DiffPAD: Denoising Diffusion-based Adversarial Patch Decontamination

Jia Fu[1,2] Xiao Zhang[3] Sepideh Pashami[1,4] Fatemeh Rahimian[1] Anders Holst[1,2]
[1]RISE Research Institutes of Sweden [2]KTH Royal Institute of Technology
[3]CISPA Helmholtz Center for Information Security [4]Halmstad University
{jia.fu, sepideh.pashami, fatemeh.rahimian, anders.holst}@ri.se xiao.zhang@cispa.de

Abstract

*In the ever-evolving adversarial machine learning landscape, developing effective defenses against patch attacks has become a critical challenge, necessitating reliable solutions to safeguard real-world AI systems. Although diffusion models have shown remarkable capacity in image synthesis and have been recently utilized to counter ℓ_p-norm bounded attacks, their potential in mitigating localized patch attacks remains largely underexplored. In this work, we propose **DiffPAD**, a novel framework that harnesses the power of diffusion models for adversarial patch decontamination. DiffPAD first performs super-resolution restoration on downsampled input images, then adopts binarization, dynamic thresholding scheme and sliding window for effective localization of adversarial patches. Such a design is inspired by the theoretically derived correlation between patch size and diffusion restoration error that is generalized across diverse patch attack scenarios. Finally, DiffPAD applies inpainting techniques to the original input images with the estimated patch region being masked. By integrating closed-form solutions for super-resolution restoration and image inpainting into the conditional reverse sampling process of a pre-trained diffusion model, DiffPAD obviates the need for text guidance or fine-tuning. Through comprehensive experiments, we demonstrate that DiffPAD not only achieves state-of-the-art adversarial robustness against patch attacks but also excels in recovering naturalistic images without patch remnants.*

1. Introduction

Despite achieving remarkable success in a wide range of machine learning applications, deep neural networks (DNNs) are extremely susceptible to adversarial examples [34], normal inputs crafted with small perturbations that are devised to induce model errors. The discovery of adversarial examples raises serious concerns about the robustness of DNNs, particularly for security-sensitive domains. Existing works primarily focus on ℓ_p-norm bounded perturbations [1, 5, 12], a specific type of global attacks, where the adversary is allowed to manipulate all the pixels within the entire image. In contrast, adversarial patch attacks [3, 18, 41] restrict the total number of pixels that can be modified, which typically occupy a small localized region within the image. Since adversarial patches can be easily attached to physical-world objects [39], they pose more realistic threats to security-critical DNN systems, ranging from surveillance cameras [42] to autonomous vehicles [11]. Therefore, it is crucial to develop effective defenses that are robust to adversarial patch attacks.

Numerous defenses have been proposed to enhance the robustness of DNNs, such as adversarial purification [28], adversarial training [1] and certified defenses [40]. In particular, adversarial purification [13, 31, 32] leverages the power of generative models to remove adversarial noise. Compared with adversarial training and certified defenses, adversarial purification has the potential to protect the target model without the need for adaptive retraining. Witnessing the exceptional capability of diffusion models for image synthesis tasks [9, 14], recent works utilize diffusion models for adversarial defenses [4, 24, 38], which achieve state-of-the-art performance in building robust models. However, all these methods are designed to defend against global attacks. When such defenses encounter localized patch attacks, their performance will drop to different extents, failing to fulfill the security requirements of real-world applications. It remains unclear how adversarial purification and diffusion models can help mitigate adversarial patch attacks. In order to defend against localized patch attacks, more specialized strategies, such as adversarial patch detection and segmentation [21, 35], have been developed to isolate and neutralize adversarial patches before processing images through the model. These methods leverage the advancements in digital image processing to detect anomalies that are indicative of patch tampering. Nevertheless, these methods often struggle to reconstruct the original images with high fidelity and are not successful in defending against adaptive attacks that can avoid gradient obfuscations [2].

To address the limitations of previous defense strategies

and exploit the full potential of diffusion models in defending against patch attacks, we propose **DiffPAD**, a **Diff**usion-based framework for adversarial **PA**tch **D**econtamination. Figure 1 depicts the workflow of DiffPAD. By decomposing the defense task into patch localization and inpainting, we address these two sub-problems via a diffusion model's conditional reverse sampling process. Such a conditional process incorporates the visual information of the clean region to retain label semantics integrally. We also show evidence that justifies the usefulness of conditional diffusion models in mitigating the distribution deviation caused by adversarial patch variations, and achieve effective patch localization through a single diffusion sampling process, where the discrepancy in adversarial region is accentuated by resolution degradation and restoration techniques. Comprehensive experiments demonstrate the substantial advancement of DiffPAD in adversarial patch defense over existing methods, offering a robust and scalable solution that aligns with real-world application requirements. Our main contributions are further summarized as follows:

- We integrate the closed-form solution of image super-resolution into the reverse sampling process of a pre-trained diffusion model for patch localization, eliminating the need for fine-tuning and multiple reverse generations. Patch decontamination is then accomplished using the same diffusion process, but by switching the closed-form solution to inpainting.
- We prove a linear correlation between the patch size and an upper bound of diffusion restoration error. Such a relationship is empirically verified across different classification models and patch attacks with varying patch sizes and random positions, which facilitates the efficiency and accuracy of patch detection.
- Through comprehensive experiments on image classification and facial recognition tasks, we demonstrate that DiffPAD achieves state-of-the-art adversarial robustness against both adaptive and non-adaptive patch attacks, and is capable of completely removing patch remnants and generating naturalistic images.

2. Related work

This section introduces the most related works to ours. Other discussions are provided in supplementary materials.

Defenses against adversarial patches. To enhance the model robustness against adversarial attacks, various defense strategies have been proposed. Initial attempts focused on simple preprocessing-based defenses, such as JPG compression [10], thermometer encoding [15], defensive distillation [25]. However, these methods have been shown ineffective against adaptive attacks [2]. Adversarial training [1], which optimizes the neural network parameters by incorporating adversarially generated inputs in training, is by far the most popular. Nevertheless, adversarial training suffers from high computational costs, due to the iterative steps required for generating strong adversarial examples. Certified defenses [40] have also been developed, but they cannot achieve a similar level of robustness to adversarial training. When adversarial training and certified defenses are applied to defend against adversarial patches [27, 29], the learned model is usually only effective to the specific attacks employed in training but shows inferior generalization performance to unseen patch attacks. To achieve comparable robustness against adversarial patches, more specialized patch detection schemes have been developed to localize and purify adversarial patches. For instance, Liu *et al.* [21] trained a patch segmenter to generate pixel-level masks for adversarial patches, while Tarchoun *et al.* [35] identified patch localization based on the property that the entropy of adversarial patches is higher compared with other regions. Our work falls into the line of patch detection-based defenses, but aims to leverage the power of diffusion models to achieve better defense performance.

Adversarial purification. Adversarial purification [24, 28, 31, 32] refers to a special family of preprocessing-based defenses that makes use of generative models. For example, DiffPure [24] gradually injects Gaussian noise in the forward diffusion steps followed by denoising during the reverse generation phase, where the adversarial noise is purified along the process. Such state-of-the-art generative models offer a promising avenue for mitigating adversarial examples [6, 37]. Nevertheless, adversarial purification frameworks are typically designed for purify ℓ_p-norm bounded perturbations, which may fall short against the discrete and localized nature of adversarial patches. So far, we have only identified a single existing method, DIFFender [16], that utilizes diffusion models to defend against patch attacks. DIFFender adopts a text-guided diffusion model to localize the adversarial patch and then reconstruct the original image. Unfortunately, DIFFender not only requires expensive multiple reverse diffusion processes for effective patch localization, but also relies on heuristic manually-designed prompts or complex prompt tuning steps, hindering automation in practical use cases.

