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:*

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

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

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

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:

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

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:

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

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 $\boldsymbol{x}' \sim \mathcal{N}\left(e^{\frac{\gamma}{2}}\boldsymbol{x}_T, \left(e^\gamma - 1\right)\mathbf{I}_d\right)$ 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

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

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 $2\left(e^\gamma - 1\right)$.

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,

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

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 $\|\boldsymbol{x}^a - \boldsymbol{x}^c\| \le \varepsilon\,|\mathbf{A}|$, 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

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

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

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

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

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

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.

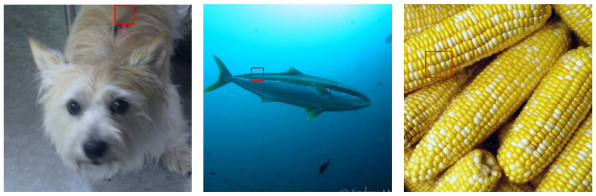

Figure 1. Examples of clean images where DiffPAD spuriously detects an adversarial patch of small size (marked by the red box).

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