Robust Evaluation of Diffusion-Based Adversarial Purification

Minjong Lee  
CSED POSTECH  
minjong.lee@postech.ac.kr

Dongwoo Kim  
CSED, GSAI POSTECH  
dongwoo.kim@postech.ac.kr

Abstract

We question the current evaluation practice on diffusion-based purification methods. Diffusion-based purification methods aim to remove adversarial effects from an input data point at test time. The approach gains increasing attention as an alternative to adversarial training due to the disentangling between training and testing. Well-known white-box attacks are often employed to measure the robustness of the purification. However, it is unknown whether these attacks are the most effective for the diffusion-based purification since the attacks are often tailored for adversarial training. We analyze the current practices and provide a new guideline for measuring the robustness of purification methods against adversarial attacks. Based on our analysis, we further propose a new purification strategy improving robustness compared to the current diffusion-based purification methods.

1. Introduction

Adversarial attacks [21, 6] can cause deep neural networks (DNNs) to produce incorrect outputs by adding imperceptible perturbations to inputs. While various adversarial defenses have been proposed, adversarial training [41, 13] has shown promising results in building robust DNNs. Since adversarial training feeds the model both normal and adversarial examples during training time, one needs to pre-determine which attack method is used to generate the adversarial examples. On the other hand, adaptive test-time defense [19, 16] has recently gained increasing attention since it adaptively removes the adversarial effect at test time without adversarial training. Adversarial purification [31, 24], one of the adaptive test-time defenses, uses generative models to restore the clean examples from the adversarial examples.

Diffusion-based generative models [17, 35] has been suggested as a potential solution for adversarial purification [24, 36]. Diffusion models learn transformations from data distributions to well-known simple distributions such as the Gaussian and vice versa through forward and reverse processes, respectively. When applied to the purification, the forward process gradually adds noise to the input, and the reverse process gradually removes the noises to uncover the original image without imperceptible adversarial noise. With a theoretical guarantee, the recent success of DiffPure [24] against many adversarial training methods shows the potential of using diffusion processes for improving the robustness against adversarial attacks.

Evaluating the robustness of adaptive test-time defenses is, however, known to be difficult due to their complex defense algorithms and properties. Croce et al. [8] shows that finding worse-case perturbation is important to measure the robustness of the defenses. Randomness and iterative calls of adaptive test-time defenses, however, make their gradients obfuscated. Therefore, the gradient-based attack methods [21, 6] might be inappropriate for measuring the robustness under such obfuscation. New algorithms, such as Backward Pass Differentiable Approximation (BPDA) [3], and additional recommendations [8] have been proposed to evaluate their robustness accurately. However, it is unclear whether these algorithms and recommendations can still be used to evaluate diffusion-based purification.

In the first part of this work, we analyze the existing evaluation methods for diffusion-based purification. We find that the adjoint method, often used to compute a full gradient of the iterative process, relies on the performance of an underlying numerical solver. Tailored to the diffusion models, we propose a surrogate process, an alternative method to approximate the gradient from the iterative procedure and show the strong robustness in recent work can be weaker than claimed with the surrogate process. We then compare two gradient-based attack methods, AutoAttack [6] and PGD [21], with the surrogate process and find that PGD is more effective for the diffusion-based purification. To this end, we propose a practical recommendation to evaluate the robustness of diffusion-based purification.

In the second part of this work, we analyze the importance of the hyperparameter for successive defenses with purification. The diffusion models are trained without adversarial examples. Thus, proper validation of hyperparameters is impossible in general. Instead, we empirically an-
alize the influence of different hyperparameter selections from the attacker’s and defender’s perspectives. Based on our analysis, we propose a gradual noise-scheduling for multi-step purification. We show that our defense strategy highly improves robustness compared to the current diffusion-based purification methods under our proposed evaluation scheme.

We summarize our contributions as follows:

• We analyze the current evaluation of diffusion-based purification and provide a recommendation for robust evaluation.
• We investigate the influence of hyperparameters on the robustness of diffusion-based purification.
• We propose a gradual noise-scheduling strategy for diffusion-based purification, improving the robustness of diffusion-based purification.

2. Preliminary

We provide the background on the adversarial attack, diffusion models, and adversarial purification in this section.