3. Preliminaries on diffusion models

A diffusion model consists of forward and reverse diffusion processes. The forward process progressively degrades the underlying distribution p_0 towards a noise distribution by adding Gaussian noise, which can be characterized by a stochastic differential equation (SDE) [43]:

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, t)\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}, \tag{1}$$

where $\boldsymbol{x}_t \in \mathbb{R}^d$ follows p_t denoting the distribution at time step t, $\boldsymbol{w}$ denotes the standard Wiener process (a.k.a. Brow-

nian motion), and $\boldsymbol{f}$: $\mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ and g: $\mathbb{R} \to \mathbb{R}$ represent the drift and diffusion coefficients, respectively. Given the distribution p_t, the reverse diffusion process with respect to Equation 1 can be formulated as:

$$\mathrm{d}\boldsymbol{x} = \left[\boldsymbol{f}(\boldsymbol{x}, t) - g(t)^2 \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] \mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}. \quad (2)$$

For generation tasks where the condition $\boldsymbol{y}$ is specified, the objective is to sample data from $p(\boldsymbol{x}|\boldsymbol{y})$. By applying Bayes' theorem, the conditional process with respect to Equation 2 can be written as:

$$\mathrm{d}\boldsymbol{x} = \left[\boldsymbol{f}(\boldsymbol{x}, t) - g(t)^2 \nabla_{\boldsymbol{x}} \big(\log p_t(\boldsymbol{x}) + \log p_t(\boldsymbol{y}|\boldsymbol{x})\big)\right] \mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}. \quad (3)$$

Denoising diffusion probabilistic models (DDPMs). DDPM [14] is a milestone in diffusion models, which offers unparalleled stability and quality for generative tasks. Specifically, DDPM models the generative process using a Markov chain $\boldsymbol{x}_T \to \boldsymbol{x}_{T-1} \to \ldots \to \boldsymbol{x}_0$, where the joint distribution is defined as:

$$p_\theta\left(\boldsymbol{x}_{0:T}\right) = p\left(\boldsymbol{x}_T\right) \prod_{t=1}^{T} p_\theta\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t\right). \quad (4)$$

DDPM sets $\boldsymbol{f}(\boldsymbol{x}, t) = -\frac{1}{2}\beta_t \boldsymbol{x}$ and $g(t) = \sqrt{\beta_t}$ and derives the discrete-time diffusion processes. According to the statistical properties of Gaussian distribution, DDPM samples $\boldsymbol{x}_t$ from $\boldsymbol{x}_0$ using the following closed-form solution:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, \quad (5)$$

where $\bar{\alpha}_t = \Pi_{s=1}^{t}(1 - \beta_s)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. For the reverse sampling, DDPM estimates $\boldsymbol{x}_0$ based on the following approximation:

$$\boldsymbol{x}_0 \approx \hat{\boldsymbol{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{x}_t - \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}_\theta\left(\boldsymbol{x}_t\right)\right), \quad (6)$$

where $\boldsymbol{\epsilon}_\theta$ denotes the neural network designed to predict the total noise between $\boldsymbol{x}_t$ and $\boldsymbol{x}_0$ based on Equation 5.

Denoising diffusion restoration models. Given the strong sample quality of DDPMs in image synthesis, DDPMs have been adapted for conditional use in image restoration tasks [7, 20, 44]. Kawar *et al.* [20] first introduced the term "DDRM", where the condition $\boldsymbol{y}$ is a degraded image of $\boldsymbol{x}$ and the distribution of DDRM is defined as:

$$p_\theta\left(\boldsymbol{x}_{0:T}|\boldsymbol{y}\right) = p\left(\boldsymbol{x}_T|\boldsymbol{y}\right) \prod_{t=1}^{T} p_\theta\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{y}\right). \quad (7)$$

In the following discussions, we refer to the family of methods that conditionally utilize DDPMs for image restoration, including the first work [20], as DDRMs. During the reverse generation process, DDRMs replace $\hat{\boldsymbol{x}}_0$ by its $\boldsymbol{y}$-conditioned counterpart $\tilde{\boldsymbol{x}}_0$. Variations among different DDRM models primarily arise in the computation of $\tilde{\boldsymbol{x}}_0$ from $\hat{\boldsymbol{x}}_0$ and $\boldsymbol{y}$. For instance, the vanilla DDRM proposed a singular value decomposition (SVD) based approach, assuming a linear degradation function to compute $\tilde{\boldsymbol{x}}_0$ by pseudo-inverse, whereas DiffPIR [44] calculated $\tilde{\boldsymbol{x}}_0$ by solving a data proximal sub-problem.

4. Proposed method: DiffPAD

In this section, we first motivate and explain our design of DiffPAD that utilizes conditional diffusion models for accurate patch localization followed by patch restoration. Mathematically, we work with the following definition of adversarial patches. Let $\boldsymbol{x}^c$ denote the clean image and $\boldsymbol{\delta}$ be the adversarial perturbation. Then, the adversarial patch contaminated image can be defined as:

$$\boldsymbol{x}^a = (\mathbf{1} - \mathbf{A}) \odot \boldsymbol{x}^c + \mathbf{A} \odot \boldsymbol{\delta}, \quad (8)$$

where $\mathbf{A} \in \{0, 1\}^d$ is the mask of the region outside the adversarial patch, and $\odot$ denotes the Hadamard product. Unlike global attacks, adversarial patches modify only a small localized region of the clean image, concealing its original visual information.

4.1. DDRMs for patch defenses

DiffPAD follows a stepwise pipeline of patch restoration after its localization, which utilizes DDRMs for defending against patch attacks (Figure 1). Equation 8 reveals that the label semantics of $\boldsymbol{x}^a$ comes from $(\mathbf{1} - \mathbf{A}) \odot \boldsymbol{x}^c$, i.e., the clean region. The clean region itself serves as the optimal condition for guiding the diffusion process to keep the image semantics of the original clean image to the greatest extent. Although $\mathbf{A}$ is unknown for a given adversarial example, all information from $(\mathbf{1} - \mathbf{A}) \odot \boldsymbol{x}^c$ is included in $\boldsymbol{x}^a$, suggesting an approach to construct the condition based on $\boldsymbol{x}^a$. If we denote $\mathcal{H}$ as an image degradation function and set $\boldsymbol{y} = \mathcal{H}(\boldsymbol{x}^a)$, the corresponding diffusion process naturally translates to DDRMs. A moderate image degradation function, such as image compression, typically preserves enough image semantics to allow high-quality restoration by DDRMs. Employing DDRMs as the foundation model of DiffPAD frees us from selecting a specific halt time step or introducing extra text prompts to keep label semantics during reverse diffusion sampling. Based on variational inference, the ELBO objective of DDRMs can be rewritten in the form of DDPMs objective, as shown in Theorem 3.2 in [20], which supports the feasibility of approximating the optimal solutions of our DDRM-based framework by pre-trained DDPM models without any fine-tuning. The Gaussian noise of DDPMs is in nature the discretization of the variance preserving SDE [43], so does DDRMs. Therefore, the following property held in the forward process of

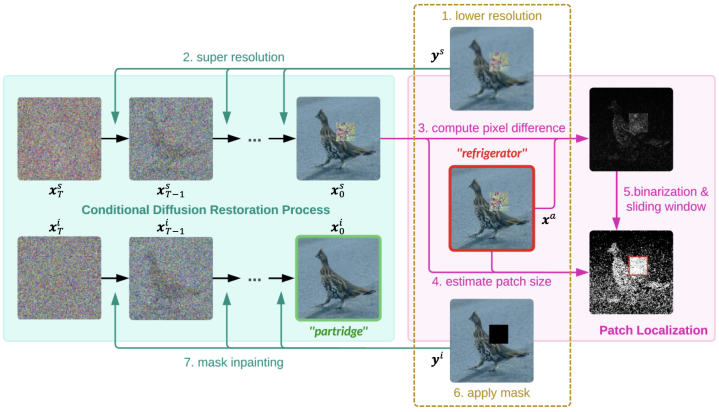

Figure 1. The overall pipeline of DiffPAD, which follows steps numbered from 1 to 7 in order. Text and blocks in turquoise, pink and yellow correspond to the conditional diffusion restoration module, patch localization module and image degradation operations, respectively. The input of DiffPAD is the adversarial patch contaminated image $\boldsymbol{x}^a$ (with red frame), and the output is the decontaminated image $\boldsymbol{x}_0^i$ (with green frame).

