# Adversarial Purification via Super-Resolution and Diffusion

## Supplementary Material

### 1. Proof of Proposition 1

*Proof.* Let the SR model $\mathcal{H}_{\omega^*}: \mathbb{R}^d \mapsto \mathbb{R}^D$ be ideally optimized, smooth, and continuously differentiable function. Also, let the degeneration function $\mathfrak{D}: \mathbb{R}^d \mapsto \mathbb{R}^d$ be differentiable. Given an input $\mathfrak{D}(\mathbf{x}_0)$, we can apply a first-order Taylor expansion of $\mathcal{R}\mathcal{H}_{\omega^*}$ around the data $\mathbf{x}_0$:

$$\begin{aligned}\mathcal{R}\mathcal{H}_{\omega^*}(\mathfrak{D}(\mathbf{x}_0)) = \mathcal{R}\mathcal{H}_{\omega^*}(\mathbf{x}_0) \\ + \mathbf{J}_{\mathcal{R}\mathcal{H}_{\omega^*}}(\mathbf{x}_0)(\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0) \\ + \mathcal{O}(\|\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0\|^2),\end{aligned} \tag{1}$$

where $\mathcal{R}: \mathbb{R}^D \mapsto \mathbb{R}^d$ is a linear transformation used for image downsampling and $\mathbf{J}_{\mathcal{R}\mathcal{H}_{\omega^*}}(\mathbf{x}_0)$ is the Jacobian matrix of $\mathcal{R}\mathcal{H}_{\omega^*}(\mathbf{x}_0))$ at $\mathbf{x}_0$. $\mathcal{O}(\cdot)$ represents higher-order term.

By assumption, the SR model $\mathcal{H}_{\omega^*}$ is perfectly optimized by the log-posterior [30]. Then, the observation model in SR satisfies:

$$\mathcal{R}\mathcal{H}_{\omega^*}(\mathbf{x}_0) \simeq \mathbf{x}_0. \tag{2}$$

Since the data manifold $\mathcal{M}$ is also smooth, we consider its local structure around $\mathbf{x}_0$. In a manifold assumption [5], $\mathcal{M}$ is locally linear and well approximated by its tangent space $T_{\mathbf{x}_0}\mathcal{M}$ in a sufficiently small neighborhood of $\mathbf{x}_0$:

$$\mathcal{M} \cap \mathcal{B}(\mathbf{x}_0, dr) = T_{\mathbf{x}_0}\mathcal{M} \cap \mathcal{B}(\mathbf{x}_0, dr), \quad T_{\mathbf{x}_0}\mathcal{M} \simeq \mathbb{R}^k, \tag{3}$$

where $k \ll d$. This implies that the Jacobian $\mathbf{J}_{\mathcal{R}\mathcal{H}_{\omega^*}}(\mathbf{x}_0)$ transforms inputs along directions within $T_{\mathbf{x}_0}\mathcal{M}$ while minimally distorting the local geometry of the manifold. Thus, we approximate:

$$\mathbf{J}_{\mathcal{R}\mathcal{H}_{\omega^*}}(\mathbf{x}_0)(\cdot) \approx \Pi_{T_{\mathbf{x}_0}\mathcal{M}}(\cdot). \tag{4}$$

Substitute the approximation into the Taylor expansion:

$$\begin{aligned}\mathcal{R}\mathcal{H}_{\omega^*}(\mathfrak{D}(\mathbf{x}_0)) = \mathcal{R}\mathcal{H}_{\omega^*}(\mathbf{x}_0) \\ + \Pi_{T_{\mathbf{x}_0}\mathcal{M}}(\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0) \\ + \mathcal{O}(\|\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0\|^2).\end{aligned} \tag{5}$$

Substitute that $\mathcal{R}\mathcal{H}_{\omega^*}(\mathbf{x}_0)$ behaves a near-identity mapping for $\mathbf{x}_0$ by our assumption. Then, we have:

$$\mathcal{R}\mathcal{H}_{\omega^*}(\mathfrak{D}(\mathbf{x}_0)) \approx \Pi_{T_{\mathbf{x}_0}\mathcal{M}}(\mathfrak{D}(\mathbf{x}_0)) + \mathcal{O}(\|\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0\|^2). \tag{6}$$

Now, define the residual error $\delta$ between the downsampled SR data and its projection onto the tangent space:

$$\delta = \|\mathcal{R}\mathcal{H}_{\omega^*}(\mathfrak{D}(\mathbf{x}_0)) - \Pi_{T_{\mathbf{x}_0}\mathcal{M}}(\mathfrak{D}(\mathbf{x}_0))\|_F, \tag{7}$$

where $\|\cdot\|_F$ is the Frobenius norm. The higher-order term $\mathcal{O}(\|\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0\|^2)$ captures all higher-order deviations, we can bound it by introducing a constant $C > 0$:

$$\|\mathcal{O}(\|\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0\|^2)\|_F \leq C\|\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0\|^2. \tag{8}$$

Thus, we have:

$$\delta \leq C\|\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0\|^2. \tag{9}$$

Now, assuming that $\mathfrak{D}(\mathbf{x}_0)$ remains sufficiently closer to $\mathbf{x}_0$, i.e., $\mathfrak{D}(\mathbf{x}_0) \in \mathcal{B}(\mathbf{x}_0, \epsilon)$, we set:

$$\|\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0\| < \epsilon. \tag{10}$$

Substituting this bound:

$$\|\mathcal{R}\mathcal{H}_{\omega^*}(\mathfrak{D}(\mathbf{x}_0)) - \Pi_{T_{\mathbf{x}_0}\mathcal{M}}(\mathfrak{D}(\mathbf{x}_0))\|_F = \delta \leq C\epsilon^2. \tag{11}$$

Since $\epsilon$ can be made arbitrarily small for sufficiently minor degradation, such as adversarial attacks, we conclude $\delta \to 0$ as $\epsilon \to 0$. This completes the proof. □

### 2. Incorporating Curvature into the Proof

In this section, we explicitly incorporate curvature effects into the error bound by refining our previous analysis.