2.1. Adversarial Attacks

Adversarial attacks aim to manipulate or trick machine learning models by adding imperceptible perturbations to input data that can cause the model to misclassify or produce incorrect outputs. The adversarial attacks can be categorized into black-box, grey-box, and white-box attacks. The black-box attack assumes that the attacker knows nothing about the internal structure of the classifier and defender. The white-box attack assumes that the attacker can obtain any information about the defender and the target classifier, including the architecture and parameter weights. The grey-box lies between the white- and black-box attacks, where the attacker partially knows the target model. In this work, we only focus on the performance of purification in the white-box attack since the white-box attack is the most difficult to defend from the defender’s perspective.

The Projected Gradient Descent (PGD) [21] method is a common white-box attack. PGD is a gradient-based attack that iteratively updates an adversarial example using the following rule

$$x_{t+1} = \Pi_{\mathcal{X}}(x_t + \alpha_t \text{sign}(\nabla_x \mathcal{L}(f_\theta(x_t), y)|_{x=x_t})),$$

where \(f_\theta\) represents a classifier, and \(\Pi_{\mathcal{X}}\) indicates a projection operation onto \(\mathcal{X}\). PGD can only be applied for the differentiable defense methods. For non-differentiable defense methods, the Backward Pass Differentiable Approximation (BPDA) [3] is widely used, which computes the gradient of the non-differentiable function by using a differentiable approximation. Expectation over Transformation (EOT) [2] can be additionally employed for randomized defenses, which optimizes the expectation of the randomness. AutoAttack [6] is an ensemble of four different types of attacks. In this work, we measure the robustness of purification methods against these attack methods.

2.2. Diffusion Models

Recently, diffusion-based models [17, 35] have gained increasing attention in generative models. Unlike the VAEs and GANs, the diffusion-based models produce samples by gradually removing noise from random noise. The training of diffusion-based models consists of two processes, the forward process, and the reverse denoising process. The forward process adds Gaussian noise over \(T\) steps to the observed input \(x_0\) with a predefined variance scheduler \(\beta_t\), whose joint distribution is defined as

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}),$$

where \(q(x_t|x_{t-1})\) is a Gaussian transition kernel from \(x_{t-1}\) to \(x_t\)

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I).$$

The reverse process denoises the random noise \(x_T\) over \(T\) times, whose joint distribution is defined as

$$p_\theta(x_0:T) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t).$$

The transition distribution from \(x_t\) to \(x_{t-1}\) is often modeled by Gaussian distribution

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma^2_t I),$$

where \(\sigma_t\) is a variance, and \(\mu_\theta\) is a predicted mean of \(x_{t-1}\) derived from a learnable denoising model \(\epsilon_\theta\). The denoising model is often trained by predicting a random noise at each time step via following objective

$$L(\theta) = \mathbb{E}_{t,x_0,\epsilon} \left[ ||\epsilon - \epsilon_\theta(x_t, t)||^2 \right],$$

where \(\epsilon\) is a Gaussian noise, i.e., \(\epsilon \sim \mathcal{N}(0, I)\). The model \(\epsilon_\theta\) takes the noisy input \(x_t\) and the time step \(t\) to predict the actual noise \(\epsilon\) at time \(t\). In the Denoising Diffusion Probabilistic Model (DDPM) [17], the reverse denoising process is performed over \(T\) steps through random sampling, resulting in a slower generation of samples compared with GANs and VAEs.

Based on the fact that the multiple denoising steps can be performed at a single step via a non-Markovian process, Song et al. [33] proposes a new sampling strategy, which we call Denoising Diffusion Implicit Model (DDIM) sampler, to accelerate the reverse denoising process. In this work, we compare the performances of DDPM and DDIM samplers in the diffusion-based purification approach.
2.3. Adversarial Purification

Adversarial purification via generative models is a technique used to improve the robustness of machine learning models against adversarial attacks [31]. The idea behind this technique is to use a generative model to learn the underlying distribution of the clean data and use it to purify the adversarial examples.

Diffusion-based generative models can be used as a purification process if we assume that the imperceptible adversarial signals as noise [24]. To do so, the purification process adds noise to the adversarial example via the forward process with $t^*$ steps, and it removes noises via the denoising process. The choice of the number of forward steps $t^*$ is essential since too much noise can remove the semantic information of the original example, or too little noise cannot remove adversarial perturbation. In theory, as we add more noise to the adversarial example, the distributions over the noisy adversarial example and the true example become close to each other [24]. Therefore, the denoised examples are likely to be similar.

