arXiv:2501.16904v2 [cs.CV] 26 Feb 2026

Diffusion or Non-Diffusion Adversarial Defenses: Rethinking the Relation between Classifier and Adversarial Purifier

Yuan-Chih Chen and Chun-Shien Lu

Institute of Information Science, Academia Sinica, Taipei, Taiwan, ROC
{willpower057, lcs}@iis.sinica.edu.tw

Abstract. Adversarial defense research continues to face challenges in combating against advanced adversarial attacks, yet with diffusion models increasingly favoring their defensive capabilities. Unlike most prior studies that focus on diffusion models for test-time defense, we explore the generalization loss in classifiers caused by diffusion models. We compare diffusion-based and non-diffusion-based adversarial purifiers, demonstrating that non-diffusion models can also achieve well performance under a practical setting of non-adaptive attack. While non-diffusion models show promising adversarial robustness, they particularly excel in defense transferability and color generalization without relying on additional data beyond the training set. Notably, a non-diffusion model trained on CIFAR-10 achieves state-of-the-art performance when tested directly on ImageNet, surpassing existing diffusion-based models trained specifically on ImageNet.

Keywords: Adversarial attack · Adversarial defense · Classifier · Defense transferability · Diffusion model.

1 Introduction

Proliferation of deep learning models across various domains has raised a pressing concern: Vulnerability of these models to adversarial attacks [2,5,22,24] that aim to make the model behave abnormally by manipulating the input data with imperceptible perturbations. In response to these threats, researchers have actively investigated techniques to enhance the robustness of machine learning models against adversarial attacks in two branches: (1) Adversarial training (*e.g.*, [16,17,30,35,39,36]) involves both the clean and adversarially perturbed data for model training to improve robustness. While numerous studies have explored adversarial training, a notable disparity (from RobustBench) persists between the natural accuracy and robust accuracy. (2) Adversarial purification (*e.g.*, [1,14,25,33,42]) takes a different approach by removing adversarial perturbations from input data prior to classification. The benefit of adversarial purification is that it eliminates the need to retrain the classifier and can generalize to different attacks at test time.

Although diffusion models exhibit strong generalization across different attacks—known as "attack generalization"—diffusion-based adversarial purifiers rely on pre-trained diffusion models to map images back to the training data distribution. However, this comes at the cost of reduced "classifier generalization," making them sensitive to image processing like color variation and limiting their transferability across datasets. This limitations arise due to the differences in data augmentation and training objectives between the classifier and diffusion model.

Unlike most previous studies focusing on enhancing the effectiveness of cleansing adversarial images with diffusion models, we examine the classifier generalization loss linked to diffusion purifiers in this paper. To explore this issue, we consider the practical scenario of non-adaptive attack setting and different input variations. Our findings reveal that a non-diffusion purifier, trained using purification loss, can mitigate this generalization loss. Furthermore, we explain the underlying reasons for this phenomenon and highlight the differences between diffusion-based purifiers and non-diffusion-based purifiers.

The main contributions of this paper include:

- Unlike most prior studies that focus on enhancing the performance of diffusion-based adversarial purifiers, our work is the first to investigate the classifier generalization loss induced by such purifiers.
- We explain why diffusion-based purifiers degrade classifier accuracy when processing images with slight variations from the training data, whereas purification-loss-based purifiers, a kind of non-diffusion-based purifiers, better preserve classification performance.
- We observe that diffusion-based purifiers are particularly sensitive to color variations. To investigate this phenomenon, we propose ColoredImageNet, a modified ImageNet to evaluate the impact of color shifts on purification effectiveness.

2 Related works

In this section, we review recent advancements in adversarial purification methods, followed by an exploration of how Masked Autoencoders (MAE) [11] have been applied to address the problems of demanding robustness.

