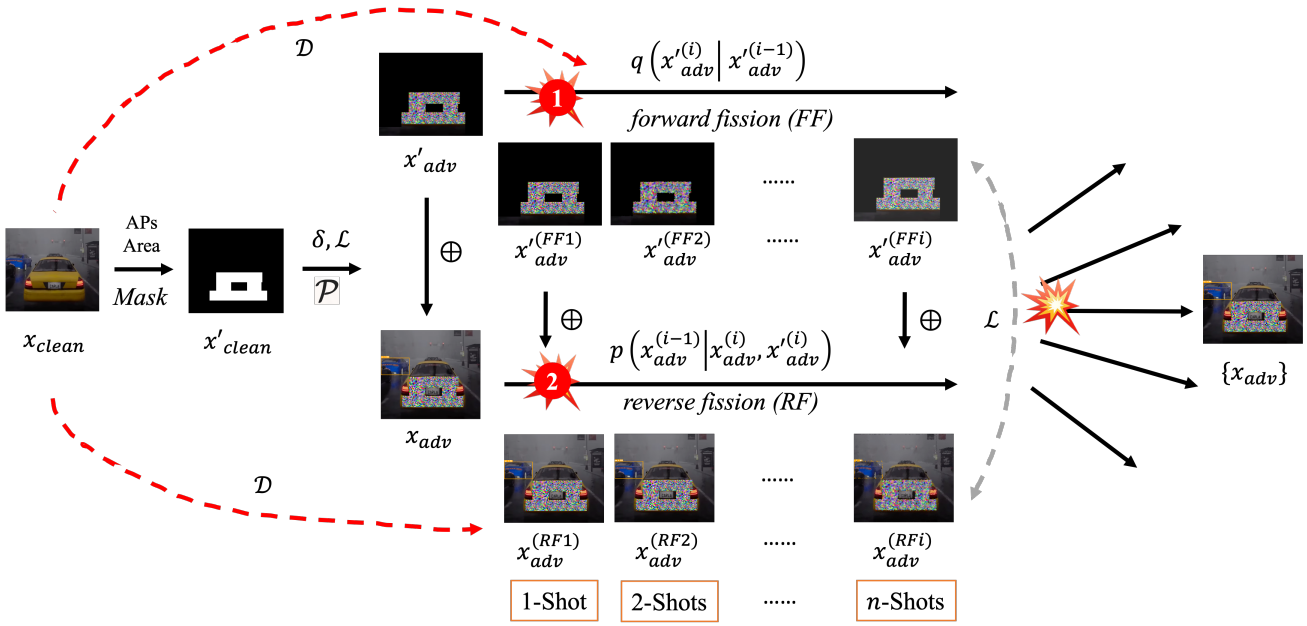

Figure 1. The core architecture of AdvDenoise framework. The three explosive emoji represent ① the split fission of the patch itself, ② the diffusion synthesis of the APs, and the collection of a set of APs (far right). The red and gray dotted line represents the return direction of the gradient descent in different periods. $\bigoplus$ represents the image overlay add.

One limitation of existing adversarial patch methods is that they are primarily designed for static scenarios, where the patch is applied to a single image or object. In real-world settings, however, the patch may be subject to various transformations, such as rotations, occlusions, or changes in viewpoint, which can impact its effectiveness. Moreover, most previous methods [15, 20, 39] focus on generating patches that cause misclassification at a single point in time, without considering the temporal aspect of an attack, where the adversary may need to maintain the misclassification over an extended period.

2.2. Diffusion Models

Diffusion models [6, 10, 14, 32, 33] have recently gained significant attention due to their impressive performance in generative tasks. These models operate by gradually adding Gaussian noise to the input data during a forward diffusion process, and then learning to reverse this process by training a denoising model to predict the added noise at each step, ultimately reconstructing the original data.

The forward diffusion process is defined as a Markov chain of gradually adding Gaussian noise $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; 0, \mathbf{I})$ to the data $\mathbf{x}_0$, resulting in a sequence of noisy latent variables $\{\mathbf{x}_1, ..., \mathbf{x}_{N-1}\}$, where T is the total number of diffusion steps. The forward process is parameterized by a variance schedule $\{\beta_1, ..., \beta_N\}$, where $\beta_t \in (0, 1)$ determines the amount of noise added at each step,

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

The reverse denoising process aims to learn the reverse transition from noise to data by modeling the conditional distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, where θ represents the parameters of the denoising model. The goal is to iteratively denoise the noisy latent variables $\mathbf{x}_T$ to recover the original data $\mathbf{x}_0$.

The denoising model is trained by optimizing the following objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{q(\mathbf{x}0),\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left|\left|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\right|\right|^2$$

where $\epsilon_\theta(\cdot, t)$ is the denoising model that predicts the noise ϵ at time step t, and $\bar{\alpha}_t$ is the cumulative product of the variance schedule. During the sampling process, the denoising model is applied iteratively to progressively denoise the initial noise $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$, generating samples from the data distribution $p(\mathbf{x}_0)$.