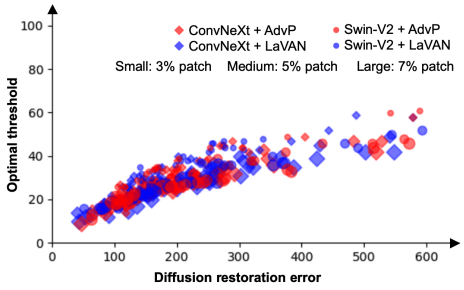

Figure 2. Illustration of the linear relationship between diffusion restoration errors and optimal thresholds for patch localization under various attacks. In particular, we vary the size of the adversarial patches generated by different attacks on various model architectures.

DDPMs also holds for DDRMs:

$$\frac{\partial D_{\mathrm{KL}}\left(p_t^c \| p_t^a\right)}{\partial t} \geq 0, \tag{9}$$

where D_{KL} denotes the Kullback-Leibler divergence. Equation 9 implies that the injected Gaussian noise will disrupt the patterns of adversarial patches, gradually aligning the adversarial distribution p^a with the clean distribution p^c through the forward DDRMs process.

Resolution degradation and restoration. We aim to localize the patch according to the property that the adversarial region exhibits more drastic changes compared to the clean region when $\boldsymbol{x}^a$ is compared with its diffusion-generated counterpart. However, as demonstrated in [16], solely relying on the stochasticity of the diffusion process to disrupt the patch pattern is inefficient. To address this challenge, we propose a *resolution degradation-restoration mechanism* to amplify the diffusion restoration error in the patch region. Such a design is motivated by the observation that image compression can enhance the model robustness against patch attacks, suggesting the high sensitivity of adversarial patches to resolution changes. In DiffPAD, we first employ bicubic down-sampling with a scaling factor s on $\boldsymbol{x}^a$ to obtain $\boldsymbol{y}^s$, serving as the initial destruction to the adversarial patch distribution, then intensify this destruction through the randomness along with DDRMs super-resolution restoration, conditioned on $\boldsymbol{y}^s$ (Steps 1-2 in Figure 1). This preparation allows DiffPAD to precisely localize the adversarial patch through a single diffusion generation, whereas DIFFender necessitates the generation of at least three samples per image to ensure robust patch localization. Different from the vanilla DDRM, which assumes that $\mathcal{H}$ is linear and uses an SVD solution to compute $\tilde{\boldsymbol{x}}_0$. DiffPAD leverages a fast closed-form solution for efficient diffusion restoration. Denoting $\eta_t = \bar{\alpha}_t\sigma^2/(1-\bar{\alpha}_t)$ where σ represents the noise level associated with $\boldsymbol{y}$, we adopt a plug-and-play image resolution restoration function from [45] with a bicubic kernel $\mathbf{k}$:

$$\tilde{\boldsymbol{x}}_0 = \mathcal{F}^{-1}\left(\frac{1}{\eta_t}\left(\mathbf{d} - \overline{\mathcal{F}}(\mathbf{k}) \odot_s \frac{(\mathcal{F}(\mathbf{k})\mathbf{d})\downarrow_s}{(\overline{\mathcal{F}}(\mathbf{k})\mathcal{F}(\mathbf{k}))\downarrow_s + \eta_t}\right)\right), \tag{10}$$

where $\mathbf{d} = \overline{\mathcal{F}}(\mathbf{k})\mathcal{F}\left(\boldsymbol{y}^s\uparrow_s\right) + \eta_t\mathcal{F}\left(\hat{\boldsymbol{x}}_0\right)$. $\mathcal{F}$, $\mathcal{F}^{-1}$, and $\overline{\mathcal{F}}$ denote Fast Fourier Transform (FFT), its inverse, and conjugate, respectively. As for operators, $\uparrow_s$ is the standard s-fold up-sampler, $\downarrow_s$ is the down-sampler that averages $s \times s$ distinct blocks, and $\odot_s$ is element-wise multiplication for distinct block processing.

Inpainting. By substituting Equation 10 with the closed-form solution for inpainting restoration, we can take advantage of the same pre-trained DDPMs used in super-resolution restoration. Once the patch is localized and masked, we adapt a plug-and-play color image demosaicing function from [45] for the image inpainting task:

$$\tilde{\boldsymbol{x}}_0 = \frac{\mathbf{M} \odot \boldsymbol{y}^i + \eta_t\hat{\boldsymbol{x}}_0}{\mathbf{M} + \eta_t}, \tag{11}$$

where $\mathbf{M} \in \{0,1\}^d$ is a customized mask on $\boldsymbol{x}^a$ for acquisition of $\boldsymbol{y}^i$. Note the division operations in Equations 10 and 11 are element-wise.

4.2. Adversarial patch localization

This section explains the algorithm for estimating patch size and localizing its position. We adopt the most common setting for adversarial patches: crafting a single square-shaped patch of random size and position on a given clean

image. Typically, the size of adversarial patches should not be too large, as it will block the label semantics and render the image unrecognizable to humans. Inspired by Theorem 3.2 of [24], which proves an upper bound on the ℓ_2 distance between a diffusion-purified ℓ_p-norm bounded adversarial example and the corresponding clean image, we establish a similar result in the following theorem for patch attacks.

Theorem 1 *Assume* $\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t)\| \leq C_\epsilon\sqrt{1-\bar{\alpha}_t}$ *and let* $\gamma := \int_0^T \beta_t \mathrm{d}t$. *With probability at least* $1-\xi$, *the* ℓ_2 *distance between the diffusion-purified image* $\hat{\boldsymbol{x}^a}$ *with adversarial patch and the corresponding clean image* $\boldsymbol{x}^c$ *satisfies:*

$$\|\hat{\boldsymbol{x}^a} - \boldsymbol{x}^c\| \leq \varepsilon\,|\mathbf{A}| + \gamma C_\epsilon + \sqrt{e^\gamma - 1}\cdot C_\xi, \tag{12}$$

where ε *is the* ℓ_2*-norm bound of the patch,* $C_\xi := \sqrt{2d + 4\sqrt{d\log\frac{1}{\xi}} + 4\log\frac{1}{\xi}}$, *and* d *is the input dimension.*

The proof of Theorem 1 is provided in the supplementary materials. Note that $\|\boldsymbol{x}^c - \boldsymbol{x}^a\| \leq \varepsilon\,|\mathbf{A}|$, then with the help of the triangle inequality, the ℓ_2 distance between a patch-contaminated image before and after diffusion reconstruction can be upper bounded as:

$$\|\hat{\boldsymbol{x}^a} - \boldsymbol{x}^a\| \leq 2\varepsilon\,|\mathbf{A}| + \gamma C_\epsilon + \sqrt{e^\gamma - 1}\cdot C_\xi, \tag{13}$$

where the bound is linearly correlated to the patch area $|\mathbf{A}|$. In practice, directly estimating the patch size based on $\|\hat{\boldsymbol{x}^a} - \boldsymbol{x}^a\|$ yields unsatisfactory results. The upper bound of ℓ_2 distance primarily emphasizes the most significant differences arising from the restoration. In other words, the subtle variations caused by the intrinsic randomness of the diffusion model should be neglected.