Anti-Adv [1] introduces an anti-adversary layer designed to steer the image x away from the decision boundary. The perturbation direction is guided by the image prediction via classifier, given the absence of true labels during inference. However, this strategy depends on the adversarial image being classified correctly—an assumption that may not always hold—potentially leading to misdirected corrections. DISCO [14] uses the concept of LIIF [3] to extract the per-pixel feature by a pre-trained EDSR [23].Its training focuses solely on purification loss. DISCO demonstrates both acceptable robust accuracy and strong model transferability across different datasets. DiffPure [25] employs a diffusion model for image purification and provides a theoretical guarantee: By introducing sufficient Gaussian noises in the forward process, adversarial perturbations can

be effectively eliminated. Regardless of the classifiers or attacks, DiffPure remains effective with the caveat that the diffusion timestep must strike a balance. Actually, it should be large enough to remove adversarial perturbations yet small enough to preserve global label semantics. Building on DiffPure, ScoreOpt [42] introduces score-based priors into the purification process. Adversarial samples are optimized to converge toward regions with higher posterior likelihood, as defined by pre-trained score-based models. More recently, MimicDiffusion [33] diverges from DiffPure's approach. Instead of adding noise to adversarial images, it starts from pure Gaussian noise and applies a reverse diffusion process guided by the adversarial input to generate purified images.

On the other hand, MAE [11] implements a masking mechanism to enhance the performance of ViT [8]. Inspired by the Masked Language Modeling (MLM) technique used in BERT [19], MAE operates as a pre-training model, focusing on learning patch representations during the pre-training stage and fine-tuning for downstream tasks. Hereafter, MAE and MLM will be interchangeably used.

Recently, some works [18,34,39,37] have employed MAE for the problems of demanding robustness. Huang *et al.* [18] and DMAE [37] focused on robustness in the context of classification tasks, rather than on image purification. DRAM [34] proposes a test-time detection method to repair adversarial samples in that the MAE reconstruction loss is directly used to detect the adversarial samples due to the assumption of different distributions of clean and adversarial samples. NIM-MAE [39] uses the MAE structure to achieve adversarial training by injecting the noise into the entire image instead of masking patches within an image.

3 Difference between Diffusion-Based Purifiers and Purification-Loss-Based Purifiers

In this section, we first introduce the preliminary and notation in Section 3.1, followed by an analysis of the accuracy drop caused by diffusion-based purifiers in Section 3.2. We examine the effectiveness of purification loss in Sec. 3.3 that will be an important component in our proposed non-diffusion-based purifier.

3.1 Preliminary and Notation

Frequently used notations are as follows: Clean image x and its corresponding label y; adversarial image x_a; classifier c, which outputs the predicted label $\tilde{y}$; an adversarial purifier $\mathcal{P}$; an image $x \in \mathbb{R}^{H \times W}$ cropped into N patches of area $ps \times ps$ (w.r.t. patch size ps) before forwarding to purifier; purifier $\mathcal{P}_{MAE} = g \circ f$ with MAE encoder f and MAE decoder g; masking ratio r of MAE and corresponding binary mask M; and $\mathbf{1}$ is a matrix with all elements of 1.

Adversarial Attack. Given a classifier c parameterized by θ, and a clean data pair (x, y), an adversarial attack aims to find an adversarial sample x_a derived from x to deceive the classifier (*i.e.,* $c(x_a) \neq y$) by the optimization as: $\max_{x_a} L(x_a, y; \theta)$, s.t. $||x_a - x||_p < \epsilon$, where L is the training loss function, $||\cdot||_p$ denotes p-norm, and ϵ means the attack budget.

Generalized Purifier. Given a classifier c and a clean data pair (x, y), a purifier $\mathcal{P}$ aims to refine x_a to prevent misclassification (*i.e.,* ensuring $c(\mathcal{P}(x_a)) = c(x)$) while preserving clean accuracy (*i.e.,* $c(\mathcal{P}(x)) = c(x)$). In this work, we further emphasize that the purifier should maintain the classifier's generalization capability, meaning it should also correctly process unseen images, particularly common real-world images (*i.e.,* $c(\mathcal{P}(x')) = c(x')$), where x' represents such unseen data.

Diffusion Model. DDPM [15] consists of a T-step forward process and a corresponding T-step reverse process. The forward process gradually corrupts a clean image $x_0 \sim q(x)$ by adding Gaussian noise, resulting in a noisy image x_T. The reverse process then aims to reconstruct the original image by denoising x_T back to x_0. The forward process is defined as:

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}}x_0, (1 - \bar{\alpha}_t)\mathbf{I}), \tag{1}$$

where t is the diffusion timestep, α_t is a predefined noise schedule, and $\bar{\alpha}_t = \prod_{i=0}^{t} \alpha_i$ represents the cumulative product of the noise schedule. The distribution $q(x)$ represents the clean training data distribution. The reverse process approximates the posterior $q(x_{t-1} \mid x_t)$ by iteratively sampling:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t\mathbf{I}), \tag{2}$$

where $\mu_\theta = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t))$, $\sigma_t\mathbf{I}$ is untrained time dependent constant, and ϵ_θ denotes the approximate noise predicted by the neural network. To train the model ϵ_θ, a clean image $x_0 \sim q(x)$ is randomly selected, and Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is added at a random timestep t. The model is trained to predict the noise component using the following objective:

$$\nabla_\theta ||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}}x_0 + \epsilon\sqrt{1 - \bar{\alpha}})||_2^2. \tag{3}$$

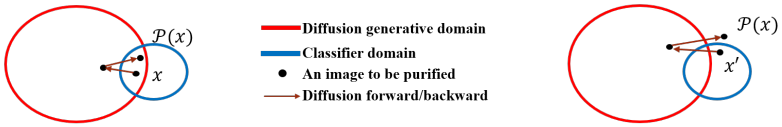
Masked Autoencoder. Given an image $x \in \mathbb{R}^{H\times W}$ and a binary mask $M \in \mathbb{R}^{H\times W}$, the goal of MAE [11] is to reconstruct the entire image $\bar{x} = x \in \mathbb{R}^{H\times W}$ from partial image $M \odot x$ with the reconstruction loss defined as:

$$L_{MAE}(x, \bar{x}) = \|(\mathbf{1} - M) \odot \bar{x} - (\mathbf{1} - M) \odot g \circ f(M \odot x)\|_2, \tag{4}$$

where $\mathbf{1}$ is a matrix equal to $1^{H\times W}$, $\odot$ is the element-wise product of two matrices of the same size, f is MAE encoder, g is MAE decoder, and M is a random binary image mask parameterized by the masking ratio r as $||M||_1 = (1-r) \times (H \times W)$. The masking ratio $r = 0$ if there is no masking.

3.2 The Discrepancy between Classifier and Diffusion Purifier

The predominant purification-based approaches [25,33,42] in adversarial defense involve the diffusion model, which primarily modifies the reverse diffusion process



(a) (b)

Fig. 1: Illustration of the discrepancy between the classifier and diffusion-based purifier. (a) When the input image x is from the training set or a similar distribution (e.g., test data split from the same dataset), it has a higher likelihood of maintaining clean accuracy, as shown in previous studies. (b) However, if the input image x' is an unseen sample that the classifier can correctly identify, diffusion-based purifiers are more prone to misclassification.

of DiffPure [25] using the same pre-trained model. However, these approaches primarily use clean accuracy to demonstrate that the purifier preserves the classifier's ability to classify non-attacked images. Different from prior studies, we find that directly applying a pre-trained diffusion model can actually be detrimental to the classifier—an issue that cannot be detected solely through clean accuracy.

As illustrated in Fig. 1, the generative domain of a diffusion model (in red circle) differs from the classification domain of a classifier (in blue circle). The intersection of both circles contains the shared training data for both models, while the outer region of blue circle indicates the areas where mis-classification occurs. Fig. 1(a) depicts scenarios commonly used in previous studies to evaluate clean accuracy, whereas Fig. 1(b) illustrates how a purifier can degrade classifier performance when the test data slightly deviates from training data.

To verify the scenario in Fig. 1(a), we present experiments in Sec. 5.2. As for Fig. 1(b), we analyze the effect of color variations in Sec. 5.3 and common color corruptions in Sec. 5.6 to support our claim. Furthermore, we investigate a scenario in that the training data for purifiers and classifiers do not overlap (*i.e.*, the red and blue circles have no intersection). For this scenario, we explore in Sec. 5.4, where a purifier is applied to a dataset different from its training dataset, and in Sec. 5.5, where we examine the transition from a low-resolution dataset to a high-resolution one.