The success of diffusion models in various domains, such as image synthesis [32], and text generation [10], has motivated researchers to explore their applications in adversarial machine learning. Recent works have demonstrated the potential of diffusion models for both attacking [3, 18, 25, 30, 44, 45, 49] and defending [5, 24, 31, 42, 43, 47] against adversarial examples.

Despite these promising applications, the potential of diffusion models in the context of adversarial patches remains largely unexplored. The ability of diffusion models

to generate high-quality samples while maintaining control over specific attributes or regions of the output could prove valuable for crafting effective adversarial patches. Additionally, the iterative nature of the diffusion process may allow for the generation of dynamic adversarial patches that can adapt to changes in the target object or environment over time.

In this work, our aim is to bridge this gap by investigating the use of diffusion models for generating adversarial patches tailored to dynamic real-world scenarios. Our approach leverages the strengths of diffusion models in controlled generation, while addressing the limitations of existing adversarial patch methods in handling temporal and transformational aspects of the attack.

3. Methodology

In this section, we introduce a novel Adversarial Patches Fast Generation Framework Using Denoise termed **AdvDenoise** as illustrated in Figure. 1. Our approach begins with a clear definition of the problem and subsequently introduces two phases, including forward fission and reverse fission, together facilitating robust and fast adversarial patches generation using denoise in complex scenarios.

3.1. Problem Definition

Let $\boldsymbol{x_{clean}}$ be the clean input (e.g., object image) to the machine learning system, $\mathcal{D}$ be the detector function (e.g., CNNs or ViTs), and $\mathcal{P}$ be the random patch rendering function. First, the area where $\boldsymbol{x_{clean}}$ needs a patch is isolated by a mask. A perturbation $\delta \sim \mathcal{N}(0,1)$ drawn from a standard normal distribution is added to $\boldsymbol{x_{clean}}$, resulting in $\boldsymbol{x'_{clean}}$. This perturbed input is then passed through $\mathcal{P}$ to generate $\boldsymbol{x'_{adv}}$, such that $\mathcal{D}(\boldsymbol{x_{clean}} \bigoplus \boldsymbol{x'_{adv}}) \neq \boldsymbol{y}$, where $\boldsymbol{y}$ represents the true label.

The notation $\bigoplus$ denotes the operation of overlaying the adversarial patch $\boldsymbol{x'_{adv}}$ onto the clean input $\boldsymbol{x_{clean}}$. By applying $\mathcal{P}$ with different random values of δ, we can generate a naive adversarial input $\boldsymbol{x_{adv}}$.

Let $\mathcal{L}$ be a loss function applied to $\mathcal{D}$ that satisfies the above inequality, encouraging the detector to misclassify the adversarial input. We introduce the $FFi, (i = 1, 2, ..., n)$ to represent the diffusion forward fission process, which generates a series of intermediate patches, and RFi to represent the reverse fission process. By solving the following optimization problem (Equation. 1), we can generate a series of final adversarial patches $\boldsymbol{x}_{\boldsymbol{adv}}^{(\boldsymbol{RF_i})}$.

$$\boldsymbol{x}_{\boldsymbol{adv}}^{(\boldsymbol{RF_i})} = \arg\max_{\boldsymbol{x}_{adv}} \mathcal{L}(\mathcal{D}(\boldsymbol{x_{clean}} \bigoplus \boldsymbol{x}_{\boldsymbol{adv}}^{\boldsymbol{\prime(FF_i)}}, \delta), \boldsymbol{y}) \quad (1)$$

3.2. Denoising Adversarial Attack Framework

The overview architecture of AdvDenoise is illustrated in Figure. 1. The procedures for the adversarial patches (APs) area mask, adversarial synthesis, including the 3D rendering, and loss function, follow previous work [22, 37, 40]. In a large number of experiments, we perceptually found that there exists a certain attack boundary around the final APs, which means that local feature diffusion from the final APs has a higher probability of finding a set of effective APs. Thus, we propose the forward and reverse fission (FF and RF) processes, which make up the denoise-based fast fission using the diffusion model.

Our training data comprises images sampled from a photo-realistic simulator CARLA [8] under varying vehicles settings, capturing diverse scenarios. We leverage a pre-trained image segmentation network [16] to isolate the target vehicles, generating binary masks. Concurrently, we render camouflage textures onto the vehicle surfaces within the same simulated environments, creating the 2D camouflage. We iteratively optimize the adversarial camouflage patches through diffusion-based fission and backpropagation in different periods, guided by our tailored loss function (Equation. 1) that balances visual inconspicuousness with the ability to evade detection or misclassification by computer vision models.