Denote $\ominus$ as the pixel-wise difference and $\boldsymbol{x}^\Delta := \hat{\boldsymbol{x}^a} \ominus \boldsymbol{x}^a$. As previously justified, the adversarial regions with nearly full-scale exhibit higher discrepancies in $\boldsymbol{x}^\Delta$, a result of our patch error amplification implemented during the diffusion restoration on resolution (Step 3 in Figure 1). To isolate the pixels that contribute most to $\|\hat{\boldsymbol{x}^a} - \boldsymbol{x}^a\|$ and count their quantity to represent the patch area, we apply the binarization on $\boldsymbol{x}^\Delta$ with dynamic threshold τ. Enlightened by Equation 13, we posit that τ more reasonably reflects the upper bound of $\|\hat{\boldsymbol{x}^a} - \boldsymbol{x}^a\|$. A higher τ indicates that the pixels filtered out by binarization are likely to have a higher diffusion restoration error, suggesting a higher probability of originating from the patch region, equivalently, a larger patch area in a given image. Therefore, we propose to model τ as being linearly correlated with the mean squared error (MSE) between $\hat{\boldsymbol{x}^a}$ and $\boldsymbol{x}^a$:

$$\tau = \mu \cdot \frac{\|\hat{\boldsymbol{x}^a} - \boldsymbol{x}^a\|}{d} + \nu, \tag{14}$$

where μ and ν are hyperparameters. Consequently, the estimated patch area will be $\tilde{A} = \left|\mathrm{Binarize}(\boldsymbol{x}^\Delta, \tau)\right|$ (Step 4 in Figure 1). To illustrate the linear relation, we craft adversarial examples from the ImageNet dataset with varying patch areas ($3\%, 5\%, 7\%$ of full image size) and random positions using different attack mechanisms (i.e., AdvP and LaVAN) across diverse classifiers (i.e., ConvNeXt and Swin-V2). Each attack consists of 25 examples. We conduct a traversal search to identify the optimal threshold for binarization on $\boldsymbol{x}^\Delta$, which provides the most accurate estimation of the original patch area for each example. By depicting the relation between the optimal thresholds and the corresponding MSE values, Figure 2 confirms their linear correlation and validates the effectiveness of our estimation method.

After obtaining the estimated patch size, DiffPAD uses a sliding window of the same size to scan the values of $\mathrm{Binarize}(\boldsymbol{x}^\Delta, \tau')$, where τ' is a fixed threshold for suppression of the faint background restoration errors caused by the intrinsic stochasticity of DDRMs. We pinpoint the window position that contains the most "1" pixels as the localized adversarial patch position (Step 5 in Figure 1). Finally, DiffPAD masks the localized patch region and reconstructs the visual content in it by diffusion restoration conditioning on the surrounding unaltered region (Steps 6-7 in Figure 1).

5. Experiments

5.1. Experimental setup

This section introduces our main experimental setup. Other details are deferred to the supplementary materials.

Datasets and networks. We evaluate DiffPAD by an image classification task on ImageNet [8] and a facial recognition task on VGG Face [26]. We employ up-to-date pre-trained classifiers ConvNeXt [23] and Swin-V2 [22], which represent the most advanced architectures in convolution neural networks (CNNs) and vision transformers (ViTs), respectively. We use Inception-V3 [33] for adaptive attacks as in DIFFender [16] to compare their results with ours. We take advantage of the pre-trained diffusion model from [9].

Attacks. We evaluate DiffPAD with three different patch attacks. AdvP [3] is the standard localized perturbation with random position. LaVAN [18] enhances the gradient updates of AdvP and delivers stronger attacks. GDPA [41] optimizes the patch's position and pattern by an extra generative network. For AdvP and LaVAN, the number of attack iterations is set as 500. For GDPA, the default 50 epochs are employed. We also consider the white-box scenario, i.e., leveraging adaptive attack for comprehensive assessment. Since DiffPAD and selected baselines are in nature preprocessing mechanisms, we approximate obfuscated gradients via BPDA [2], assuming the output of the defense function equals the clean input. For BPDA-AdvP and BPDA-LaVAN, the number of attack iterations is set as 100.

Baselines. We choose defense baselines that can serve as

Table 1. Comparisons of clean and robust accuracies (%) on ImageNet with ConvNeXt across different patch defenses.

Defense \ Attack	Clean	AdvP	LaVAN	GDPA
w/o defense	83.9	4.7	3.9	77.1
JPG [10]	77.3	74.9	73.9	76.0
SAC [21]	80.6	79.8	80.0	79.4
Jedi [35]	82.2	80.3	80.8	80.1
DiffPure [24]	76.4	74.3	74.5	75.6
DiffPAD	**82.3**	**82.3**	**82.2**	**80.4**

Table 2. Comparisons of clean and robust accuracies (%) on ImageNet with Swin-V2 across different patch defenses.

Defense \ Attack	Clean	AdvP	LaVAN	GDPA
w/o defense	83.4	0.5	0	75.7
JPG [10]	77.0	74.9	74.3	75.8
SAC [21]	81.5	81.1	80.8	79.6
Jedi [35]	81.3	79.5	79.4	79.0
DiffPure [24]	76.9	75.7	75.3	76.3
DiffPAD	**81.7**	**82.1**	**81.4**	**80.1**

purification on input images, including smoothing-based defense JPG [10], segmentation-based defense SAC [21], entropy-based defense Jedi [35], and diffusion-based defense DiffPure [24] and DIFFender [16]. Among them, SAC, Jedi and DIFFender are specialized defenses against adversarial patches. Except for DIFFender which is not open-source, all other baselines are executed by their original implementation taking default parameter settings.

Evaluation metrics. The primary metrics for evaluating defenses are clean and robust accuracies under patch attacks. We evaluate the faithfulness of images post-defense, compared to clean images by Peak Signal-to-Noise Ratio (PSNR). In addition, we compute the mean Intersection over Union (mIoU) between the estimated patch region and the ground truth, which is an auxiliary metric to reflect the accuracy of our patch detection module. For all these metrics, a higher value indicates a better performance.

5.2. Main results

Table 1 and Table 2 showcase the superior performance of DiffPAD, which outperforms all baselines on both clean and attacked images using AdvP, LaVAN and GDPA. While the accuracy gap between DiffPAD and the second-best baseline is marginal for clean data and GDPA attack, the improvement is significant for AdvP and LaVAN attacks. Notably, on ConvNeXt under AdvP attack, DiffPAD exceeds the second-ranked baseline by 2.0%. Generally, the clean or robust accuracy of JPG and DiffPure cannot surpass 78%, while SAC, Jedi, and DiffPAD are all higher than 79%. Evidently, global defenses fall short compared to specialized patch defenses under patch attacks. Without defense, both ConvNeXt and Swin-V2 report highly accurate predictions on clean data. However, their performance will significantly drop under AdvP and LaVAN attacks. In particular, almost no images remain unscathed by AdvP or LaVAN attacks on Swin-V2, with accuracy plummeting to 0.5% and 0%, respectively. After applying DiffPAD, the accuracy will be recovered to around 82%, where the reduction compared to the clean accuracy is less than 2% for both ConvNeXt and Swin-V2. Conversely, the GDPA attack results in a minor accuracy reduction, namely 6.8% on ConvNext and 7.7% on Swin-V2, yet defending against this attack proves to be more challenging. Neither DiffPAD nor selected baselines can bring the accuracy level back to above 81%. Given GDPA's inadequate attack performance on ConvNeXt and Swin-V2, we will mainly focus on DiffPAD under AdvP and LaVAN attacks in the following experiments.

Table 3. Comparisons of clean and robust accuracies (%) under adaptive BPDA attacks with Inception-V3. Results of baselines are directly drawn from DIFFender [16].

Defense \ BPDA attack	Clean	AdvP	LaVAN
w/o defense	100.0	0.0	8.2
JPG [10]	48.8	0.4	15.2
SAC [21]	92.8	84.2	65.2
Jedi [35]	92.2	67.6	20.3
DiffPure [24]	65.2	10.5	15.2
DIFFender [16]	91.4	88.3	71.9
DiffPAD	**94.1**	**90.0**	**88.3**

An interesting observation is that Jedi performs better on ConvNeXt in all scenarios, ranking as the second-best defense, while SAC serves as the second-best on Swin-V2. This distinction is likely because the features extracted by ViTs are more globally entangled. In other words, when SAC blocks the visual content corresponding to the adversarial patches, ViTs still receive contextual information from other regions. In contrast, CNN-based architectures focus more on localized details, making them more susceptible to information loss. For CNNs, employing Jedi to disrupt the pattern of adversarial patches with their surrounding pixels is a better choice. This explains why ConvNeXt is more robust than Swin-V2 in the absence of defenses. Adversarial patches influence all semantic patches in the token operations of ViTs, while their impact on CNNs is less pronounced, because the small convolution kernels limit the effective propagation of localized information overall.