To analyze the aforementioned scenarios, we define a test image that slightly deviates from the training data distribution as follows:

$$x' = y = Ax + n, \tag{5}$$

where x is a clean image sampled from the training distribution $q(x)$, A is a transformation operator, n is additive Gaussian noise, and y is the resulting transformed (or corrupted) image. When A is the identity matrix and $n = 0$, the transformation reduces to the original image, *i.e.*, $y = x$. The forward process of the diffusion-based purifier $\mathcal{P}$ can then be formulated by combining Eqs. (1)

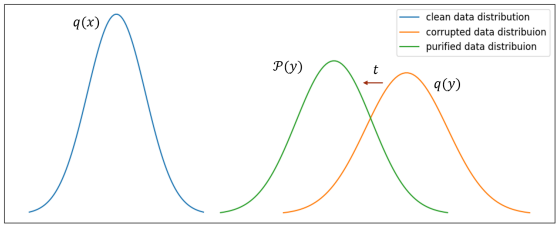


Fig. 2: Visualization of the purified image distribution across various timesteps t.

and (5) as $q(x_t \mid y) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}y, (1-\bar{\alpha}_t)\mathbf{I})$, where the purifier operates using a reduced number of diffusion steps, typically with $t \in \{0.1, 0.15\}$, in contrast to the full diffusion process ($t = T$) employed in standard DDPMs. Notably, when $t = T$, the diffusion purifier becomes equivalent to a full DDPM, since $q(x_T \mid y) \sim \mathcal{N}(0, \mathbf{I})$, leading to $\mathcal{P}(y) \sim q(x)$. Conversely, when $t = 0$, the purifier simply returns the input, *i.e.*, $\mathcal{P}(y) = Ax + n$.

According to Theorem 3.1 in [25], the relationship between the distributions $q(y)$ and $q(x)$ during the diffusion process satisfies:

$$\frac{\partial D_{KL}(q(y)||q(x))}{\partial t} \leq 0, \tag{6}$$

indicating that the KL divergence between them decreases monotonically as t increases from 0 to 1. Consequently, the purified distribution $\mathcal{P}(y)$ transits from $q(y)$ toward $q(x)$, as illustrated in Fig. 2.

In contrast to the ideal case of $A = I$ and $n = 0$, applying a non-trivial transformation A introduces a distributional shift. As a result, for $0 < t < 1$, the purified image $\mathcal{P}(y)$ deviates from $q(y)$, potentially leading to a drop in clean accuracy $c(\mathcal{P}(y))$ due to the divergence between the purified output and the true corrupted input distribution.

Specifically, diffusion models tend to generate images similar to training data, as noted in [32]. In contrast, classifiers are trained with data augmentation to enhance generalization and prevent overfitting. Since diffusion models focus on generating natural images, common data augmentation techniques—such as color jitter, rotation, and mixup—are typically avoided to prevent the generation of unrealistic or unnatural images. This fundamental difference leads to a discrepancy: classifiers leverage data augmentation to learn texture variations, improving their ability to classify unseen images, while diffusion models push unseen images toward the closest points in the training distribution, which may not align with the classifier's expectations. As we will demonstrate later in Sec. 5.3, this mismatch in augmentation strategies results in accuracy drops when handling color variations.

To give a preliminary expression, as shown in Table 1, diffusion models exhibit lower SSIM and PSNR scores compared to non-diffusion-based purifiers (Proposed MAEP in Sec. 4), indicating a decline in image quality after purification. Furthermonre, the purified images generated by diffusion models and those

Defense Methods	PSNR (↑)	SSIM (↑)
DiffPure [25]	25.50	0.73
ScoreOpt-N [42]	15.99	0.31
ScoreOpt-O [42]	22.01	0.52
MimicDiffusion [33]	22.06	0.68
DDA [9]	25.51	0.75
MAEP (Sec. 4)	**34.80**	**0.93**

Table 1: Evaluations of purifying images by non-diffusion based purifier (MAEP) and state-of-the-art diffusion-based methods. Testing dataset: ImageNet.

produced by the purifier trained with purification loss (MAEP) are shown in Fig. 3. It is evident that recent diffusion-based purifiers, such as DiffPure [25], ScoreOpt [42], and MimicDiffusion [33], significantly alter image details during the purification process, whereas the purification loss-based approach effectively preserves more of the original image details while maintaining robustness. Simply altering the reverse process of diffusion-based purifiers by introducing randomness or estimation can significantly increase semantic loss in the image. For example, this effect is observed in ScoreOpt, as demonstrated later in Sec. 5.6.

While previous studies suggest that diffusion-based purifiers can maintain clean accuracy without necessarily preserving image quality, we argue that their limitations warrant further exploration. Specifically, these purifiers rely solely on the training dataset without access to classifier-specific information, which may impact their ability to effectively purify images. Additionally, the substantial alterations introduced during purification can lead to information loss for the classifier. This issue is often overlooked because hyperparameters are typically fine-tuned using validation accuracy metrics, inadvertently masking the impact of image quality degradation.