Given an input $\mathfrak{D}(\mathbf{x}_0)$, we apply a second-order Taylor expansion of $\mathcal{R}\mathcal{H}_{\omega^*}$ around $\mathbf{x}_0$:

$$\begin{aligned}\mathcal{R}\mathcal{H}_{\omega^*}(\mathfrak{D}(\mathbf{x}_0)) = \mathcal{R}\mathcal{H}_{\omega^*}(\mathbf{x}_0) \\ + \mathbf{J}_{\mathcal{R}\mathcal{H}_{\omega^*}}(\mathbf{x}_0)(\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0) \\ + \frac{1}{2}(\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0)^T \mathbf{H}_{\mathcal{R}\mathcal{H}_{\omega^*}}(\mathbf{x}_0)(\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0) \\ + \mathcal{O}(\|\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0\|^3),\end{aligned} \tag{12}$$

where $\mathbf{J}_{\mathcal{R}\mathcal{H}_{\omega^*}}(\mathbf{x}_0)$ is the Jacobian matrix capturing the first-order derivatives and $\mathbf{H}_{\mathcal{R}\mathcal{H}_{\omega^*}}(\mathbf{x}_0)$ is the Hessian matrix capturing second-order curvature effects. $\mathcal{O}(\|\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0\|^3)$ represents higher-order deviations.

From a differential geometry, the deviation between a point $\mathfrak{D}(\mathbf{x}_0)$ and its projection onto the tangent space $T_{\mathbf{x}_0}\mathcal{M}$ is governed by the second fundamental form, $S_{\mathcal{M}}(\mathbf{v}, \mathbf{v})$. This measures the curvature of $\mathcal{M}$ along unit direction $\mathbf{v}$, determining how much $\mathcal{M}$ deviates from its tangent space:

$$\begin{aligned}\|\Pi_{\mathcal{M}}(\mathfrak{D}(\mathbf{x}_0)) - \Pi_{T_{\mathbf{x}_0}\mathcal{M}}(\mathfrak{D}(\mathbf{x}_0))\| \\ \approx \frac{1}{2}\|\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0\|^2 \sup_{\|\mathbf{v}\|=1} |S_{\mathcal{M}}(\mathbf{v}, \mathbf{v})|.\end{aligned} \tag{13}$$

Thus, we approximate the second-order term in the Taylor expansion as:

$$(\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0)^T \mathbf{H}_{\mathcal{R}\mathcal{H}_{\omega^*}}(\mathbf{x}_0)(\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0) \approx \sup_{\|\mathbf{v}\|=1} |S_{\mathcal{M}}(\mathbf{v}, \mathbf{v})| \|\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0\|^2. \tag{14}$$

Now, substituting the curvature deviation bound into our error bound from Eq 9:

$$\delta \leq \frac{1}{2}\|\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0\|^2 \sup_{|\mathbf{v}|=1} |S_{\mathcal{M}}(\mathbf{v}, \mathbf{v})| + C'\|\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0\|^3, \tag{15}$$

where $C' > 0$ is a constant that can bound higher-order deviations. Since we assume $\mathfrak{D}(\mathbf{x}_0)$ remains within a small neighborhood, i.e., $\|\mathfrak{D}(\mathbf{x}_0) - \mathbf{x}_0\| < \epsilon$, we substitute:

$$\delta \leq \frac{1}{2}\epsilon^2 \sup_{|\mathbf{v}|=1} |S_{\mathcal{M}}(\mathbf{v}, \mathbf{v})| + C'\epsilon^3. \tag{16}$$

Since $\epsilon$ is sufficiently small, the higher-order term $C'\epsilon^3$ diminishes faster than the quadratic term. Then, we have:

$$\delta \approx \frac{1}{2}\epsilon^2 \sup_{|\mathbf{v}|=1} |S_{\mathcal{M}}(\mathbf{v}, \mathbf{v})| \quad \text{as} \quad \epsilon \to 0. \tag{17}$$

Thus, the error bound $\delta$ decreases at a rate of $\mathcal{O}(\epsilon^2)$, ensuring that it vanishes as $\epsilon \to 0$.

## 3. Details of Experimental Environment

Evaluations of adversarial defenses are conducted by using the PyTorch on adaptive white-box attacks (BPDA+EOT, AutoAttack, PGD+EOT, and DiffAttack), a preprocessor-blind attack (PGD), and a black-box attack (Square Attack). The experiments utilized two infrastructures: one with 8 NVIDIA RTX 4090 GPUs for task evaluation and another with 4 NVIDIA A100 GPUs for white-box attack evaluation. Off-the-shelf DNN models were obtained from `torchvision`, Hugging-Face, and `timm`. Adversarial defense setups followed those of DiffPure [21] and ScoreOpt [36], incorporating SR models specific to each method.

We outline the package hubs overview as follows:

- `torchvision`: This package includes a wide range of datasets, pre-trained DNN architectures, and typical image transformations for computer vision tasks.
- Hugging-Face: An extensive hub offering ready-to-use foundation models and neural network architectures.
- `timm`: A repository providing state-of-the-art computer vision models, facilitating ImageNet training replication.

## 4. Implementation Details of PuriFlow

**DNN models** We describe the sources of the pre-trained models used in our study:

- WRN-28-10 and WRN-70-16: Checkpoints sourced from the official `Robstbench` leaderboard [8].
- ResNet-50: Sourced from `torchvision` model zoo.
- DeiT-S: Official pre-trained checkpoint from Hugging-Face, released by Meta research.
- BEiT-L: Utilizes a pre-trained checkpoint from `timm`.
- VP-SDE: Adopted the 256×256 unconditional diffusion model from OpenAI's guided-diffusion library.
- EDM: Adopted the 32×32 unconditional diffusion model from NVIDIA Lab for one-shot denoising.
- MDSR: Checkpoint sourced from Hugging-Face.
- EDSR: Checkpoint sourced from Hugging-Face.
- DRLN: Checkpoint sourced from Hugging-Face.
- ESRGAN: Sourced from the Real-ESRGAN custom-reproduced library, with an official pre-trained checkpoint from Real-ESRGAN [31].

**Benchmarks on adversarial training** We describe the sources of pre-trained WRN-28-10, WRN-70-16, and ResNet-50 checkpoints on adversarial training used for standard and robust accuracy comparison:

- Checkpoints obtained from the official `RobustBench` leaderboard [8].

**Adversarial attacks** The implementation sources for the adversarial attacks used in our study:

- BPDA+EOT: Based on the official project [36], with $N = 50$ attack iterations and 15 EOT iterations, following [36].
- AutoAttack: Adopted from the official project [21] and implemented according to the strategies in [21].
- PGD+EOT: Implemented from [19], with $N = 200$ attack iterations, 20 EOT iterations, and a step size of $\mu = 2/255$, based on [19].
- DiffAttack: Derived from the official project [16].
- PGD: Sourced from the `torchattacks` library, with $N = 100$ attack iterations and a step size of $\mu = 2/255$, following the adversarial attack outlined in Eq. 6.
- Square Attack: Sourced from the official AutoAttack project [7], with a query limit of $N = 5000$.