Adaptive attacks. Table 3 presents the results of the BPDA

Table 4. Effects of different modules in DifffPAD. PAD, INP and SVD stand for patch detection, inpainting restoration and singular value decomposition, respectively.

Defense \ Attack	ConvNeXt (%)		Swin-V2 (%)	
	AdvP	LaVAN	AdvP	LaVAN
DiffPAD w/o PAD	70.7	69.5	72.9	71.7
DiffPAD w/o INP	80.7	80.1	81.2	**82.1**
DiffPAD (SVD)	**82.5**	82.0	81.8	81.9
DiffPAD	82.3	**82.2**	**82.1**	81.4

Table 5. Patch localization precision in mIoU (%) of DiffPAD with varying patch sizes and random positions.

Patch \ Attack	ConvNeXt		Swin-V2	
	AdvP	LaVAN	AdvP	LaVAN
size 3%	82.27	83.34	80.42	80.57
size 5%	85.10	83.40	86.66	86.31
size 7%	83.35	83.44	86.52	86.69

adaptive attacks on Inception-V3. We observe that JPG and DiffPure significantly underperform, achieving less than 20% accuracy under both AdvP and LaVAN attacks. This suggests that the gradients of such global, nuanced rectification methods are easier to be approximated than those of localized, drastic modification methods. Jedi exhibits stronger resilience against the AdvP-BPDA attack but fails to generalize to the LaVAN-BPDA attack. On the contrary, DiffPAD maintains consistent performance across different networks and attacks, whether adaptive or not. While SAC and DIFFender show commendable adversarial robustness in adaptive settings, they are still inferior to DiffPAD. For instance, DiffPAD outperforms DIFFender by 16.4% under the LaVAN-BPDA attack, likely due to its more exact localization capabilities for adversarial patches.

Ablation study. Table 4 provides the results of an ablation study on each component of DiffPAD. Excluding patch detection means that we perform only resolution degradation-restoration on input images through a single diffusion generation. This yields even lower accuracy than the simplest JPG defense under the same attack conditions, as the resolution degeneration cannot be eliminated but rather mitigated. The blurring effect occurs globally, affecting both the structure of the adversarial patch as well as other visual details. As a result, Swin-V2 outperforms ConvNeXt by 2.2% under either the AdvP or LaVAN attack, which is consistent with previous findings that CNNs are more sensitive to fine-grained visual features.

Including the patch detection module while excluding the inpainting restoration elevates DiffPAD to perform comparably with SAC under the same attacks. The final inpainting restoration step further promotes DiffPAD to a new SOTA in adversarial patch defense. Inpainting is necessary when a patch obscures critical semantics of the input images. It supplies meaningful visual content to the patch region, helping classifiers understand label semantics and preventing them from interpreting the mask as pertinent visual data. The increased stochasticity from more diffusion steps is also beneficial for resisting adaptive attacks. The last two rows in Table 4 are outcomes of switching the closed-form solutions used in conditional sampling during the reverse diffusion process. The SVD solution [20] has been explained at the end of Section 3. The close accuracy levels under the same attacks imply that the influence of altering the conditional sampling strategy is small, suggesting the flexibility and stability of DiffPAD.

Table 6. The faithfulness (PSNR) of images after various patch defenses with reference to the original clean images. Unit: dB.

Defense \ Attack	ConvNeXt		Swin-V2	
	AdvP	LaVAN	AdvP	LaVAN
w/o defense	21.25	20.70	22.26	22.01
SAC [21]	19.63	19.57	19.42	19.42
Jedi [35]	22.77	22.57	22.95	22.94
DiffPAD	**26.38**	**26.63**	**27.43**	**27.53**

Varying patch attacks. Table 5 demonstrates the patch localization precision under diverse patch attack conditions. It can be observed from the table that all the mIoU scores break through 80%, ensuring the generalizability of the patch detection module across varying patch sizes and random positions. This finding also validates the result of Theorem 1 with strong empirical evidence.

5.3. Further analyses

Visualizations. To examine the capability of removing patch remnants, we visualize the images returned by different methods. Figure 3 displays the visual effects on adversarial patches after applying baseline defenses and DiffPAD. DiffPure generally acts like a blurring function on the entire image, which is noticeable in the background of the first example. However, the pattern of adversarial patches cannot be washed out, confusing classifiers when the patch features compound with label-related features. SAC, while seldom misidentifying non-patch regions as adversarial, often fails to completely screen out all adversarial pixels, as shown in the first example. An apparent issue with Jedi is that it crushes estimated patch regions with distortions, injecting disruptive cues that interfere with recognition, especially for the second example. Jedi also omits the adversarial patch in the third example. In contrast, examples of DiffPAD conceal any patch remnants, with both naturalistic

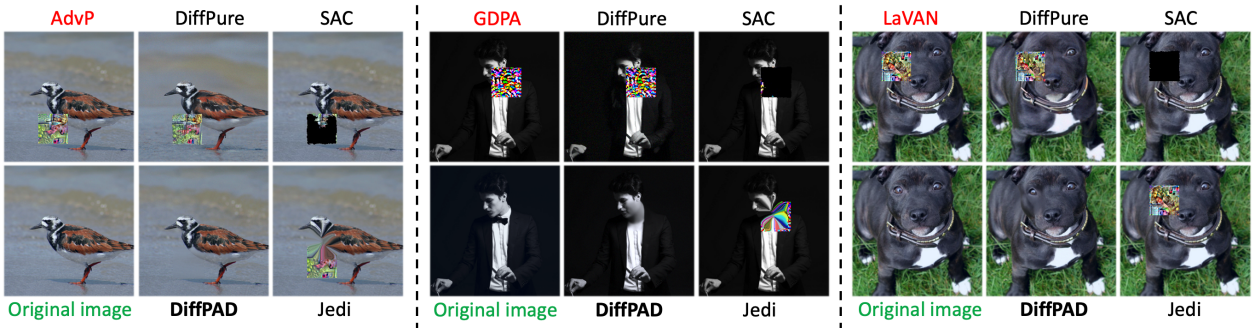

Figure 3. Illustration of three exampled visual effects on adversarial patches before and after applying different patch defenses. Note that it is difficult to find any traces of the adversarial patch from the images decontaminated by DiffPAD.

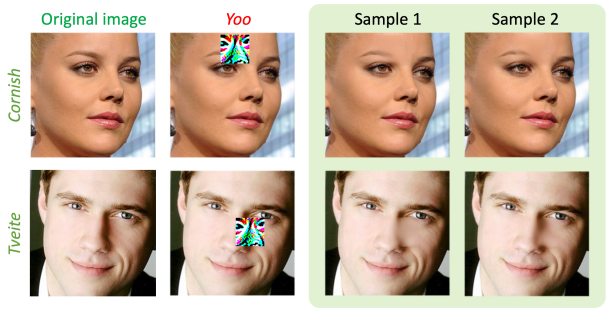

Figure 4. The performance of DiffPAD in facial recognition task on VGG Face. We run twice to attain two well-restored samples.

and meaningful visual content. Table 6 lists that all PSNR values for DiffPAD are above 26 dB under all attack conditions, confirming the remarkable fidelity of the decontaminated adversarial examples referring to their original states. Additional visualization results in the supplementary materials also show our method has a negligible effect on clean images, ensuring low false positive rates of patch detection.

Transferability to facial recognition. To assess the transferability of DiffPAD, we apply the same hyperparameters selected for ImageNet experiments to the VGG Face in a facial recognition task. We also choose a new network–VGG16 [17] pre-trained by [36], to test DiffPAD in a completely different task domain. We let GDPA attack the face images given its high success rate on VGG architectures [41]. By substituting the weights of the diffusion model with those pre-trained on the FFHQ [19] dataset as in [7], we obtain perfect facial restoration results, shown in Figure 4. VGG16 then regains the correct perdition with high confidence. The different samples convey the same success, manifesting the strong defensive capabilities of DiffPAD across various task domains.