The use of diffusion models in AdvDenoise enables faster and more efficient attacks compared to iterative optimization techniques. Diffusion models, known for their stable training and high-quality sample generation capabilities, can leverage their learned representations to rapidly produce adversarial patches from noise inputs. This contrasts with traditional gradient-based methods, which often suffer from slow convergence and computational inefficiencies due to the need for multiple forward and backward passes through the target model.

By leveraging the above framwork workflow, AdvDenoise can efficiently explore the local feature space around the final APs and generate a diverse set of candidate APs. This not only improves the attack success rate but also enhances the robustness of the generated APs against various transformations and environmental conditions, as the diffusion model learns to capture the underlying distribution of effective APs.

3.3. Denoise-based Fast Fission

Previous works [2, 5, 22, 37, 40] use end-to-end method to generate APs are very resource consuming. Instead, we adopt DMs as accelerator because we believe that the denoise-based fission (FF and RF) will quickly spread a series of patch sets across an adversarial range. The experimental results in Figure. 2 show the effectiveness of AdvDenoise.

The forward fission process, denoted as FF_i, takes the

final APs $\boldsymbol{x_{adv}}$ as input and progressively adds Gaussian noise to generate a series of intermediate noisy patches $\boldsymbol{x_{adv}^{(FF_i)}}$, where N is the number of diffusion adversarial steps. And the reverse fission process, denoted as RF_i, aims to denoise the intermediate noisy patches $\boldsymbol{x_{adv}^{(FF_i)}}$ to generate a set of candidate APs $\boldsymbol{x_{adv}^{(RF_i)}}$.

Once a possible patch is obtained, we leverage the iterative forward and reverse fission processes to efficiently explore the local feature space around the initial patch and generate an improved set of adversarial patches. In the forward process, we apply a probability variation space at each step, effectively performing an iterative optimization procedure to find better APs rapidly. These processes can be described using conditional probability distributions as shown in Equation. 2 and Equation. 3, respectively.

$$\begin{aligned} FFi &= q\left(x_{adv}^{\prime(i)} | x_{adv}^{\prime(i-1)}\right) \\ &= \mathcal{N}\left(x_{adv}^{\prime(i)}; \mu_f(x_{adv}^{\prime(i-1)}), \beta I\right) \\ \mu_f(x_{adv}^{\prime(i-1)}) &= \sqrt{1-\beta}(x_{adv}^{\prime(i-1)} \\ &\quad + \alpha \nabla_{x'_{adv}} \mathcal{L}(\mathcal{D}((\boldsymbol{x_{clean}} \bigoplus \boldsymbol{x_{adv}^{\prime(i-1)}}), \boldsymbol{y}))) \end{aligned} \tag{2}$$

$$\begin{aligned} RFi &= p\left(x_{adv}^{(i-1)} | x_{adv}^{(i)}, x_{adv}^{\prime(i)}\right) \\ &= \mathcal{N}\left(x_{adv}^{(i-1)}; \mu_r(x_{adv}^{(i)}, x_{adv}^{\prime(i)}), \beta I\right) \\ \mu_r(x_{adv}^{(i)}, x_{adv}^{\prime(i)}) &= \sqrt{1-\beta}(x_{adv}^{(i)} \\ &\quad + \alpha \nabla_{x_{adv}} \mathcal{L}(\mathcal{D}((\boldsymbol{x_{adv}^{(i)}} \bigoplus \boldsymbol{x_{adv}^{\prime(i)}}), \boldsymbol{y}))) \end{aligned} \tag{3}$$

where I is the identity matrix, β is an adjustable parameter, ∇ represents the backpropagation gradient vector, and $\bigoplus$ denotes the image overlay operation.

In the forward fission process (Equation. 2), we iteratively add Gaussian noise to the previous step's adversarial patch $x_{adv}^{\prime(i-1)}$ to obtain a noisy intermediate patch $x_{adv}^{\prime(i)}$. The mean of this Gaussian distribution, $\mu_f(x_{adv}^{\prime(i-1)})$, is computed by taking a weighted sum of the previous patch and the gradient of the loss function with respect to the adversarial patch, scaled by a factor α. This gradient term encourages the generation of patches that maximize the misclassification of the target object.

In the reverse fission process (Equation. 3), we aim to denoise the intermediate noisy patch $x_{adv}^{\prime(i)}$ to obtain a candidate adversarial patch $x_{adv}^{(i-1)}$. The mean of this conditional distribution, $\mu_r(x_{adv}^{(i)}, x_{adv}^{\prime(i)})$, is computed similarly to the forward process, but with an additional term $x_{adv}^{(i)}$ representing the target clean input overlaid with the noisy patch. This term ensures that the generated candidate patch retains the adversarial properties while maintaining visual realism.