3. Evaluation for Diffusion-Based Purification

In this section, we first review the current practices in evaluating diffusion-based purification methods. We then curate three research questions to address their potential limitations and provide our answers to these questions through empirical evaluations.


Evaluation of the diffusion-based purification against gradient-based white-box attacks is non-trivial due to many function calls on the denoising process. Multiple function calls in the denoising step often require an impractical amount of memory, making it unfeasible to compute the gradient of the full defense process. Because of this problem, most defenses [39, 36, 16] consider BPDA the strongest adaptive white-box attack in the current practice since it does not rely on the gradients of defense methods. However, the vulnerability of diffusion-based purification on white-box attacks has yet to be fully identified. The importance of testing in adaptive white-box attacks of purification has been recognized only recently by the work of DiffPure [24].

DiffPure calculates the full gradients of their defense process using an adjoint method. The adjoint method is employed to avoid the extensive use of memory while obtaining the full gradient. DiffPure is evaluated on AutoAttack, a de facto evaluation method in adversarial training. Although their evaluation framework is more robust than the previous work, the design choices of their evaluation still raise questions since 1) the adjoint method relies on the performance of an underlying numerical solver [42], and 2) there is no comprehensive comparison between different attacks using the full gradient.

Based on our observation, we carefully curate the following three research questions to address the robustness of the current evaluation framework in diffusion-based purification:

- **RQ1.** Is the adjoint method the best way to generate adversarial examples with full gradients? Is there any alternative to the adjoint method?
- **RQ2.** Is AutoAttack still better than the other gradient-based attacks, such as PGD, when the alternative is available?
- **RQ3.** Is BPDA still more effective than the best combination of full-gradient attacks?

In the next section, we re-evaluate the existing purification methods to answer these questions.

3.2. Experimental Results & Analysis

We evaluate the performance of three diffusion-based purification methods: ADP$^1$ [39], DiffPure [24], GDMP [36]. We additionally evaluate two non-diffusion-based adaptive test-time defenses: SODEF [19], and DISCO [16] to address whether our findings still hold for the non-diffusion-based purification methods. We evaluate their robustness on CIFAR-10 against three gradient-based attacks, including PGD, BPDA, and AutoAttack, with a maximum attack strength of $\ell_\infty(\epsilon = 8/255)$. A comprehensive description of evaluation configurations is provided in Appendix A.

**Surrogate process and its gradient.** The adjoint method can compute the exact gradient in theory, but in practice, the adjoint relies on the performance of the numerical solver, whose performance becomes problematic in some cases as reported by Zhuang et al. [42]. To answer the RQ1, we compare the adjoint method against the full gradient obtained from back-propagation if possible, and if not due to the memory issue, we use the approximated gradient obtained from a surrogate process. The surrogate process utilizes the fact that given the total amount of noise, we can denoise the same amount of noise with different numbers of denoising steps [33]. Therefore, instead of using the entire denoising steps, we can mimic the original denoising process with fewer function calls, whose gradients can be obtained by back-propagating the forward and denosing process directly.

The gradients obtained from the surrogate process differ from the exact gradients. However, if the accumulated

---

$^1$Although ADP uses a score-based model and Langevin dynamics, since the concept is similar to the diffusion model, we consider ADP diffusion-based purification.
denoising steps can be approximated with fewer denoising steps, we can use the approximated gradients as a proxy of the exact gradients. The surrogate process can also relax the randomness occurring in multiple denoising steps.

**RQ1: Is the adjoint method the best way to generate adversarial examples with full gradients?**  We compare the adjoint method with the full gradient obtained from direct back-propagation of the defense process with the original or surrogate processes.

Table 1 shows that the robust accuracy of DiffPure [24] with the direct back-propagation is 46.84% on PGD+EOT attack, which is 27.54% lower than the reported accuracy with the adjoint method. The results show that direct back-propagation is more effective than the adjoint method. Furthermore, we use a surrogate process for GDMP [36], and the robust accuracy is 24.06%, which is 51.53% lower than the reported accuracy against the BPDA attack. It can be concluded that, in cases where the gradients of the defense process are unavailable to calculate, the surrogate process can be an alternative to generate adversarial examples.