Limitations and future works. Although our experiments illustrate the broad applicability of DiffPAD across various patch attack conditions, its practicability in real-time applications is constrained by the computational cost. It is worth noting that we optimize the efficiency of DiffPAD to execute only two rounds of diffusion generation per image, whereas DIFFender requires at least four rounds. Given the inherent complexity of diffusion models, the sampling process iterates multiple neural function evaluations (NFEs) at each time step, slowing down DiffPAD's inference speed compared to SAC and Jedi. Another limitation of DiffPAD is that adversarial patches are assumed to be or can be enclosed by a square. This assumption might not hold for more irregularly shaped patches such as adversarial eyeglasses [30]. Addressing these issues will be crucial for broadening the flexibility of DiffPAD in physical environments where such attacks may occur. Enhancing DiffPAD to overcome these limitations will increase its versatility as a tool for combating all adversarial patch attacks.

6. Conclusion

In this paper, we propose DiffPAD, a novel diffusion-based adversarial patch defense framework. DiffPAD effectively decontaminates adversarial patches through two stages of conditional diffusion generation. We first guide the diffusion to restore the downsampled input images by embedding a super-resolution function between two consecutive reverse samplings. We formulate the correlation between diffusion restoration error and patch size, which inspires a dynamic thresholded binarization scheme and a sliding window approach for precise patch localization. Finally, by swapping the super-resolution solution to the inpainting solution, the diffusion model seamlessly fills in the masked region with vivid visual details, erasing traces of the original patch. Extensive experiments across various attack methods, patch sizes, target models, datasets, and task domains, through comprehensive evaluation metrics, demonstrate that DiffPAD not only boosts adversarial robustness universally but also sustains the naturalistic integrity of images. As a framework built on the pre-trained diffusion model, DiffPAD achieves SOTA performance without the encumbrance of text guidance or fine-tuning, making it a potent and handy solution for adversarial patch defense.

Acknowledgment

This work was in part financially supported by the Digital Futures project Learning and Sharing under Privacy Constraints (DataLEASH). This work was also part of the project Swedish Wireless Innovation Network (SweWIN) approved by the Swedish Innovation Agency (VINNOVA). The computations were enabled by the resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council.

References

[1] Madry Aleksander, Makelov Aleksandar, Schmidt Ludwig, Tsipras Dimitris, and Vladu Adrian. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 1, 2

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283, 2018. 1, 2, 5

[3] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 1, 5

[4] Nicholas Carlini, Florian Tramer, Krishnamurthy Dj Dvijotham, Leslie Rice, Mingjie Sun, and J Zico Kolter. (certified!!) adversarial robustness for free! In *International Conference on Learning Representations*, 2022. 1

[5] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57, 2017. 1

[6] Xiao Chaowei, Chen Zhongzhu, Jin Kun, Wang Jiongxiao, Nie Weili, Liu Mingyan, Anandkumar Anima, Li Bo, and Song Dawn. Densepure: Understanding diffusion models for adversarial robustness. In *International Conference on Learning Representations*, 2023. 2

[7] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *International Conference on Computer Vision*, 2021. 3, 8

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, pages 8780–8794, 2021. 1, 5

[10] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016. 2, 6

[11] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. 1

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 1

[13] Mitch Hill, Jonathan Mitchell, and Song-Chun Zhu. Stochastic security: Adversarial defense using long-run dynamics of energy-based models. In *International Conference on Learning Representations*, 2021. 1

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851, 2020. 1, 3

[15] Buckman Jacob, Roy Aurko, Raffel Colin, and Goodfellow Ian. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018. 2

[16] Caixin Kang, Yinpeng Dong, Zhengyi Wang, Shouwei Ruan, Hang Su, and Xingxing Wei. Diffender: Diffusion-based adversarial defense against patch attacks in the physical world. *arXiv preprint arXiv:2306.09124*, 2023. 2, 4, 5, 6

[17] Simonyan Karen and Zisserman Andrew. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 8

[18] Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and visible adversarial noise. In *International Conference on Machine Learning*, pages 2507–2515, 2018. 1, 5

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 8

[20] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, pages 23593–23606, 2022. 3, 7

[21] Jiang Liu, Alexander Levine, Chun Pong Lau, Rama Chellappa, and Soheil Feizi. Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14973–14982, 2022. 1, 2, 6, 7

[22] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022. 5

[23] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 5

[24] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning*, pages 16805–16827, 2022. 1, 2, 5, 6

[25] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy*, pages 582–597, 2016. 2

[26] Omkar Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015. 5

[27] Chiang* Ping-yeh, Ni* Renkun, Abdelkader Ahmed, Zhu Chen, Studor Christoph, and Goldstein Tom. Certified defenses for adversarial patches. In *International Conference on Learning Representations*, 2020. 2

[28] Samangouei Pouya, Kabkab Maya, and Chellappa Rama. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018. 1, 2

[29] Sukrut Rao, David Stutz, and Bernt Schiele. Adversarial training against location-optimized adversarial patches. In *European Conference on Computer Vision*, pages 429–448, 2020. 2

[30] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. A general framework for adversarial examples with objectives. *ACM Transactions on Privacy and Security*, pages 1–30, 2019. 8

[31] Changhao Shi, Chester Holtz, and Gal Mishne. Online adversarial purification based on self-supervised learning. In *International Conference on Learning Representations*, 2020. 1, 2

[32] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018. 1, 2

[33] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 5

[34] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. 1

[35] Bilel Tarchoun, Anouar Ben Khalifa, Mohamed Ali Mahjoub, Nael Abu-Ghazaleh, and Ihsen Alouani. Jedi: entropy-based localization and removal of adversarial patches. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4087–4095, 2023. 1, 2, 6, 7

[36] Wu Tong, Tong Liang, and Vorobeychik Yevgeniy. Defending against physically realizable attacks on image classification. In *International Conference on Learning Representations*, 2020. 8

[37] Jinyi Wang, Zhaoyang Lyu, Dahua Lin, Bo Dai, and Hongfei Fu. Guided diffusion model for adversarial purification. *arXiv preprint arXiv:2205.14969*, 2022. 2

[38] 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, 2023. 1

[39] Xingxing Wei, Ying Guo, and Jie Yu. Adversarial sticker: A stealthy attack method in the physical world. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2711–2725, 2022. 1

[40] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295, 2018. 1, 2

[41] Li Xiang and Ji Shihao. Generative dynamic patch attack. *British Machine Vision Conference*, 2021. 1, 5, 8

[42] 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 *European Conference on Computer Vision*, pages 665–681, 2020. 1

[43] Song Yang, Sohl-Dickstein Jascha, P Kingma Diederik, Kumar Abhishek, Ermon Stefano, and Poole Ben. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2, 3

[44] Zhu Yuanzhi, Zhang Kai, Liang Jingyun, Cao Jiezhang, Wen Bihan, Timofte Radu, and Van Gool Luc. Denoising diffusion models for plug-and-play image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (NTIRE)*, 2023. 3

[45] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 6360–6376, 2021. 4

Supplementary Materials for DiffPAD: Denoising Diffusion-based Adversarial Patch Decontamination

Jia Fu[1,2] Xiao Zhang[3] Sepideh Pashami[1,4] Fatemeh Rahimian[1] Anders Holst[1,2]
[1]RISE Research Institutes of Sweden [2]KTH Royal Institute of Technology
[3]CISPA Helmholtz Center for Information Security [4]Halmstad University
{jia.fu, sepideh.pashami, fatemeh.rahimian, anders.holst}@ri.se xiao.zhang@cispa.de

1. Additional discussions on related work

In this section, we provide more detailed discussions of related works on adversarial patch attacks and diffusion-based adversarial defenses.