3.3 Purification Loss vs. Clean Accuracy

In the literature, DISCO [14] is found to achieve both acceptable clean and robust accuracy by employing just one purification loss, while preserving model transferability. The purification loss is defined as:

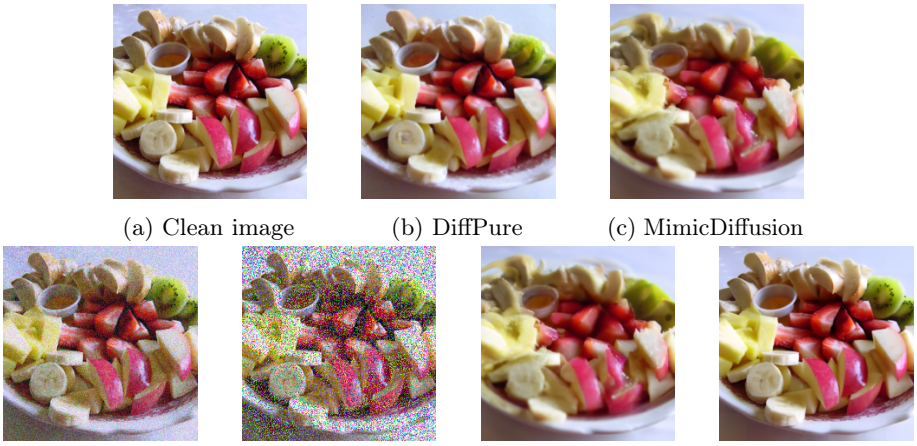
$$L_{purify}^{DISCO}(x, \mathcal{P}(x_a)) = \ell_1(x, \mathcal{P}(x_a)), \tag{7}$$

where $\mathcal{P}$ is a purifier used to purify input image x_a by reconstructing the clean image x in terms of ℓ_1-norm loss between x and x_a.

Specifically, DISCO shows that it can purify the adversarial image efficiently, expressed as:

$$\mathcal{P}(x_a) \approx x, \tag{8}$$

$$c(\mathcal{P}(x_a)) \approx c(x), \tag{9}$$



(a) Clean image (b) DiffPure (c) MimicDiffusion

(d) ScoreOpt-O (e) ScoreOpt-N (f) DDA (g) MAEP

Fig. 3: Illustration of purified clean images from ImageNet using diffusion-based purifiers (DiffPure [25], MimicDiffusion [33], ScoreOpt [42], and DDA [9]) compared to the purification loss-based purifier (MAEP in Sec. 4): (a) The original, unmodified image from ImageNet. (b) DiffPure introduces significant alterations to image details and textures, with noticeable changes in the appearance of the kiwi, bread, and strawberry. (c) MimicDiffusion substantially modifies the structural composition of the image. (d-e) ScoreOpt, after the denoising process, still retains some noises. (f) DDA, guided by a low-pass filter and designed to restore corrupted images to their natural states, still alters the image's texture. (g) MAEP applies minimal modifications, such as slight smoothing, to effectively purify the image while preserving its original structure.

where Eq. (8) denotes the perceptual similarity between the clean image x and purified image $\mathcal{P}(x_a)$, Eq. (9) indicates label-preservation, and c is a pre-trained classifier from RobustBench or PyTorch official website. Nevertheless, there is still a room for DISCO to improve label-preservation for purified clean images, defined as:

$$c(\mathcal{P}(x)) \approx c(x). \tag{10}$$

Although DISCO [14] does not ensure to preserve the clean accuracy in Eq. (10), it indeed shows good trade-off between the clean accuracy and robust accuracy in several testing scenarios, including different classifiers, different attack algorithms (such as Autoattack [5], PGD [24], FAB [4], BIM [22], BPDA [2], and FGSM [10]), and transferability to different datasets.

Based on the above observations, we conjecture that the purification loss ($\ell_1(\mathcal{P}(x_a), x)$ in Eq. (7)) can efficiently purify the adversarial image without remarkably sacrificing clean accuracy. Here, we provide a feasible but simple explanation of why DISCO can have acceptable performance without needing to consider $\ell_1(\mathcal{P}(x), x)$, as illustrated in Fig. 4.