The candidate APs $\boldsymbol{x_{adv}^{(RF_i)}}$ are then evaluated using the loss function described in Equation. 1, and the AP with the highest adversarial effectiveness and visual realism is selected as the final output. By iterating through these forward and reverse fission processes, AdvDenoise can efficiently explore the local feature space and generate diverse and effective adversarial patches tailored to the target object and environment.

The pseudo-code of AdvDenoise is shown in Algorithm. 1.

Algorithm 1 : AdvDenoise

Input: clean object image: $\boldsymbol{x_{clean}}$; detectors from CNNs or ViTs: $\mathcal{D}$; random noise patch generator $\mathcal{P}$

Output: A set of $\{\boldsymbol{x_{adv}}\}$ obtained from fission

Phase I: Forward Fission

1: $\boldsymbol{x'_{clean}} \leftarrow Mask(\boldsymbol{x_{clean}})$
2: $\boldsymbol{x'_{adv}} \leftarrow P(\boldsymbol{x'_{clean}})$
3: **if** $\mathcal{D}(\boldsymbol{x_{clean}} \bigoplus \boldsymbol{x'_{adv}}) \neq \boldsymbol{y}$ # Calculate by Equation. 1 **then**
4: **for** $i = 1, 2, ..., n$ **do**
5: $\boldsymbol{x_{adv}^{\prime(FFi)}} \leftarrow \boldsymbol{x'_{adv}}$ # Calculate by Equation. 2
6: **end for**
7: **else**
Return to Line 2.
8: **end if**

Phase II: Reverse Fission

1: **for** $i = 1, 2, ..., n$ **do**
2: $\boldsymbol{X_{adv}^{\prime(RFi)}} \leftarrow \boldsymbol{x_{clean}} \bigoplus \boldsymbol{x_{adv}^{\prime(FFi)}}$
3: **if** $\boldsymbol{X_{adv}^{\prime(RFi)}} \neq \boldsymbol{y}$ # Calculate by Equation. 1 **then**
4: $\boldsymbol{x_{adv}^{(RFi)}} \leftarrow \boldsymbol{x_{adv}^{\prime(RFi)}}$ # Calculate by Equation. 3
5: $\{\boldsymbol{x_{adv}}\} \leftarrow \boldsymbol{x_{adv}^{(RFi)}}$
6: **else**
Break. # Get series of AdvDenoise APs.
7: **end if**
8: **end for**

4. Experiments

In this section, we present the results of our comprehensive experiments conducted on three popular metrics. We compare the performance of our proposed method with 6 target models and 4 state-of-the-art adversarial patches attack approaches. To gain a deeper understanding of the key components contributing to the effectiveness of our AdvDenoise framework, we conduct a ablation study.

4.1. Experimental Settings

Datasets and Competitors. We evaluate our AdvDenoise framework on the CARLA simulator v0.9.14 [8], which provides a rich and diverse set of scenarios for testing adversarial patch attacks in simulated physical domains. For the training and evaluation of our method, we leverage

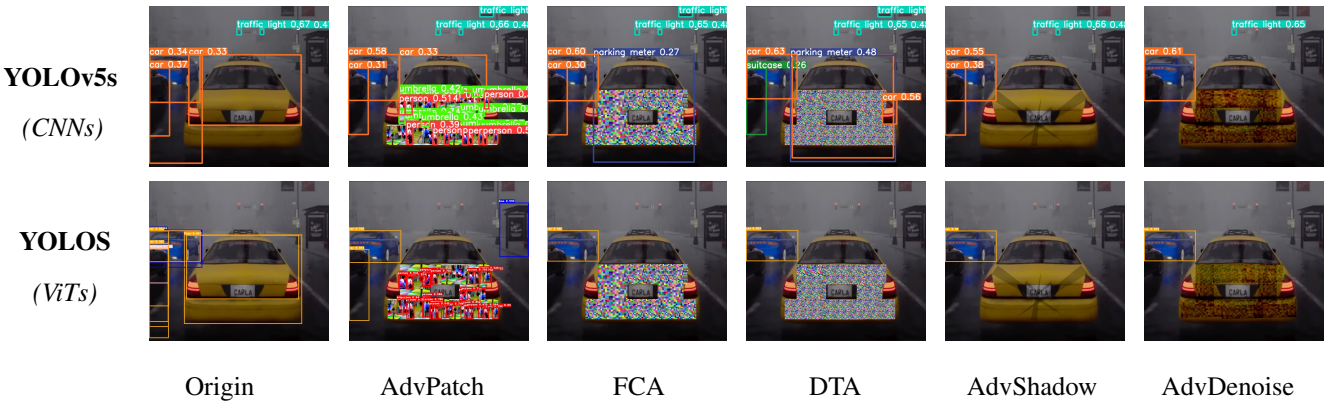

Figure 2. APs visual comparison results among of different approaches under classic representatives of CNNs and ViTs.

two high-resolution image datasets which generated by the CARLA using default settings. The training set comprises 50,000 carefully curated images, while the testing set consists of 15,000 distinct images. These datasets are designed to capture the diversity and complexity of real-world scenarios by sampling images from a wide range of vehicles and distances.