SODEF [19], a non-diffusion-based purification, originally uses the adjoint method to generate adversarial examples. We evaluate SODEF with direct back-propagation for the attack and observe 49.28% robust accuracy against AutoAttack, which is lower than 53.69% of the underlying model without defense. This result suggests that the use of a numerical solver would not be effective for the adversarial attack.

**RQ2: Is AutoAttack still better than the other gradient-based attacks, such as PGD, when the alternative is available?**  AutoAttack has recently been used as a standard method to evaluate defenses due to its robustness against defenses. Although AutoAttack may not be an ideal choice for randomized defenses\(^2\), still many purification methods, such as DiffPure, rely on AutoAttack. However, as shown in Table 2, AutoAttack has a lower success rate than PGD+EOT against diffusion-based purification methods. For the \(\ell_{\infty}\) threat model (\(\epsilon = 8/255\)), PGD+EOT shows 16.76% and 26.05% more attack success rate than AutoAttack against DiffPure and ADP, respectively. We observe a similar result with the \(\ell_2\) threat model (\(\epsilon = 0.5\)). Therefore, evaluation with PGD+EOT for diffusion-based purification can be useful to evaluate their robustness. Further results of the difference between PGD+EOT against DiffPure with additional settings can be found in Appendix B.

**RQ3: Is BPDA still more effective than the best combination of full-gradient attacks?**  BPDA [3] has been widely used to evaluate defenses that can cause gradient obfuscation. Because multiple function calls can cause gradient obfuscation, ADP, GDMP, and DISCO have been evaluated on BPDA as the strongest adaptive white-box attack. However, our evaluation shows that BPDA has a

---

\(^2\)https://github.com/fra31/auto-attack/blob/master/flags_doc.md
lower attack success rate than the attacks using direct gradients of the defense process, as shown in Table 3. Against PGD+EOT using direct gradients of defense process, ADP and GDMP show robust accuracy of 33.48% and 24.06%, respectively, significantly lower than the reported accuracy with BPDA [39, 36]. DISCO even has 0% robust accuracy. From the results, we suggest that the direct gradients of the defense process need to be tested to check the robustness.

**Recommendation.** We propose an overall guideline for evaluating diffusion-based purifications as follows. We recommend using PGD+EOT rather than AutoAttack. When calculating gradients, it is best to directly back-propagate the full defense process. If this is unavailable due to memory constraints, using the surrogate process rather than the adjoint method is recommended. Note that our recommendation generally follows the suggestions made by Croce et al. [8] but is more tailored for the diffusion-based purification.

### 4. Analysis of Hyperparameters

The performance of diffusion-based purification is significantly influenced by varying hyperparameter configurations. In this section, we explore the importance of hyperparameters in defense processes.

#### 4.1. Experimental Settings

Understanding the importance of hyperparameters can help build a better defense mechanism. We investigate the effect of various hyperparameters of diffusion-based purification methods to determine the most robust configuration for the adaptive attack. Specifically, the following three hyperparameters are evaluated:

1. The number of forward steps.
2. The number of denoising steps.
3. The number of purification steps.

In addition, we re-evaluate the efficiency of several techniques proposed in previous works under our defense scheme.

We evaluate the purification against PGD+EOT on CIFAR-10. We provide the additional results on CIFAR-10 and ImageNet in Appendix C. Although we do not report ImageNet results in the main text, the overall findings are similar to those from CIFAR-10. We use a naturally pretrained WideResNet-28-10 [40] as an underlying classifier provided by Robustbench [7]. For a diffusion model, we use pretrained DDPM++ [35]. The variances for the diffusion model are linearly increasing from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$ when $T = 1000$ [17]. We use two different denoising models: DDPM [17] and DDIM [35].

For all experiments, we report the mean and standard deviation over five runs to measure the standard and robust accuracy. PGD uses 200 update iterations. 20 samples are used to compute EOT. Following the settings in DiffPure [24], we use a fixed subset of 512 randomly sampled images. To calculate gradients, we use direct gradients of the entire process. If impossible, we compute the approximated gradients from a surrogate process. In each experiment, we explain the defense process and the surrogate process in more detail.

#### 4.2. The Number of Forward Steps

We explore the effect of forward noising steps on robustness by varying the number of forward steps from 30 to 300, resulting in the changes of total variance ranged from 0.012 to 0.606. The same number of forward steps are used for both attack and defense, and we set five denoising steps for both attack and defense for all experiments.