**Ablation study** Details of pre-trained models and algorithms used in our ablation studies:

- Wavelet (Denoising): Official implementation from the `scikit-learn` package.
- TVM (Denoising): Official implementation from the `scikit-learn` package.
- NL-means (Denoising): Official implementation from the `scikit-learn` package.
- VE-SDE: Adopted the 256×256 unconditional diffusion model from OpenAI's guided-diffusion library.

*For the use of the SR model, we required a scalable pre-trained model supporting multiple upscaling ratios (e.g.,* $\times 2$, $\times 4$, $\times 8$*) to evaluate its interaction with diffusion models. ESRGAN was the only option that met these criteria, making it an unavoidable choice.*

**Algorithm 1** (PuriFlow) SR integrated with VP-SDE

**Input:** Image $\mathbf{x}$
**Given:** $\mathcal{H}_\omega; \mathbf{s}_{\theta^*}; \mathcal{F}_\phi; t'; \{\alpha_t\}_{t=1}^{t'}$
  $\mathbf{x}' \leftarrow \mathcal{H}_\omega(\mathbf{x})$ ▷ SR on unknown $\mathbf{x} \in \{\mathbf{x}_0, \mathbf{x}_0^{adv}\}$
  $\mathbf{x}' \leftarrow \mathcal{R}(\mathbf{x}')$ ▷ Downscale from $\mathbb{R}^D$ to $\mathbb{R}^d$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$
  $\mathbf{x}_{t'} \leftarrow \sqrt{\bar{\alpha}_{t'}}\mathbf{x}' + \sqrt{1-\bar{\alpha}_{t'}}\mathbf{z}$
  $\mathbf{x}_0 \leftarrow \text{SDEINT}(\mathbf{x}_{t'}, \boldsymbol{f}(\cdot, t), g(t), \bar{\mathbf{w}}, t', 0)$ ▷ until $t'$ is equal to 0
  $k^* \leftarrow \text{argmax}_{k\in[1,K]}\mathcal{F}_\phi(\mathbf{x}_0)$
**Output:** Predicted label $k^*$

**Algorithm 2** (PuriFlow) SR integrated with OSD

**Input:** Image $\mathbf{x}$
**Given:** $\mathcal{H}_\omega; \mathbf{s}_{\theta^*}; \mathcal{F}_\phi; t'; \sigma_{t'}^2$
  $\mathbf{x}' \leftarrow \mathcal{H}_\omega(\mathbf{x})$ ▷ SR on unknown $\mathbf{x} \in \{\mathbf{x}_0, \mathbf{x}_0^{adv}\}$
  $\mathbf{x}' \leftarrow \mathcal{R}(\mathbf{x}')$ ▷ Downscale from $\mathbb{R}^D$ to $\mathbb{R}^d$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$
  $\mathbf{x}_{t'} \leftarrow \sqrt{\bar{\alpha}_{t'}}\mathbf{x}' + \sqrt{1-\bar{\alpha}_{t'}}\mathbf{z}$
  $\mathbf{x}_0 \leftarrow \mathbf{x}_{t'} + \sigma_{t'}^2 \mathbf{s}_{\theta^*}(\mathbf{x}_{t'}, t')$
  $k^* \leftarrow \text{argmax}_{k\in[1,K]}\mathcal{F}_\phi(\mathbf{x}_0)$
**Output:** Predicted label $k^*$

**Algorithm 3** (PuriFlow) SR integrated with iterative OSD

**Input:** Image $\mathbf{x}$
**Given:** $\mathcal{H}_\omega; \mathbf{s}_{\theta^*}; \mathcal{F}_\phi; t'; \sigma_{t'}^2$; M
  $\mathbf{x}' \leftarrow \mathcal{H}_\omega(\mathbf{x})$ ▷ SR on unknown $\mathbf{x} \in \{\mathbf{x}_0, \mathbf{x}_0^{adv}\}$
  $\mathbf{x}' \leftarrow \mathcal{R}(\mathbf{x}')$ ▷ Downscale from $\mathbb{R}^D$ to $\mathbb{R}^d$
  $\mathbf{x}^0 \leftarrow \mathbf{x}'$
  **for** $i = M$ **to** 1 **do** ▷ $M$-th optimization iterations
    $\mathbf{z}_1, \mathbf{z}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$
    $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}_d)$
    $\mathbf{x}_{t'}^i \leftarrow \mathbf{x}^i + \sigma_{t'}\mathbf{z}_1$
    $\mathbf{x}_{t'}^c \leftarrow \mathbf{x}' + \sigma_{t'}\mathbf{z}_2$
    $g \leftarrow \nabla_{\mathbf{x}^i}[||\mathbf{x}_{t'}^i + \sigma_{t'}^2 \mathbf{s}_{\theta^*}(\mathbf{x}_{t'}^i, t') - \mathbf{x}^i||_2^2 + ||\mathbf{x}^i - \mathbf{x}' + \sigma_{t'}^2(\mathbf{s}_{\theta^*}(\mathbf{x}_{t'}^i, t') - \mathbf{s}_{\theta^*}(\mathbf{x}_{t'}^c, t'))||_2^2]$
    $\mathbf{x}_{i-1} \leftarrow \mathbf{x}^i - \eta \cdot g$
  **end for**
  $k^* \leftarrow \text{argmax}_{k\in[1,K]}\mathcal{F}_\phi(\mathbf{x}_0)$
**Output:** Predicted label $k^*$

**Dataset** We evaluate PuriFlow against previous adversarial defense methods using the ImageNet-1k dataset [9], which contains 50k validation images, and CIFAR-10, comprising 10k test images. For robust accuracy, following prior research [3, 20, 21, 36], we use a subset of 512 validation and test images from both datasets. Standard accuracy is measured on the full dataset, which is also used to evaluate defenses against the pre-processor blind attack (PGD).

## 5. PuriFlow Process in Algorithm

We present PuriFlow, an adversarial purification method that combines an SR model with diffusion models based on VP-SDE [21] and iterative OSD [36]. The drift coefficients for the reverse-time SDE in VP-SDE are defined as:

$$\boldsymbol{f}(\mathbf{x}_t, t) = -\frac{1}{2}(1-\alpha_t)\Big(\mathbf{x}_t + 2\mathbf{s}_{\theta^*}(\mathbf{x}_t, t)\Big), \quad (18)$$

$$g(t) = \sqrt{1-\alpha_t}, \quad (19)$$

where $\alpha_t = 1 - \beta_t$. SR-combined VP-SDE process is described in Algorithm 1. Additionally, in this supplementary material, we evaluate the certified robustness of the OSD method, following [4].