For fair comparisons, we focus on vehicle-based adversarial patches in a multi-scenario physical domain setting. We choose AdvPatch [39], FCA [40], DTA [37], and AdvShadow [48] as competing methods. Although these methods are designed to attack object detection models, there can be significant variability in their performance due to different experimental settings. To ensure a fair evaluation, we carefully identified localization areas that maximize the capabilities of each competitor in a unified scenario through extensive testing.

Target Models. Following the comprehensive survey by Zou et al. [50] and considering the practical relevance of adversarial attacks, we selected a diverse set of target models for our experiments. These include Mask RCNN [13], SSD [28], YOLOv5s [17], YOLOS [11], ViDT [35], and CFDT [23], encompassing both Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs). And these models are the official implementation PyTorch version.

Implementation Details. To verify the effectiveness of AdvDenoise, the hyperparameters are set as follows, we set the batch size, initial learning rate, and dropout rate to 32, 1e-5, and 0.4, respectively, for the attack setting. We employ the Adam optimizer [19] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, the max epoch is 5, and early stopping for the input. All experiments were conducted on a server equipped with 4 * NVIDIA GeForce RTX 3090 GPUs.

4.2. Evaluation Metrics

Attack Success Rate (ASR). This evaluation metric, commonly used in previous works [2, 22], is defined as the percentage of target objects that were correctly detected before the perturbation, but were not detected or falsely detected after applying the adversarial patch. The higher values indicating more powerful adversarial attack. The ASR is defined as the percentage of input images for which the application of the adversarial patch successfully causes the target model to produce an incorrect prediction or classification. It can be calculated as follows, $ASR = (N_{adv})/(N_{input}) * 100\%$, where N_{adv}, N_{input} means number of successful adversarial attacks and total number of input images, respectively.

Structural Similarity Index Measure (SSIM). To quantitatively assess the visual similarity between the generated adversarial patches $\boldsymbol{x_{adv}}$ and the original clean images $\boldsymbol{x_{clean}}$, we employ the SSIM. SSIM [9, 12, 41] is a widely-adopted metric that objectively quantifies the perceived visual distortion between two images by considering their luminance, contrast, and structural information. The SSIM index is calculated as, $SSIM(x_{adv}, x_{clean}) = [l(x_{adv}, x_{clean})]^{\alpha} * [c(x_{adv}, x_{clean})]^{\beta} * [s(x_{adv}, x_{clean})]^{\gamma}$, where l, c, s represents luminance, contrast and structural similarity respectively. The higher values indicating greater structural similarity between the two images. A value of 1 signifies that the two images are identical, while lower values indicate increasing dissimilarity.

Time and Cost. We measure the training time required for generating adversarial patches, starting from initializing the random patch noise. The computational cost is estimated based on the pricing structure of Amazon Web Services (AWS) in 2023†.

†https://aws.amazon.com/pricing

By evaluating our method using these diverse metrics, we can comprehensively assess the effectiveness of AdvDenoise in generating adversarial patches that achieve high attack success rates while maintaining visual realism and computational efficiency, making it suitable for practical real-world applications. Note that, the time here is only for the model training time (some exist early stopping). And the cost is also directly calculated the cost of the machine according to this time, and the artificial cost is not considered.

Method	ASR	SSIM	1-S *	100-S *	Cost
AdvPatch	56.78	0.12	**3.5h**	348h	$ 3.92k
FCA	79.66	0.59	4.2h	400h	$ 4.51k
DTA	80.29	0.54	4.5h	370h	$ 4.17k
AdvShadow	**85.28**	**0.78**	4.1h	405h	$ 4.57k
AdvDenoise	82.49	0.66	3.7h	**210h**	**$ 2.37k**

* S repersents number of shots.

Table 1. Main comparison with state-of-the-art methods.

4.3. Comparison with State-of-the-Art Methods

The results presented in Table. 1 and Figure. 2 demonstrate the significant improvements achieved by our AdvDenoise framework. With an Attack Success Rate (ASR) peaking at 82.49%, comparable to advanced state-of-the-art methods, AdvDenoise also exhibits exceptional performance in terms of visual realism, as evidenced by its Structural Similarity Index Measure (SSIM) scores being very close to the state-of-the-art.

However, AdvDenoise's true strength lies in its remarkable computational efficiency. When generating adversarial patches, either in a single-shot or multi-shot (100-Shots) setting, AdvDenoise requires almost the lowest computational cost among all evaluated methods. In the specified run-time environment, the total time required for generating 100 adversarial patches decreased from 405 hours to 210 hours, a remarkable 48.15% reduction in time complexity compared to the state-of-the-art. Consequently, the corresponding computational costs have also been greatly reduced. While, it should be noted that in the single-shot setting, AdvPatch is actually more efficient (0.2h faster). One explanation is that AdvPatch without requiring iterative optimization or multiple passes through the target model.