As shown in Figure 1, the standard accuracy continuously decreases as the number of forward steps increases since more forward steps induce more noise. The robust accuracy increases first and decreases after 200 forward steps, i.e., $t^* = 200$. When the number of forward steps is small, the DDPM is more robust than the DDIM. However, DDIM shows better accuracy for both standard and robust than DDPM after 200 forward steps.

#### 4.3. The Number of Denoising Steps

Defenders may use fewer denoising steps to accelerate the defense process. From the other perspective, attackers may want to use fewer denoising steps than those used in the defense due to memory constraints. We explore the influence of the number of denoising steps through the following three experimental settings:

(a) The number of denoising steps in attack is set to five, and the number of denoising steps in defense is ranged from one to the maximum number of denoising steps.

(b) The number of denoising steps in both the attack and defense are the same, ranging from one to 20.

**Figure 1:** Standard and robust accuracy as we change the number of forward steps against PGD+EOT $\ell_\infty(\epsilon = 8/255)$ on CIFAR-10. Five denoising steps for both attack and defense are used.
The number of denoising steps in defense is set to the maximum number of denoising steps, and the number of denoising steps in attack is ranged from one to 20.\(^3\)

The results are displayed in Figure 2. From the defense perspective, the results of (a) and (b) demonstrate that more denoising steps can improve robustness. DDPM gains more advantage from having more denoising steps than DDIM. (c) shows the effect of the number of denoising steps in the attack. As the number of denoising steps increases, the attack success rate slightly increases. However, we also find that increasing the number of denoising steps in an attack can decrease the attack success rate when the number of forward steps is 200 (i.e., \(t^* = 200\)).

\(^3\)The 20 denoising steps is the maximum limit of 40GB of memory.

---

4.4. The Number of Purification Steps

Although a single forward and reverse process can purify the input image, one can apply the purification process multiple times as proposed in Wang et al. [36]. We denote the number of forward and denoising processes as the number of \(\text{purification step}\). Similar to the case of the denoising step, computing the gradients of multiple purification steps is impossible due to memory constraints in most cases.

The number of purification steps can also differ between attack and defense. Through experiments, we measure the changes in robust accuracy with the different number of purification steps in the defense and attack. For all experiments, we fixed the number of forward steps to 100.
Table 4: Standard and robust accuracy of GDMP \cite{36} against BPDA and PGD+EOT $\ell_\infty (\epsilon = 8/255)$ on CIFAR-10. We compare two types of guidance, MSE and SSIM, and the defense without guidance.

<table>
<thead>
<tr>
<th>Guidance</th>
<th>Underlying Classifier</th>
<th>$t^*$</th>
<th>Robust Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No guide</td>
<td>TRADES \cite{41}</td>
<td>0</td>
<td>55.32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>200</td>
<td>51.52±1.96</td>
</tr>
<tr>
<td>MSE</td>
<td>Gowal et al. \cite{13}</td>
<td>0</td>
<td>69.03</td>
</tr>
<tr>
<td>SSIM</td>
<td></td>
<td>100</td>
<td>58.24±0.49</td>
</tr>
<tr>
<td></td>
<td></td>
<td>200</td>
<td>52.97±1.38</td>
</tr>
</tbody>
</table>

Table 5: Combination of diffusion models with adversarial training evaluated on CIFAR-10 against PGD+EOT $\ell_\infty (\epsilon = 8/255)$. Five denoising steps are used for both attack and defense.

<table>
<thead>
<tr>
<th>$t^*$</th>
<th>Defense</th>
<th>Attack</th>
<th>Robust Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>DDPM</td>
<td>DDPM</td>
<td>44.77±1.48</td>
</tr>
<tr>
<td></td>
<td>DDIM</td>
<td>DDPM</td>
<td>46.68±1.25</td>
</tr>
<tr>
<td>200</td>
<td>DDPM</td>
<td>DDPM</td>
<td>50.43±1.11</td>
</tr>
<tr>
<td></td>
<td>DDIM</td>
<td>DDPM</td>
<td>52.15±1.88</td>
</tr>
</tbody>
</table>

Table 6: Robust accuracy against PGD+EOT $\ell_\infty (\epsilon = 8/255)$ on CIFAR-10 when using a denoising model in attack different from the denoising model in defense.