**SR-integrated One-Shot Denoisor (OSD).** The one-shot denoiser, a special case of the reverse diffusion process, decouples the temporal sequences of the reverse-time SDE and is applied either once or iteratively with a fixed time step $t'$. For example, SR can replace the original reverse-time SDE in a one-shot scheme, as shown in Algorithm 2. Furthermore, SR can be incorporated into one-shot denoising optimization methods, as described in Algorithm 3 [36].

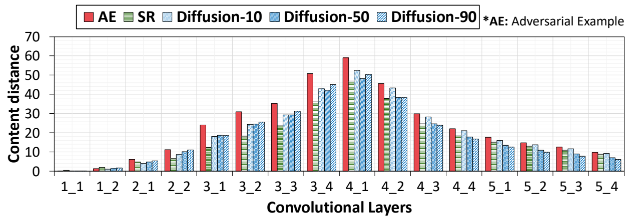
| Type | Method | Standard (%) | Robust (%) |
|---|---|---|---|
| *WRN-28-10* | | 95.63 | 0.00 |
| AT | Pang et al. [22] | 88.62 | 64.95 |
| AT | Gowal et al. [11] | 88.54 | 65.93 |
| AT | Gowal et al. [12] | 87.51 | 66.01 |
| AP | Wu et al. [33] | 89.16 | 70.07 |
| AP | Nie et al. [21] | 89.44 | 70.64 |
| AP | Bai et al. [1] | **91.41** | 82.81 |
| AP | **PuriFlow([21])** | 90.06 | **87.10** |
| *WRN-70-16* | | 95.79 | 0.00 |
| AT | Gowal et al. [11] | 85.29 | 68.66 |
| AT | Gowal et al. [12] | 88.74 | 69.03 |
| AT | Rebuffi et al. [27] | 88.54 | 69.97 |
| AP | Yoon et al. [34] | 86.76 | 60.86 |
| AP | Nie et al. [21] | 90.07 | 71.29 |
| AP | Wang et al. [32] | **93.25** | 70.69 |
| AP | Bai et al. [1] | 92.97 | 82.81 |
| AP | **PuriFlow([21])** | 90.21 | **86.91** |

Table 1. Evaluation comparing adversarial training (AT) and purification (AP) methods against AutoAttack with $\ell_\infty$-perturbations $\epsilon = 8/255$ on CIFAR-10. AP methods are set to time $t' = 100$.

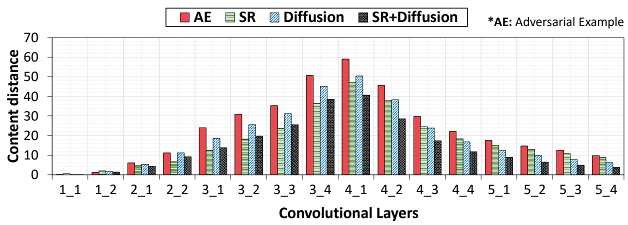
## 6. Supplemental Experiments

This section provides supplemental experimental results that are not surrounded in the manuscript.

**AutoAttack** Table 1 demonstrates that PuriFlow achieves robust accuracy gains of 16.46% for WRN-28-10 and 16.62% for WRN-70-16 over a diffusion-only approach on CIFAR-10. Furthermore, it surpasses guided-diffusion purification [1] by 4.29% and 4.1% for WRN-28-10 and WRN-70-16, respectively. As demonstrated in Table 9 of the manuscript, PuriFlow shows more efficiency, particularly considering the overhead noted in the study [1].




(a) Impact of SR compared to diffusion with increments of $t'$.




(b) Synergistic effect in feature restoration of SR combined with diffusion. Diffusion and SR+Diffusion use the same diffusion time $t' = 90$.

Figure 1. Content distance from the ground truth for each image type, measured across all convolutional layers of VGG-19. Measurements follow [10], using the same experimental settings as Figure 5 in the manuscript.

| Method | Type | Standard (%) | Robust (%) |
|---|---|---|---|
| Nearest-Neighbor | Interpolation | 70.23 | 66.57 |
| Bicubic | Interpolation | 72.41 | 68.10 |
| **Bilinear** | **Interpolation** | **72.97** | **68.75** |

Table 2. Evaluation of PuriFlow under a preprocessor-blind attack, a PGD attack targeting ResNet-50 within $\ell_\infty$-norm ball of radius $\epsilon = 4/255$ on the ImageNet-1k. PuriFlow uses $t' = 90$ and three different interpolation methods to down-scale SR images.

## 6.1. Visualization of Real-Valued Content Distance

This section visualizes the content distance values, as represented by the normalized illustration in Figure 4 of the manuscript. Figure 1 displays the content distances across all convolutional layers in VGG-19, following [10]. The distances vary significantly across layers, with initial layers (e.g., Conv1_1 and Conv1_2) showing negligibly small values. Figure 1a demonstrates that SR outperforms diffusion for varied times in early layers. Furthermore, Figure 1b shows that combining SR with diffusion synergistically reduces the content distance by compensating for diffusion's limited effectiveness in the early layers. This, in turn, enhances its impact in the later layers.

## 6.2. Additional Ablation Study

**Impact of down-scaling criteria to SR image.** In this study, we conduct a case study to assess the efficacy of different interpolation methods for downscaling SR images in the PuriFlow process. Our default choice, ESRGAN×2, generates the SR images. We evaluate common interpolation techniques, including Nearest-Neighbor and Bicubic, in addition to our default method, Bilinear. As shown in Table 2, the results demonstrate that Bilinear interpolation surpasses the other two methods in standard and robust accuracy. Figure 1 further reveals that all examined downscaling techniques contribute to decreasing the cross-entropy of adversarial examples before their diffusion processes. However, it is noteworthy that Nearest-Neighbor interpolation exhibits some inconsistencies in reducing cross-entropy in conjunction with the diffusion process.

| Method | SR model | Ratio | Standard (%) | Robust (%) |
|---|---|---|---|---|
| *WRN-70-16* | | | 95.79 | 0.00 |
| Nie et al. [21] | - | - | 90.07 | 71.29 |
| PuriFlow([21]) | MDSR | ×2 | 90.21 | 86.91 |
| PuriFlow([21]) | LDM-SR | ×4 | 87.21 | 86.32 |

Table 3. Evaluation of performance on CIFAR-10 using diffusion-based SR (LDM-SR [28]) integrated with diffusion, under a white-box attack, AutoAttack. All diffusion times are set to $t' = 100$, and LDM's time is set to 5. Note that LDM-SR is officially available as a ×4 model from Hugging-Face.

| Reverse time ($\tilde{t}'$) | Forward time $t' = 90$ | | | |
|---|---|---|---|---|
| | $\tilde{t}' = 50$ | $\tilde{t}' = 70$ | $\tilde{t}' = 90$ | $\tilde{t}' = 110$ |
| Standard (%) | 56.36% | 62.78% | **72.97%** | 66.72% |
| Robust (%) | 51.83% | 59.16% | **68.75%** | 63.75% |

Table 4. Evalution of PuriFlow with decoupling the forward and reverse diffusion times using the default diffusion model, VP-SDE. This analysis uses a PGD attack targeting ResNet-50 within an $\ell_\infty$-norm ball of a radius $\epsilon = 4/255$ on ImageNet-1k. According to varying reverse diffusion time $\tilde{t}'$, the noise schedule functions of $\boldsymbol{f}(\cdot, t)$ and $g(t)$ are adapted to the set of $\{\alpha_t\}_{t=1}^{\tilde{t}'}$.