This outstanding computational efficiency, coupled with the high attack success rates and visual realism, suggests that AdvDenoise is particularly well-suited for realistic adversarial attack scenarios. By leveraging the power of denoising diffusion models and the proposed forward and reverse fission processes, AdvDenoise can effectively generate a diverse set of high-quality adversarial patches tailored to the target object and environment.

The ability to rapidly explore the local feature space around the initial patches and generate candidate solutions not only improves the overall attack success rate but also enhances the robustness of the generated patches against various transformations and environmental conditions. This robustness is crucial in practical real-world settings, where the adversarial patch may be subjected to various distortions and changes in viewpoint or lighting conditions.

The significant reduction in computational cost achieved by AdvDenoise makes it more accessible and practical for researchers and practitioners working on adversarial machine learning, enabling them to conduct extensive experiments and evaluations without the need for excessive computational resources.

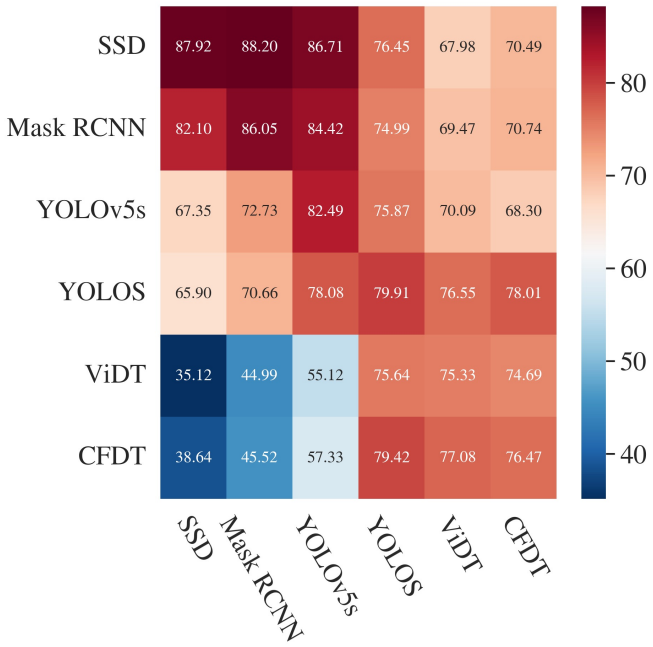

Figure 3. Heatmap of AdvDenoise adversarial transferability. We report top-1 attack success rate (%) of each method.

4.4. Robustness and Transferability on Targets

Our extensive experiments, illustrated in Figure. 3, highlight AdvDenoise's superior attack robustness (the attack success rate) and transferability across diverse target models. The main diagonal entries show that the Attack Success Rate (ASR) of the white-box attack on pre-trained models against either Convolutional Neural Networks (CNNs) or Vision Transformers (ViTs) is consistently around 80%, demonstrating AdvDenoise's effectiveness in generating adversarial patches tailored to the specific architecture of the target model.

Interestingly, we observe that the antidiagonal entries exhibit higher values than the secondary diagonal entries.

This indicates that the adversarial patches generated based on CNN models still maintain a certain level of generalization when attacking ViT models, although the overall performance is not as strong as when attacking the source CNN models. Notably, for the CFDT model, we observe an average decrease in ASR of 31.85% when attacked by patches generated from CNN models, suggesting that the transferability of adversarial patches between fundamentally different architectures (CNNs and ViTs) is limited.

We also find that off-diagonal entries sometimes exhibit higher values than the corresponding diagonal entries, indicating that certain adversarial patches transfer more effectively to different models than to the source model they were generated for. This intriguing phenomenon may result from the patches capturing generalized vulnerabilities shared across multiple models, model-specific sensitivities, or overfitting to idiosyncrasies of the source model during the patch generation process. This complex interplay between adversarial tactics and diverse model architectures warrants further investigation to deepen our understanding of these dynamics.

AdvDenoise's ability to generate adversarial patches that are not only effective against the source model but also exhibit varying degrees of transferability to other models is a valuable asset in practical adversarial attack scenarios. This robustness and transferability can aid in assessing the vulnerability of diverse machine learning systems to adversarial threats, enabling the development of more robust and secure models.

4.5. Ablation Study

The study helps us disentangle the impact of various design choices and analyze their individual contributions to the overall performance of our approach.