4.5. Other Techniques

We evaluate several other techniques proposed in earlier work \cite{39, 24, 36} within our new evaluation framework.

Guidance. GDMP \cite{36} proposes to use gradients of a distance between an original input example and a target example to preserve semantic information while denoising. They show guidance can improve robustness against preprocessor-blind attacks. However, as shown in Table 4, when the gradients of the surrogate process are used in the attack, the guidance of GDMP decreases the robust accuracy. Specifically, the defense with guidance using the SSIM similarity has 6.88\% robust accuracy, which is 31.56\% lower than the defense without guidance.

Ensemble of multiple purification runs. ADP \cite{39} uses the ensemble of multiple purification runs as the predicted label to mitigate the randomness in the defense. For diffusion-based purification methods, as shown in Figure 5, multiple purification runs especially can help improve standard accuracy while the robust accuracy keeps the same level. In particular, for $t^* = 200$ with 40 purification runs, standard accuracy is 8.17\% higher than the case without ensemble.

Combination with adversarial training. An adjoint method based DiffPure \cite{24} shows robustness can be improved by using diffusion models together with adversarial training. However, as shown in Table 5, the adversarial training with purification shows lower robustness than the classifier without purification.

Transferability of gradients from different samplers in the attack. One may employ a sampler of diffusion mod-
els to generate adversarial examples different from those used in defense. For example, attacks using gradients from DDPM could be transferred to the defense using DDIM and vice-versa. We test whether the gradients from a different sampler of denoising models can improve the attack success rate. As shown in Table 6, although the transferred attack is valid, the attack success rates using different samplers are slightly lower than those using the original samplers.

### 5. Gradual Noise-Scheduling for Multi-Step Purification

In this section, we propose a new sampling strategy for diffusion-based purification and compare the performance with other state-of-the-art defenses.

**Gradual noise-scheduling strategy.** As highlighted in Section 4, selecting appropriate hyperparameter values is essential to improve robustness. Thus, we conduct an extensive exploration of hyperparameter settings to maximize robust accuracy. In particular, we mainly focus on the fact that each purification step can contain a different number of forward steps. We empirically find that fewer forward steps in the first few purification steps can improve robustness.

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Standard</th>
<th>PGD</th>
<th>AutoAttack</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRN-28-10</td>
<td>Goyal et al. [13]</td>
<td>87.51</td>
<td>66.01</td>
<td>63.38</td>
</tr>
<tr>
<td>AT</td>
<td>Goyal et al. [12]*</td>
<td>88.54</td>
<td>65.93</td>
<td>62.76</td>
</tr>
<tr>
<td></td>
<td>Pang et al. [25]</td>
<td>88.62</td>
<td>64.95</td>
<td>61.04</td>
</tr>
<tr>
<td>WRN-70-16</td>
<td>Yoon et al. [39]</td>
<td>85.66±0.51</td>
<td>53.48±0.86</td>
<td>59.53±0.87</td>
</tr>
<tr>
<td>AP</td>
<td>Nie et al. [24]</td>
<td>90.07±0.97</td>
<td>46.84±1.44</td>
<td>63.60±0.81</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>90.16±0.64</td>
<td>55.82±0.59</td>
<td>70.47±1.53</td>
</tr>
</tbody>
</table>

#### Table 7: Standard and robust accuracy against PGD+EOT (left: $\ell_\infty(\epsilon=8/255)$, right: $\ell_2(\epsilon=0.5)$) on CIFAR-10. Adversarial Training (AT) and Adversarial Purification (AP) methods are evaluated. * This method uses WideResNet-34-10 as a classifier. † This method uses WideResNet-34-10 as a classifier.

Based on this observation, for CIFAR-10, we set the number of forward steps as $\{30 \times 4, 50 \times 2, 125 \times 2\}$ for eight purification steps. For ImageNet and SVHN, we set the number of forward steps as $\{30 \times 4, 50 \times 2, 200 \times 2\} \times (30 \times 4, 50 \times 2, 80 \times 2)$, respectively. We set the number of denoising steps to equal the number of forward steps for all purification steps. We use the DDPM and an ensemble of ten purification runs.