**Impact of using diffusion-based SR.** In this study, we replace conventional SR models with a recent diffusion-based SR method for further evaluation, using AutoAttack on a WRN-70-16 classifier. Since LDM-SR is restricted to a ×4 resolution, it shows performance differences compared to the ×2 MDSR, which we identified as the optimal choice in this study. Purification with LDM-SR achieves a robust accuracy improvement of 15.03%, similar to existing SR. However, its standard accuracy is 3% lower than MDSR due to increased uncertainty from the higher resolution ratio. Nevertheless, the results confirm the effectiveness of diffusion-based purification in enhancing robust accuracy, even though we employ a different type of SR.

**Separating forward and reverse diffusion times in PuriFlow.** In our study, as outlined in Algorithm 1, PuriFlow is configured with an equal forward and reverse diffusion time $t'$, used during the diffusion of down-scaled SR images. Table 4 explores the performances when diverging the reverse

| Reverse interval ($d$) | Forward time $t' = 90$ | | | |
|---|---|---|---|---|
| | $d = 1$ | $d = 5$ | $d = 15$ | $d = 30$ |
| Standard (%) | **73.39%** | 54.04% | 53.25% | 52.53% |
| Robust (%) | **66.51%** | 49.60% | 48.89% | 48.24% |

Table 5. Evaluation of PuriFlow with varying intervals in the reverse process, using VE-SDE as the diffusion model. This study involves a PGD attack targeting ResNet-50 within an $\ell_\infty$-norm ball of radius $\epsilon = 4/255$ on ImageNet-1k. According to different interval values $d$, the influenced repetition frequency of the reverse noise schedule function $\boldsymbol{f}_{rev}(\cdot, t)$ is determined as $t'/d$.

diffusion time $\tilde{t}' \in \{50, 70, 110\}$ from the set forward diffusion time of $t' = 90$, contrasting it with our standard approach of a consistent reverse diffusion time $\tilde{t}' = t' = 90$. When $\tilde{t}' = t' = 90$, both forward and reverse diffusion processes follow the same noise schedule defined by $\{\alpha_t\}_{t=1}^{90}$. In scenarios with different $\tilde{t}'$ values, the reverse diffusion employs $\{\alpha_t\}_{t=1}^{50}$, $\{\alpha_t\}_{t=1}^{70}$, and $\{\alpha_t\}_{t=1}^{110}$, while maintaining the forward diffusion noise schedule at $\{\alpha_t\}_{t=1}^{90}$. Our findings reveal that aligning the forward and reverse diffusion times ($\tilde{t}' = t' = 90$) offers superior standard and robust accuracy compared to shorter reverse diffusion times ($\tilde{t}'$ of 50 and 70). Additionally, this aligned approach surpasses the performance of a longer reverse diffusion time ($\tilde{t}'$ of 110) in both standard and robust accuracy metrics by 6.25% and 5.00%, respectively. This result demonstrates the effectiveness of maintaining consistent diffusion times $t' = \tilde{t}'$ in both forward and reverse processes to maximize purification effects with PuriFlow.

**Adopting intervals in reverse diffusion time of VE-SDE.** As detailed in Table 6, our manuscript assesses VE-SDE, a model shifting VP-SDE through non-Markovian diffusion processes. These non-Markovian processes enable deterministic generative models capable of producing samples more rapidly. This speed enhancement is achieved by adjusting the number of reverse diffusion steps based on a time interval $d$. Table 5 demonstrates the results when employing different interval values $d \in \{5, 15, 30\}$, compared to our default choice of $d = 1$. With $d = 1$, the reverse diffusion process is conducted in the same steps as the forward diffusion time $t' = 90$. For other values of $d$, reverse diffusion occurs 16, 6, and 3 times, respectively, aligned with $t'/d$. The observations indicate that our selected interval $d = 1$ shows superior standard and robust accuracy performances over other interval utilization. Remarkably, as the interval $d$ increases, there is a noticeable, gradual decrease in performance across both metrics. This result suggests the advantage of using VE-SDE without intervals in purification, emphasizing its preserving accuracy.

**Impact of diffusion time $t'$.** Figure 2 illustrates how PuriFlow's standard and robust accuracy and wall time vary

| | ESRGAN×2 | | | ESRGAN×4 | | |
|---|---|---|---|---|---|---|
| Diffusion time ($t'$) | 10 | 30 | 50 | 10 | 30 | 50 |
| Standard (%) | 74.00 | 74.01 | 73.68 | 73.46 | 73.28 | 73.05 |
| Robust (%) | 49.92 | 61.42 | 65.68 | 46.41 | 59.83 | 64.57 |

Table 6. Effect of upscaling ratio and diffusion time ($t'$) on standard and robust accuracy, following the settings in Table 6.

| SR model | MDSR | ESRGAN |
|---|---|---|
| BPDA+EOT (`s/it.`) | 4.85 | 4.96 |
| PGD+EOT (`s/it.`) | 32.45 | 32.95 |
| AutoAttack (`s/it.`) | 433.71 | 444.73 |
| DiffAttack (`s/it.`) | 451.30 | 460.40 |

Table 7. Time required for an iteration of white-box attacks for a single image, using two SR models, VP-SDE [21] ($t' = 100$), and WRN-70-16 on CIFAR-10.