Role of Guidance Scale. The guidance scale parameter in diffusion models controls the trade-off between fidelity and diversity during the generative process. We ablate this parameter to understand its effect on the visual quality and attack success rates of the generated adversarial patches. As table. 2, we evaluated the performance of AdvDenoise with the above experimental settings and following guidance scale values: $[1.0, 10.0]$. Our results suggest that a carefully chosen guidance scale value can strike a balance between patch realism and adversarial effectiveness. If the primary objective is to maximize adversarial effectiveness while maintaining reasonable visual realism, a guidance scale value between 4.0 and 6.0 could be a suitable choice. This range strikes a balance between a high ASR $(76.41\%, 82.49\%)$ and an acceptable SSIM $(0.66, 0.78)$. If the primary focus is on maximizing adversarial effectiveness without significant concerns about visual realism, higher guidance scale values (e.g., > 8.0) could be chosen, as they achieve the highest ASR but with lower SSIM.

Guidance Scale	ASR	SSIM
1.0	58.30	0.90
2.0	64.72	0.88
3.0	70.11	0.85
4.0	76.41	0.78
5.0	79.86	0.74
6.0	82.49	0.66
7.0	83.01	0.57
8.0	84.23	0.50
9.0	85.36	0.44
10.0	85.48	0.39

Table 2. Impact of Guidance Scale on Adversarial Patch Attack Success Rate (ASR) and Structural Similarity Index (SSIM).

Through these ablation studies, we aim to provide a comprehensive analysis of the key factors influencing the performance of AdvDenoise. The findings from these studies not only validate the effectiveness of our framework but also offer guidance for future improvements and adaptations to specific application requirements.

5. Conclusion

In this paper, we introduced AdvDenoise, a novel framework for generating universal and robust adversarial patches using denoising diffusion models. Our approach leverages the power of diffusion models and the proposed forward and reverse fission processes to efficiently explore the local feature space around an initial adversarial patch, generating a diverse set of candidate patches tailored to the target object and environment.

Limitation and Future Work. Although AdvDenoise shows promising results with visually realistic patches and significantly reduced time complexity, our method necessitates further rigorous mathematical analysis and theoretical grounding. In the future, a crucial research direction is investigating whether AdvDenoise can generate adversarial patches that exploit vulnerabilities shared across diverse model architectures and paradigms.

Acknowlegement

We express our gratitude to Dr. Xingbin Wang for suggestions, and anonymous reviewers for their constructive comments. This research was supported by the National Key R&D. Program of China (Grant No. 2023YFC3305404); in part by the Institute for Industrial Innovation and Finance (IIIF), Tsinghua University, the Hong Kong General Research Fund (Grant No. 17503722), NSFC HY Working Fund (Grant No. 03070100001), Tsinghua SIGS Basic Support Fund (Grant No. 07010100003), and Tsinghua SIGS Research Support Fund (Grant No. 01030100049).

References

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018. 1

[2] T B Brown, D Mané, A Roy, M Abadi, et al. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 1, 2, 4, 6

[3] J Chen, H Chen, K Chen, et al. Diffusion models for imperceptible and transferable adversarial attack. *arXiv preprint arXiv:2305.08192*, 2023. 2, 3

[4] Zhaoyu Chen, Bo Li, Shuang Wu, Kaixun Jiang, Shouhong Ding, and Wenqiang Zhang. Content-based unrestricted adversarial attack. *Advances in Neural Information Processing Systems*, 36, 2024.

[5] X Dai, K Liang, and B Xiao. Advdiff: Generating unrestricted adversarial examples using diffusion models. *arXiv preprint arXiv:2307.12499*, 2023. 2, 3, 4

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 1, 2, 3

[7] Bao Gia Doan, Minhui Xue, Shiqing Ma, Ehsan Abbasnejad, and Damith C Ranasinghe. Tnt attacks! universal naturalistic adversarial patches against deep neural network systems. *IEEE Transactions on Information Forensics and Security*, 17:3816–3830, 2022. 1

[8] A Dosovitskiy, G Ros, F Codevilla, et al. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 4, 5

[9] D-P Fan, M-M Cheng, Y Liu, et al. Structure-measure: A new way to evaluate foreground maps. In *Proceedings of the IEEE international conference on computer vision*, pages 4548–4557, 2017. 6

[10] W-C Fan, Y-C Chen, D Chen, et al. Frido: Feature pyramid diffusion for complex scene image synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 579–587, 2023. 2, 3

[11] Y Fang, B Liao, X Wang, et al. You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems*, 34:26183–26197, 2021. 6

[12] Muhammad Zaid Hameed and Andras Gyorgy. Perceptually constrained adversarial attacks. *arXiv preprint arXiv:2102.07140*, 2021. 6

[13] K He, G Gkioxari, P Dollár, et al. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 6

[14] J Ho, A Jain, and P Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2, 3

[15] Yu-Chih-Tuan Hu, Bo-Han Kung, Daniel Stanley Tan, Jun-Cheng Chen, Kai-Lung Hua, and Wen-Huang Cheng. Naturalistic physical adversarial patch for object detectors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7848–7857, 2021. 1, 2, 3