#### Experimental settings. We conduct evaluations on three datasets, CIFAR-10 [20], ImageNet [9], and SVHN [23]. We use three diffusion model architectures, DDPM++ [35], Guided Diffusion [10], and DDPM [17] for each dataset. We use pretrained models for CIFAR-10 and ImageNet, but we trained a model for SVHN. Pretrained WideResNet-28-10, WideResNet-70-16, and ResNet-50 [40, 14] are served as baseline classifiers. We compare our method with adversarial training and diffusion-based purification methods. We evaluate diffusion-based purification methods on the PGD+EOT attack with 200 update iterations, except for ImageNet, which uses 20 iterations. We set the number of EOT to 20.
versarial training methods, we use 20 update iterations for the PGD attack. For our method, we report the worst robust accuracy with surrogate processes. We explain the detailed settings in Appendix D.

**Results.** Table 7 shows the defense performance against $\ell_\infty(\epsilon = 8/255)$ and $\ell_2(\epsilon = 0.5)$ threat models on CIFAR-10, respectively. Our method outperforms other diffusion-based purification methods. Specifically, compared to DiffPure on $\ell_\infty$ PGD attack, our method improves robust accuracy by 8.98% with WideResNet-28-10 and by 5.75% with WideResNet-70-16, respectively. Despite the improvement in robustness, the purification methods perform worse than the adversarial training methods. Table 8 shows the performance against $\ell_\infty(\epsilon = 4/255)$ threat model on ImageNet. Our method outperforms both adversarial training and purification methods. Compared to DiffPure and Salman et al. [30], our method improves robust accuracy by 4.34% and 3.94%, respectively. Results on SVHN against threats model $\ell_\infty(\epsilon = 8/255)$ are similar with CIFAR-10. Although our method improves robust accuracy by 15.35% compared to the DiffPure framework, which uses $t^* = 0.075$, our method performs worse than the adversarial training methods.

We additionally compare the robustness of our defense strategy with other adversarial purification methods against BPDA ($\ell_\infty(\epsilon = 8/255)$). As shown in Table 10, our proposed method outperforms all other adversarial purification methods, achieving a robust accuracy of 88.40%, 6.95% greater than the robust accuracy of DiffPure. Furthermore, Table 11 shows our robustness against other attacks, including the Square attack [1], a black-box attack. Our defense shows strong robustness higher than 80% against all attacks.

### 6. Related Work

Adversarial training [21, 41] is one of the most successful adversarial defense methods. These methods train a classifier with adversarial examples in a training phase. Zhang et al. [41] and Pang et al. [25] propose loss functions that can effectively utilize the trade-off between robustness and accuracy. Huang et al. [18] analyze architectural factors with respect to robustness. Rebuffi et al. [28] and Gowal et al. [13] improve robustness by utilizing data augmentations.

Adaptive test-time defenses purify adversarial examples using extra neural networks that utilize techniques from other domains. ADP [39] jointly uses denoising score matching and Langevin dynamics for purification. DiffPure [24] demonstrates from the stochastic differential equations (SDE) perspective that diffusion models can purify adversarial examples. GDMP [36] uses the guidance of diffusion models to recover adversarial examples as similar as possible to the original examples. SODEF [19] uses a Lyapunov-stable ODE block so that the input converges to a stable point that can be correctly classified. DISCO [16] is one of the denoising models that predict clean RGB value using local implicit functions.

### 7. Conclusion

Throughout the paper, we first analyze the current evaluation methods for diffusion-based adversarial purification and then propose a recommendation for the reliable evaluation of the robustness of adversarial purification. We further investigate the influence of hyperparameters of the diffusion model on the robustness of the purification. Based on our analysis, we propose a new strategy to maximize the benefit of the purification methods.

**Acknowledgements** This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01906, Artificial Intelligence Graduate School Program (POSTECH)) and National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2020R1C1C1011375)
References


[10] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. ArXiv, abs/2105.05233, 2021. 8


[23] Yuval Netzer, Tao Wang, Adam Coates, A. Bissacco, Bo Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. 2011. 8


[27] Rahul Rade and Seyed-Mohsen Moosavi-Dezfooli. Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In International Conference on Learning Representations, 2022. 8


[29] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A. Mann. Fixing data augmentation to improve adversarial robustness. ArXiv, abs/2103.01946, 2021. 8


[34] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. ArXiv, abs/1710.10766, 2017. 9