with diffusion time $t'$. Figure 2a shows the impact of robust accuracy peaks at $t' = 300$ and significantly drops at $t' = 150$ when SR is equipped with a one-shot denoisor. Using ResNet-50 and DeiT-S, PuriFlow, which integrates SR with solving SDEs, significantly improves robust accuracy up to $t' = 110$, then plateaus at $t' = 130$ and $t' = 150$, while standard accuracy decreases slightly by about 3% as $t'$ increases. Despite this, PuriFlow at $t' = 150$ maintains higher standard accuracy (70.96% for ResNet-50 and 74.93% for DeiT-S) compared to Nie et al. [21] (67.79% and 73.63%), indicating the effectiveness of SR preprocessing in mitigating negative effects of extended diffusion times. Furthermore, wall time (*s/img*) increases proportionally with $t'$, irrespective of the classifier used. Efficient processing times of approximately 1.37 seconds per image at $t' = 10$ for preprocessor-blind attacks and 16 seconds per image at $t' = 130$ for AutoAttack were achieved, which is relatively efficient compared to 19 seconds per image at $t' = 150$ for Nie et al. [21]. *Note that this purification overhead originates from the diffusion process, as shown in Table 10 of the manuscript.*

**Diffusion time and SR factors** Table 6 shows that while SR is effective, its synergy with diffusion seems hindered as the SR ratio increases. This is possibly due to an increase in unknown pixels for the inverse problem, which may lead to changes in information at a higher SR ratio.

### 6.3. White-Box Attack Overhead on PuriFlow

Table 7 presents the time consumed per attack iteration when executing white-box attacks on the entire PuriFlow pipeline. The overhead is primarily accumulated from the dependency on the number of diffusion times $t'$ per attack iteration. BPDA+EOT and PGD+EOT exhibit relatively shorter attack times than AutoAttack and DiffAttack, as

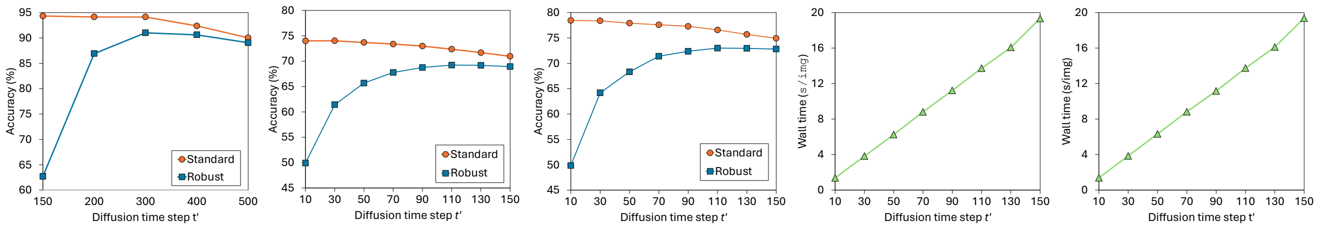



Figure 2. **(a)** Impact of diffusion time $t'$ in SR+iOSD under a white-box attack, BPDA+EOT, within $\ell_\infty$-norm ball of radius $\epsilon = 8/255$ on CIFAR-10, and the same impact in SR+Diffusion under a preprocessor-blind attack. Accuracy and wall time for PGD attacks targeting **(b),(d)** ResNet-50 and **(c),(e)** DeiT-S within $\ell_\infty$-norm ball of radius $\epsilon = 4/255$ on the ImageNet-1k.

| Method | OTS? | Certified accuracy at radius $\epsilon$ | | | | |
|---|---|---|---|---|---|---|
| | | 0.5 | 1.0 | 1.5 | 2.0 | 3.0 |
| RS [6] | ✗ | 49.0 | 37.0 | 29.0 | 19.0 | 12.0 |
| SmoothAdv [29] | ✗ | 56.0 | 43.0 | 37.0 | 27.0 | 20.0 |
| Consistency [14] | ✗ | 50.0 | 44.0 | 34.0 | 24.0 | 17.0 |
| MACER [35] | ✗ | 57.0 | 43.0 | 31.0 | 25.0 | 14.0 |
| Boosting [13] | ✗ | 57.0 | 44.6 | 38.4 | 28.6 | 21.2 |
| SmoothMix [15] | ✗ | 50.0 | 43.0 | 38.0 | 26.0 | 20.0 |
| Lee [18] | ✓ | 41.0 | 24.0 | 11.0 | - | - |
| DDS [4] | ✓ | 71.1 | 54.3 | 38.1 | 29.5 | 13.1 |
| **PuriFlow ([4])** | ✓ | **72.1** | **59.7** | **45.0** | **33.6** | **21.3** |

Table 8. Comparative evaluation of various models on ImageNet-1k, based on the results by DDS[4]. The "OTS?" indicates whether solely off-the-shelf models were used in the defense methods. Symbols ✗ and ✓ distinguish between the techniques that develop custom-develop models for randomized smoothing defenses and those using pre-trained score functions for one-shot denoising. Each method was tested under three different noise levels $\sigma \in \{0.25, 0.5, 1.0\}$, within an $\ell_2$-norm ball of radius $\epsilon$. The best accuracies reported for each noise level are compared.

| Noise level | Certified accuracy at radius $\epsilon$ | | | | | |
|---|---|---|---|---|---|---|
| | 0.0 | 0.5 | 1.0 | 1.5 | 2.0 | 3.0 |
| $\sigma = 0.25$ | **80.5** | **72.1** | 00.0 | 00.0 | 00.0 | 00.0 |
| $\sigma = 0.50$ | 76.8 | 69.0 | **59.7** | **45.0** | 00.0 | 00.0 |
| $\sigma = 1.00$ | 63.6 | 56.1 | 49.0 | 42.0 | **33.6** | **21.3** |

Table 9. Certified accuracy on PuriFlow on the ImageNet-1k dataset in three noise levels $\sigma$, within $\ell_2$-norm ball of radius $\epsilon \in \{0.0, 0.5, 1.0, 1.5, 2.0, 3.0\}$, followed by Carlini et al. [4].

the latter ensemble multiple attacks increase their execution time. While BPDA+EOT and PGD+EOT are highly efficient and effective attack methods from the attackers' perspective, PuriFlow demonstrates strong test-time defense capabilities by integrating efficient SR models. Notably, the effectiveness of PuriFlow in defending against AutoAttack and DiffAttack, which have significant overheads, emphasizes the dramatic efficiency of its SR-based approach.

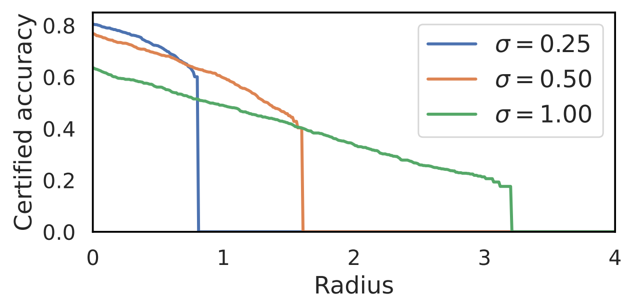


Figure 3. Certified accuracy of PuriFlow against randomized perturbations within an $\ell_2$-norm ball, considering three different levels of Gaussian noise ($\sigma$). The bounds are calculated using a sample of 1,000 images from the ImageNet-1k dataset.