1.1. Adversarial patch attacks

Since Szegedy *et al.* [10] revealed the adversarial vulnerabilities of neural networks, where normal inputs crafted with imperceptible perturbations can induce erroneous predictions, numerous attack algorithms [1, 3, 4] have been proposed to study the model behavior in the presence of adversarial examples. However, most existing works focused on global attacks defined by some ℓ_p-norm, thereby not directly applicable to threatening real-world systems. Brown *et al.* [2] first introduced the concept of adversarial patches, where the adversary is only allowed to manipulate a small region of an image to launch the evasion attack. Subsequently, LaVAN [6] enhanced the design of the loss function, enabling the adversarial patch to cover only 2% of the given image. Meanwhile, GDPA [13] improved the attack strategy by adversarially refining the patch's location rather than positioning it randomly. These research efforts lay the foundation for realizing adversarial patches in the physical world. For example, an adversarial patch printed on a T-shirt [14] can succeed in evading human detectors, while Wei *et al.* [12] proposed adversarial stickers, which feature meaningful patterns and achieve good performance in both digital and physical realms.

1.2. Diffusion-based adversarial defenses

We further discuss the limitations of existing diffusion-based adversarial defenses, including DiffPure and DIFFender. DiffPure [8] has proved that forward diffusion disrupts the distribution of both clean data and adversarial perturbations. During the reverse diffusion process, clean data can be stochastically recovered, while adversarial effects are progressively eliminated. This process can be executed using the standard DDPM framework. Necessarily, to preserve the label semantics of the image, DiffPure halts the diffusion at a specific timestep $t^* \in (0, T)$ then commences the reverse diffusion from $\boldsymbol{x}_{t^*}$ back to $\boldsymbol{x}_0$. DIFFender [5] identified a critical limitation of DiffPure in adversarial patch defense. DiffPure struggles to completely remove the adversarial patch, which requires a larger t^*, whereas a smaller t^* is essential for maintaining image semantics. Alternatively, DIFFender retains image semantics with the aid of additional prompts and fine-tunes a text-guided diffusion model for patch localization and restoration. However, prompt learning introduces new challenges, as well as limited prior contained within the text prompts renders DIFFender less efficient, necessitating the generation of at least three samples per image to ensure robust patch localization.

2. Proof of Theorem 1

For the sake of completeness, we provide detailed proof of our main theoretical result presented in Section 4.2. Our proof technique mainly follows from the proof of Theorem 3.2 in [8]. Below, we first restate the problem statement of Theorem 1 that we are going to prove.

Theorem 1 *Assume* $\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t)\| \leq C_\epsilon\sqrt{1-\bar{\alpha}_t}$ *and let* $\gamma := \int_0^T \beta_t \mathrm{d}t$. *With probability at least* $1-\xi$, *the* ℓ_2 *distance between the diffusion-purified image* $\hat{\boldsymbol{x}^a}$ *with adversarial patch and the corresponding clean image* $\boldsymbol{x}^c$ *satisfies:*

$$\|\hat{\boldsymbol{x}^a} - \boldsymbol{x}^c\| \leq \varepsilon\,|\mathbf{A}| + \gamma C_\epsilon + \sqrt{e^\gamma - 1}\cdot C_\xi, \tag{12}$$

where ε *is the* ℓ_2*-norm bound of the patch,* $C_\xi := \sqrt{2d + 4\sqrt{d\log\frac{1}{\xi}} + 4\log\frac{1}{\xi}}$, *and* d *is the input dimension.*

Proof: For variance preserving SDE, given the adversarial example $\boldsymbol{x}^a$ defined in Equation 8, after the forward diffusion process, we have

$$\boldsymbol{x}_T = \sqrt{\alpha_T}\cdot\boldsymbol{x}^a + \sqrt{1-\alpha_T}\cdot\boldsymbol{\epsilon}', \tag{15}$$

where $\alpha_T = e^{-\int_0^T \beta_t \mathrm{d}t}$ and $\boldsymbol{\epsilon}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. As diffusion-restored adversarial example $\hat{\boldsymbol{x}^a}$ does not have a closed-form solution, we apply an SDE solver with the Euler–Maruyama discretization, where the drift and diffusion coefficients of the reverse-time SDE are given by:

$$\begin{aligned} \boldsymbol{f}_{\mathrm{rev}}(\boldsymbol{x}, t) &:= -\frac{1}{2}\beta_t \left[\boldsymbol{x} + 2\boldsymbol{s}_\theta(\boldsymbol{x}_t)\right], \\ g_{\mathrm{rev}}(t) &:= \sqrt{\beta_t}, \end{aligned} \tag{16}$$

where $\boldsymbol{s}_\theta(\boldsymbol{x}_t)$ denotes the score function. The ℓ_2 distance between $\hat{\boldsymbol{x}^a}$ and the corresponding clean data $\boldsymbol{x}^c$ can be bounded as:

$$\begin{aligned} &\|\hat{\boldsymbol{x}^a} - \boldsymbol{x}^c\| = \left\|\boldsymbol{x}_T + \left(\hat{\boldsymbol{x}^a} - \boldsymbol{x}_T\right) - \boldsymbol{x}^c\right\| \\ &= \|\boldsymbol{x}_T + \int_T^0 -\frac{1}{2}\beta_t \left[\boldsymbol{x} + 2\boldsymbol{s}_\theta(\boldsymbol{x}_t)\right] \mathrm{d}t + \int_T^0 \sqrt{\beta_t}\mathrm{d}\boldsymbol{w} - \boldsymbol{x}^c\| \\ &\leq \| \underbrace{\boldsymbol{x}_T + \int_T^0 -\frac{1}{2}\beta_t \boldsymbol{x}\mathrm{d}t + \int_T^0 \sqrt{\beta_t}\mathrm{d}\boldsymbol{w}}_{\text{Integration of linear SDE}} - \boldsymbol{x}^c\| \\ &\qquad + \| \int_T^0 -\beta_t \boldsymbol{s}_\theta(\boldsymbol{x}_t)\mathrm{d}t\|, \end{aligned} \tag{17}$$

where the second equation is obtained by using the integration of the reverse-time SDE, and the last line is derived by separating the integration of the linear SDE from non-linear SDE involving $\boldsymbol{s}_\theta(\boldsymbol{x}_t)$ through the triangle inequality.

Notice that the above linear SDE is a time-varying Ornstein–Uhlenbeck process, where the time increment inversely starts from T to 0 with the initial value $\boldsymbol{x}_T$. Denote its solution by $\boldsymbol{x}'$ that follows a Gaussian distribution, the mean $\boldsymbol{\mu}_0$ and covariance matrix $\boldsymbol{\Sigma}_0$ of $\boldsymbol{x}'$ will be the solutions of the following two differential equations:

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{\mu}}{\mathrm{d}t} &= -\frac{1}{2}\beta_t \boldsymbol{\mu}, \\ \frac{\mathrm{d}\boldsymbol{\Sigma}}{\mathrm{d}t} &= -\beta_t \boldsymbol{\Sigma} + \beta_t \mathbf{I}_d, \end{aligned} \tag{18}$$

with the initial conditions $\boldsymbol{\mu}_T = \boldsymbol{x}_T$ and $\boldsymbol{\Sigma}_T = \mathbf{0}$. By solving these two differential equations, we have $\boldsymbol{x}' \sim \mathcal{N}\left(e^{\frac{\gamma}{2}}\boldsymbol{x}_T, (e^\gamma - 1)\mathbf{I}_d\right)$ that is conditioned on $\boldsymbol{x}_T$, where $\gamma := \int_0^T \beta_t \mathrm{d}t$. Taking the advantage of reparameterization trick, we obtain

$$\begin{aligned} &\boldsymbol{x}' - \boldsymbol{x}^c \\ &= e^{\frac{\gamma}{2}}\boldsymbol{x}_T + \sqrt{e^\gamma - 1}\cdot\boldsymbol{\epsilon}'' - \boldsymbol{x}^c \\ &= e^{\frac{\gamma}{2}}\left(e^{-\frac{\gamma}{2}}\boldsymbol{x}^a + \sqrt{1 - e^{-\gamma}}\cdot\boldsymbol{\epsilon}'\right) + \sqrt{e^\gamma - 1}\cdot\boldsymbol{\epsilon}'' - \boldsymbol{x}^c \\ &= \sqrt{e^\gamma - 1}\cdot(\boldsymbol{\epsilon}' + \boldsymbol{\epsilon}'') + \boldsymbol{x}^a - \boldsymbol{x}^c, \end{aligned} \tag{19}$$

where the second equation follows by substituting Equation 15. Since $\boldsymbol{\epsilon}'' \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ and $\boldsymbol{\epsilon}' \perp \boldsymbol{\epsilon}''$, the first term of the last line in Equation 19 can be combined as a zero-mean Normal variable with variance $2(e^\gamma - 1)$.