[16] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2989–2998, 2023. 4

[17] G Jocher, A Chaurasia, A Stoken, et al. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation, 2022. 6

[18] Mintong Kang, Dawn Song, and Bo Li. Diffattack: Evasion attacks against diffusion-based adversarial purification. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[19] D Kingma and J Ba. Adam: A method for stochastic optimization. In *Proc. of ICLR*, 2015. 6

[20] 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. 1, 3

[21] Minjong Lee and Dongwoo Kim. Robust evaluation of diffusion-based adversarial purification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 134–144, 2023. 2

[22] J Li, S Zhang, X Wang, et al. Mimic octopus attack: Dynamic camouflage adversarial examples using mimetic feature for 3d humans. In *International Conference on Information Security and Cryptology*, pages 429–444. Springer, 2022. 1, 2, 4, 6

[23] Z Li, Y Nie, K Han, et al. A transformer-based object detector with coarse-fine crossing representations. *Advances in Neural Information Processing Systems*, 35:38733–38746, 2022. 6

[24] Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples. *arXiv preprint arXiv:2302.04578*, 2023. 3

[25] Shuo-Yen Lin, Ernie Chu, Che-Hsien Lin, Jun-Cheng Chen, and Jia-Ching Wang. Diffusion to confusion: Naturalistic adversarial patch generation based on diffusion model for object detector. *arXiv preprint arXiv:2307.08076*, 2023. 2, 3

[26] 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*, pages 1028–1035, 2019. 1

[27] Jiang Liu, Chun Pong Lau, and Rama Chellappa. Diffprotect: Generate adversarial examples with diffusion models for facial privacy protection. *arXiv preprint arXiv:2305.13625*, 2023. 2

[28] W Liu, D Anguelov, D Erhan, et al. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016. 6

[29] Yisroel M. Ipatch: a remote adversarial patch. *Cybersecurity*, 6(1):18, 2023. 1, 2

[30] Kento Oonishi, Tsunato Nakai, and Daisuke Suzuki. Multiple remote adversarial patches: Generating patches based on diffusion models for object detection using cnns. In *NeurIPS ML Safety Workshop*, 2022. 2, 3

[31] Yidong Ouyang, Liyan Xie, and Guang Cheng. Improving adversarial robustness through the contrastive-guided diffusion process. In *International Conference on Machine Learning*, pages 26699–26723. PMLR, 2023. 3

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2, 3

[33] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023. 3

[34] 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. 1

[35] H Song, D Sun, S Chun, et al. Vidt: An efficient and effective fully transformer-based object detector. *arXiv preprint arXiv:2110.03921*, 2021. 6

[36] Blake E Strom, Andy Applebaum, Doug P Miller, Kathryn C Nickels, Adam G Pennington, and Cody B Thomas. Mitre att&ck: Design and philosophy. In *Technical report*. The MITRE Corporation, 2018. 1

[37] N Suryanto, Y Kim, H Kang, et al. Dta: Physical camouflage attacks using differentiable transformation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15305–15314, 2022. 1, 4, 6

[38] C Szegedy, W Zaremba, I Sutskever, et al. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1, 2

[39] S Thys, Wiebe V R, and T Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 1, 2, 3, 6

[40] D Wang, T Jiang, J Sun, et al. Fca: Learning a 3d full-coverage vehicle camouflage for multi-view physical adversarial attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2414–2422, 2022. 1, 2, 4, 6

[41] Z Wang, A C Bovik, H R Sheikh, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[42] Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. In *International Conference on Machine Learning*, pages 36246–36263. PMLR, 2023. 3

[43] Shutong Wu, Jiongxiao Wang, Wei Ping, Weili Nie, and Chaowei Xiao. Defending against adversarial audio via diffusion model. *arXiv preprint arXiv:2303.01507*, 2023. 3

[44] H Xue, A Araujo, B Hu, et al. Diffusion-based adversarial sample generation for improved stealthiness and controllability. *arXiv preprint arXiv:2305.16494*, 2023. 2, 3

[45] Haotian Xue, Alexandre Araujo, Bin Hu, and Yongxin Chen. Diffusion-based adversarial sample generation for improved stealthiness and controllability. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 3

[46] C Yan, Z Xu, Z Yin, et al. Rolling colors: Adversarial laser exploits against traffic light recognition. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1957–1974, 2022. 1, 2

[47] B Zheng. Latent magic: An investigation into adversarial examples crafted in the semantic latent space. *arXiv preprint arXiv:2305.12906*, 2023. 2, 3

[48] Y Zhong, X Liu, D Zhai, et al. Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15345–15354, 2022. 1, 2, 6

[49] Haomin Zhuang, Yihua Zhang, and Sijia Liu. A pilot study of query-free adversarial attack against stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2384–2391, 2023. 3

[50] Z Zou, K Chen, Z Shi, et al. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 2023. 6