## 6.4. Certified Robustness

One direction to obtain a certified model that provably resists adversarial attacks is to develop Gaussian smoothed models [6]. These models denoted as $\hat{\mathcal{F}}\phi(\mathbf{x})$ are designed to be robust against noise-corrupted images $\hat{\mathcal{F}}_\phi(\mathbf{x}) = \mathbb{E}_\delta[\mathcal{F}_\phi(\mathbf{x} + \delta)]$ where $\delta \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ and $\delta \in \mathbb{R}^d$. PuriFlow, which requires no additional training, seamlessly integrates into a dual-phase framework, consists of an off-the-shelf model that purifies noise-corrupted images $\mathbf{x}+\delta$ and a pre-trained classifier for predictions of the denoised images.

Table 8 demonstrates the significance of PuriFlow, as summarized in Algorithm 2 in constructing a Gaussian smoothed model, utilizing a pre-trained BEiT-L as the classifier in conjunction with a single step OSD [4]. PuriFlow achieves a promising level of certified robustness by leveraging solely off-the-shelf components for super-resolution, diffusion (one-shot denoiser), and classification. To ensure a fair comparison with DDS [4], our evaluation adheres to the experimental protocols specified by DDS for selecting diffusion time $t'$, with $\sigma^2$ calculated as $\sigma^2 = (1 - \alpha_{t'})/\alpha_{t'}$ and employing the 1k images, an image per class from the ImageNet-1k validation set.

Our model, which combines BEiT-L and SR-integrated

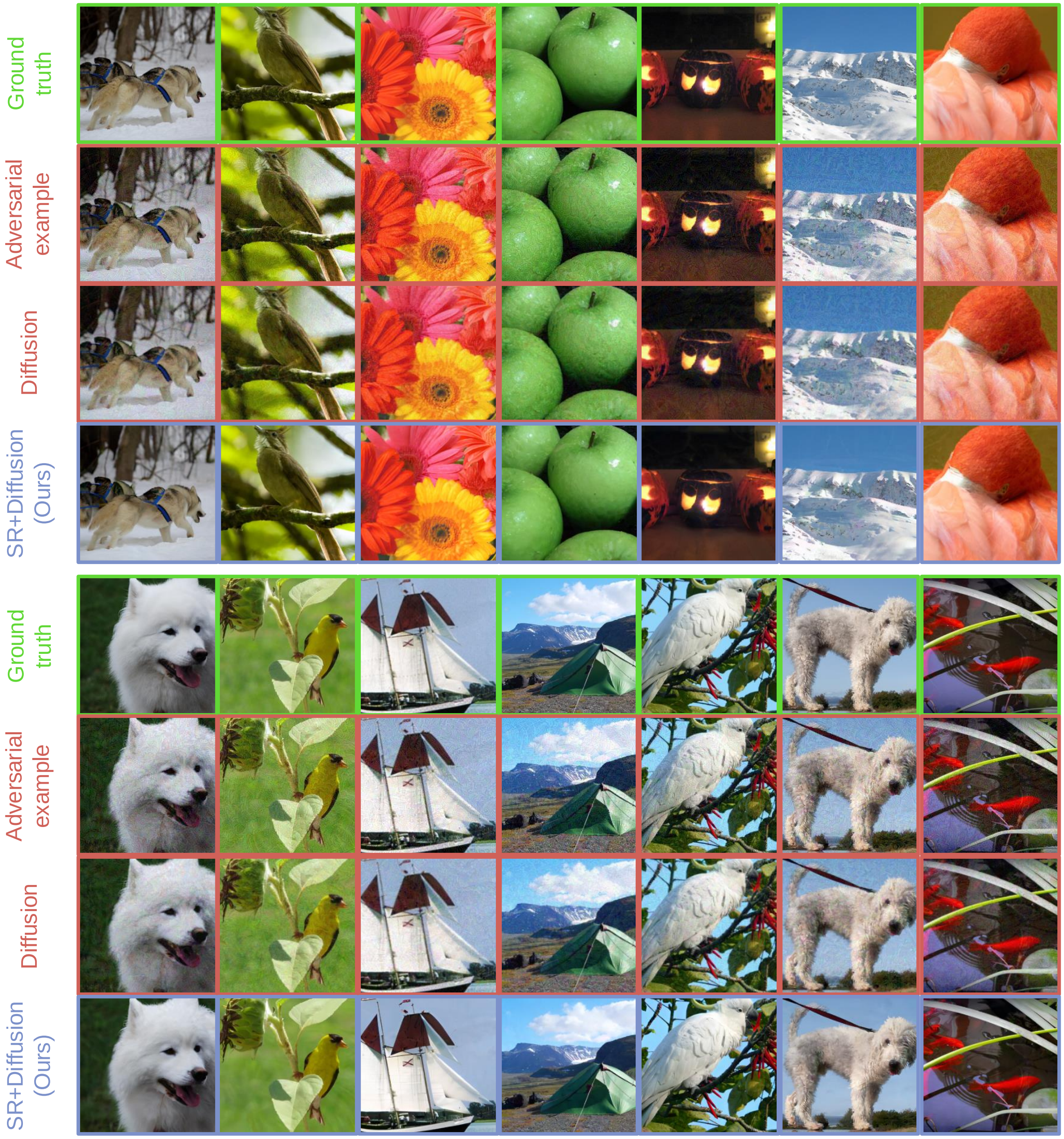



Figure 4. Visualization of adversarial example purification using PGD attack on ResNet-50 with $\ell_\infty$-norm ball radius $\epsilon = 16/255$ for the ImageNet-1k dataset. Original images are framed in green. Images within red boxes indicate incorrect predictions, while those in blue boxes show correct classifications. The diffusion time is set to $t' = 30$ for both Diffusion [21] and SR+Diffusion (Ours).

OSD, demonstrates its effectiveness as evidenced in Figure 3 and Table 9. Table 9 details the certified Top-1 accuracy at each specified radius, while Figure 3 illustrates the variations in certified accuracy to different sigma values as the radius changes. We evaluate the robustness of this model using randomized smoothed images across a range from $\epsilon = 0.5$ to $\epsilon = 3.0$, with $\ell_2$-perturbations.