We know the connection between the score function and the noise prediction $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t)$ in DDPM can be formulated as:

$$\boldsymbol{s}_\theta(\boldsymbol{x}_t) = -\frac{\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t)}{\sqrt{1 - \bar{\alpha}_t}}. \tag{20}$$

Assuming that the ℓ_2-norm of $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t)$ is upper-bounded by $C_\epsilon\sqrt{1 - \bar{\alpha}_t}$. In other words, we assume that the ℓ_2-norm of $\boldsymbol{s}_\theta(\boldsymbol{x}_t)$ is upper-bounded by constant C_ϵ. Hence,

$$\begin{aligned} \|\hat{\boldsymbol{x}^a} - \boldsymbol{x}^c\| &\leq \|\sqrt{2(e^\gamma - 1)}\cdot\boldsymbol{\epsilon} + \boldsymbol{x}^a - \boldsymbol{x}^c\| + \gamma C_\epsilon \\ &\leq \|\boldsymbol{x}^a - \boldsymbol{x}^c\| + \gamma C_\epsilon + \sqrt{2(e^\gamma - 1)}\cdot\|\boldsymbol{\epsilon}\|, \end{aligned} \tag{21}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. We denote the ℓ_2-norm bound of the pixels in adversarial patch region as ε, since $\boldsymbol{x}^a - \boldsymbol{x}^c = \mathbf{A} \odot (\boldsymbol{\delta} - \boldsymbol{x}^c)$, we can obtain $\|\boldsymbol{x}^a - \boldsymbol{x}^c\| \leq \varepsilon|\mathbf{A}|$, where $|\mathbf{A}|$ represents the pixel number, i.e., the size of adversarial patch. Furthermore, $\|\boldsymbol{\epsilon}\|^2 \sim \chi^2(d)$, from the concentration inequality, we attain

$$\Pr\left(\|\boldsymbol{\epsilon}\|^2 \geq d + 2\sqrt{d\sigma} + 2\sigma\right) \leq e^{-\sigma}. \tag{22}$$

Let $e^{-\sigma} = \xi$, we get

$$\Pr\left(\|\boldsymbol{\epsilon}\| \geq \sqrt{d + 2\sqrt{d\log\frac{1}{\xi}} + 2\log\frac{1}{\xi}}\right) \leq \xi. \tag{23}$$

Finally, at least of the probability $1 - \xi$, we have

$$\|\hat{\boldsymbol{x}^a} - \boldsymbol{x}^c\| \leq \varepsilon|\mathbf{A}| + \gamma C_\epsilon + \sqrt{e^\gamma - 1}\cdot C_\xi, \tag{24}$$

where constant $C_\xi := \sqrt{2d + 4\sqrt{d\log\frac{1}{\xi}} + 4\log\frac{1}{\xi}}$, which completes the proof of Theorem 1.

3. Experimental details

3.1. Hyperparameter setup

All our experiments are conducted in Pytorch on four Nvidia A100 GPUs. We set $\mu = 0.066$ and $\nu = 14.90$ in Equation 14, which is determined using grid search. In practice, to reduce the redundant computations, the threshold τ' is fixed as 9. We treat input images with diffusion restoration errors less than 62 as clean images to prevent excess defense. We run 20 NFEs for both super-resolution and inpainting restoration. Noise level $\sigma = 0.001$ and scaling factor $s = 4$ are hyperparameters in close-form solutions (Equation 10, 11). Additionally, we repeat three rounds of each experiment related to DiffPAD and report averaged statistics, due to the stochasticity of diffusion processes. In the evaluation phase, we adopt the same subset of the original ImageNet validation set as [9], which contains 1000 images covering all categories. For a fair comparison with DIFFender, we randomly choose 512 images from this subset which can be correctly classified before the attacks.

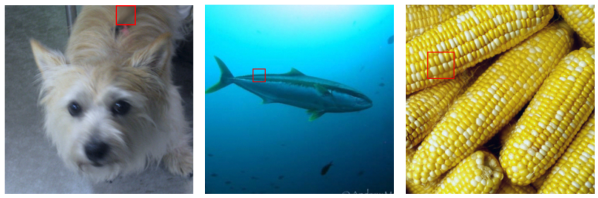

Figure 1. Examples of clean images where DiffPAD spuriously detects an adversarial patch of small size (marked by the red box).

Table 1. Comparisons of robust accuracies (%) against global attacks on ImageNet with Inception-V3. The best (blue) and second-best (red) results are highlighted. PAD stands for patch detection.

Defense \ Attack	FGSM	PGD	C&W
w/o defense	14.3	0.2	0.1
JPG	27.6	10.6	34.9
SAC	19.6	2.8	4.0
Jedi	25.9	5.6	22.5
DiffPure	64.4	64.6	65.8
DiffPAD w/o PAD	50.3	51.1	53.3

3.2. False positive of patch detection

Figure 1 visualizes how clean images appear when processed with DiffPAD. We can see that the estimated patches are quite small. The inpainting is competent in recovering an image almost identical to its original version, thereby avoiding excessive defense and ensuring the recognition performance remains unaffected on the clean dataset. This is also confirmed by the clean accuracies of DiffPAD, which is always the highest compared to the other defenses.

3.3. Computational complexity

For each image resized to 256×256, SAC [7] costs 0.27s, Jedi [11] costs 0.32s, DiffPAD costs 2.45s, and DiffPure costs 8.59s, on average.

4. Generalizability to global attacks

Although DiffPAD targets localized patch attacks, the proposed diffusion-based resolution degradation-restoration mechanism can serve as a handy tool to mitigate ℓ_p-norm bounded perturbations. Table 1 compares the robust accuracies of DiffPAD with other baselines used in the main paper against FGSM [4], PGD [1], and C&W [3] attacks. The trivial image transformation and other patch defenses demonstrate limited effectiveness, far less than the SOTA model DiffPure in such attack settings. However, DiffPAD (40 NFEs) is second only to DiffPure and achieves 80% of its performance, taking only 30% of its runtime.

References

[1] Madry Aleksander, Makelov Aleksandar, Schmidt Ludwig, Tsipras Dimitris, and Vladu Adrian. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 1, 3

[2] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 1

[3] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57, 2017. 1, 3

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 1, 3

[5] Caixin Kang, Yinpeng Dong, Zhengyi Wang, Shouwei Ruan, Hang Su, and Xingxing Wei. Diffender: Diffusion-based adversarial defense against patch attacks in the physical world. *arXiv preprint arXiv:2306.09124*, 2023. 1

[6] Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and visible adversarial noise. In *International Conference on Machine Learning*, pages 2507–2515, 2018. 1

[7] Jiang Liu, Alexander Levine, Chun Pong Lau, Rama Chellappa, and Soheil Feizi. Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14973–14982, 2022. 3

[8] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning*, pages 16805–16827, 2022. 1

[9] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. In *European Conference on Computer Vision*, 2020. 2

[10] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. 1

[11] Bilel Tarchoun, Anouar Ben Khalifa, Mohamed Ali Mahjoub, Nael Abu-Ghazaleh, and Ihsen Alouani. Jedi: entropy-based localization and removal of adversarial patches. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4087–4095, 2023. 3

[12] Xingxing Wei, Ying Guo, and Jie Yu. Adversarial sticker: A stealthy attack method in the physical world. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2711–2725, 2022. 1

[13] Li Xiang and Ji Shihao. Generative dynamic patch attack. *British Machine Vision Conference*, 2021. 1

[14] 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 *European Conference on Computer Vision*, pages 665–681, 2020. 1