## 7. Visualization of Purified Adversarial Image

This section presents a case study that qualitatively assesses the efficacy of our purification process on adversarial examples. Through the visualization in Figure 4, we demonstrate the distinct advantages of PuriFlow over Nie et al. [21]. Our observations reveal that Nie et al. [21] struggles to effectively remove adversarial noise when using a diffusion time of $t' = 30$ under a strong adversarial attack. In stark contrast, PuriFlow, even with the same diffusion time of $t' = 30$, demonstrates remarkable success in denoising and restoring images to a state similar to their original form, leading to accurate predictions. We conjecture that content restoration contributes to stable convergence during the denoising steps in the diffusion process.

## 8. Limitations

Our approach uses SR models to restore and align adversarial examples with their ground truth in diffusion-based purification. However, this synergy does not fully mitigate the overhead associated with diffusion time, which remains a dominant factor in the purification process. Enhancing this aspect could lead to significant improvements in robustness and practical applicability. In future work, we will focus on accelerating diffusion techniques [17] and integrating SR models to develop a more effective and efficient test-time defense.

## 9. Societal Impact

Adversarial purification can significantly enhance the security of DNNs in safety-critical systems, such as autonomous vehicles, healthcare diagnostics [2], and security surveillance. By effectively neutralizing adversarial attacks, these systems become more reliable and trustworthy. With improved defenses against adversarial attacks, the trustworthiness of AI systems increases in various sectors [23–26]. This can lead to broader acceptance and integration of AI technologies in everyday life. Nevertheless, while adversarial purification strengthens defenses, the underlying knowledge could be used maliciously. By understanding how purification methods work, adversaries might develop more sophisticated attacks. That is, smaller organizations or entities with fewer resources may not have access to advanced adversarial purification techniques, leading to a disparity in the security and reliability of AI systems. In conclusion, while adversarial purification presents a significant step forward in enhancing the security and reliability of AI systems, it also presents challenges that require careful consideration and responsible handling to ensure its overwhelmingly positive impact on society.

## References


[1] Mingyuan Bai, Wei Huang, Tenghui Li, Andong Wang, Junbin Gao, Cesar F Caiafa, and Qibin Zhao. Diffusion models demand contrastive guidance for adversarial purification to advance. In *International Conference on Machine Learning*, 2024. 3

[2] Yeong Hak Bang, Yoon Ho Choi, Mincheol Park, Soo-Yong Shin, and Seok Jin Kim. Clinical relevance of deep learning models in predicting the onset timing of cancer pain exacerbation. *Scientific Reports*, 2023. 8

[3] Kartikeya Bhardwaj, Dibakar Gope, James Ward, Paul Whatmough, and Danny Loh. Super-efficient super resolution for fast adversarial defense at the edge. In *2022 Design, Automation & Test in Europe Conference & Exhibition*, 2022. 3

[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*, 2023. 3, 6

[5] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 2022. 1

[6] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, 2019. 6

[7] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, 2020. 2

[8] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 2

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009. 3

[10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 4

[11] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020. 3

[12] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 2021. 3

[13] Miklos Z. Horvath, Mark Niklas Mueller, Marc Fischer, and Martin Vechev. Boosting randomized smoothing with variance reduced classifiers. In *International Conference on Learning Representations*, 2022. 6

[14] Jongheon Jeong and Jinwoo Shin. Consistency regularization for certified robustness of smoothed classifiers. *Advances in Neural Information Processing Systems*, 2020. 6

[15] Jongheon Jeong, Sejun Park, Minkyu Kim, Heung-Chang Lee, Do-Guk Kim, and Jinwoo Shin. Smoothmix: Training confidence-calibrated smoothed classifiers for certified robustness. *Advances in Neural Information Processing Systems*, 2021. 6

[16] Mintong Kang, Dawn Song, and Bo Li. Diffattack: Evasion attacks against diffusion-based adversarial purification. *Advances in Neural Information Processing Systems*, 2024. 2

[17] Sungbin Kim, Hyunwuk Lee, Wonho Cho, Mincheol Park, and Won Woo Ro. Ditto: Accelerating diffusion model via temporal value similarity. In *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2025. 8

[18] Kyungmin Lee. Provable defense by denoised smoothing with learned score function. In *ICLR Workshop on Security and Safety in Machine Learning Systems*, 2021. 6

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

[20] Aamir Mustafa, Salman H. Khan, Munawar Hayat, Jianbing Shen, and Ling Shao. Image super-resolution as a defense against adversarial attacks. *IEEE Transactions on Image Processing*, 2020. 3

[21] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning*, 2022. 2, 3, 4, 5, 7, 8

[22] Tianyu Pang, Min Lin, Xiao Yang, Jun Zhu, and Shuicheng Yan. Robustness and accuracy could be reconcilable by (proper) definition. In *International Conference on Machine Learning*, 2022. 3

[23] Cheonjun Park, Mincheol Park, Hyun Jae Oh, Minkyu Kim, Myung Kuk Yoon, Suhyun Kim, and Won Woo Ro. Balanced column-wise block pruning for maximizing gpu parallelism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. 8

[24] Cheonjun Park, Mincheol Park, Hyunchan Moon, Myung Kuk Yoon, Seokjin Go, Suhyun Kim, and Won Woo Ro. Deprune: Depth-wise separable convolution pruning for maximizing gpu parallelism. *Advances in Neural Information Processing Systems*, 2024.

[25] Mincheol Park, Dongjin Kim, Cheonjun Park, Yuna Park, Gyeong Eun Gong, Won Woo Ro, and Suhyun Kim. Reprune: Channel pruning via kernel representative selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[26] Mincheol Park, Woojeong Kim, Junsik Bang, Yuna Park, Won Woo Ro, and Suhyun Kim. Perspective shifts: Cultivating teacher diversity in online knowledge distillation. *Knowledge-Based Systems*, 2025. 8

[27] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021. 3

[28] 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*, 2022. 4

[29] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in Neural Information Processing Systems*, 2019. 6

[30] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised MAP inference for image super-resolution. In *International Conference on Learning Representations*, 2017. 1

[31] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2

[32] 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*, 2023. 3

[33] Boxi Wu, Heng Pan, Li Shen, Jindong Gu, Shuai Zhao, Zhifeng Li, Deng Cai, Xiaofei He, and Wei Liu. Attacking adversarial attacks as a defense. *arXiv preprint arXiv:2106.04938*, 2021. 3

[34] Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models. In *International Conference on Machine Learning*, 2021. 3

[35] Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. In *International Conference on Learning Representations*, 2020. 6

[36] Boya Zhang, Weijian Luo, and Zhihua Zhang. Enhancing adversarial robustness via score-based optimization. *Advances in Neural Information Processing Systems*, 2023. 2, 3