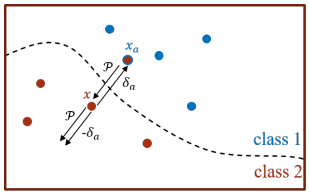


Fig. 4: Purification loss learns the direction from x_a to x and the direction of $-\delta_a$ (one-step adversarial perturbation along negative gradient) is roughly the same as $\mathcal{P}(x_a)$ due to Eq. (11).

	Acc. (%)	Formula
$c(\mathcal{P}(x_a))$	95.16	Eq. (9)
$c(\mathcal{P}(x))$	89.94	Eq. (13)

Table 2: Verification of our conjecture on the purification loss ($\ell_1(\mathcal{P}(x_a), x)$ in Eq. (7). For $c(\mathcal{P}(x)) \approx c(x - \delta_a)$, we pre-process the training dataset by adding $-\delta_a$ (via PGD) and feeding it to a non-defense classifier (ResNet-18) pre-trained on CIFAR-10 for testing. Since our derivation introduces approximation to remove the influence of purifier and is only based on a classifier, it means that this is a theoretical accuracy of a purifier and Eq. (7) can properly maintain clean accuracy.

Specifically, based on the assumption that an adversarial image is created to be similar to its clean counterpart in term of L_∞-norm as:

$$x \approx x_a \ s.t. \ |x - x_a|_\infty < \epsilon, \tag{11}$$

we can derive

$$\mathcal{P}(x) - x \approx \mathcal{P}(x_a) - x_a = -\delta_a, \tag{12}$$

where the direction of purification, $\mathcal{P}(x) - x$, for clean image x is similar to $\mathcal{P}(x_a) - x_a$ of an adversarial image due to the adversarial condition specified in Eq. (11).

In practice, input images fall into two categories: adversarial images (x_a) and clean images (x). The robust accuracy of purifier is tied to x_a and the prediction of x_a can be calculated by $c(\mathcal{P}(x_a)) \approx c(x)$, as detailed in Eq. (9). The clean accuracy of purifier, which DISCO didn't make a discussion, is related to x and the prediction of $\mathcal{P}(x)$ can be derived from Eq. (12) as:

$$c(\mathcal{P}(x)) = c(x + (\mathcal{P}(x) - x)) \approx c(x + (\mathcal{P}(x_a) - x_a)) = c(x - \delta_a), \tag{13}$$

where $\mathcal{P}(x_a) - x_a$ is roughly equal to $-\delta_a$, as illustrated in Fig. 4. To verify the above conjecture, we present the results in Table 2.

Overall, to better preserve image semantics while maintaining clean accuracy, we propose incorporating the purification loss (Eq. (7)) into Masked Language Model (MLM) in Sec. 4.

4 Masked AutoEncoder Purifier (MAEP)

Building on our findings in Sec. 3.2 and Sec. 3.3, MAEP leverages the masking mechanism to enhance purification by enforcing a purification loss that removes adversarial perturbations while preserving image semantics through a reconstruction loss.

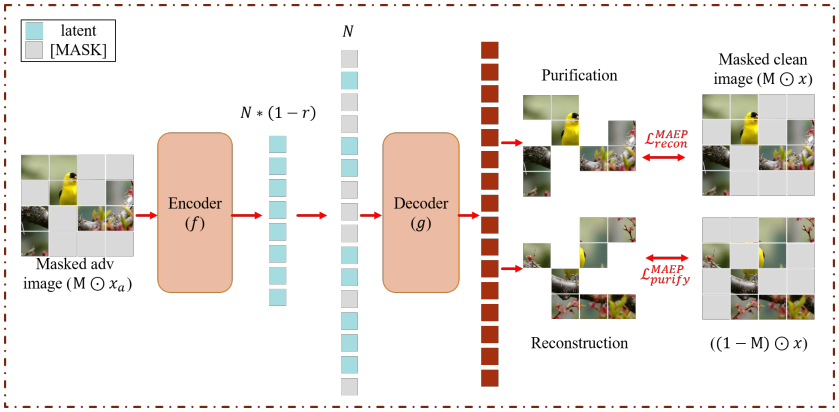


Fig. 5: MAEP workflow: MAEP purifies adversarial images using purification loss. It then learns adversarial patch representations to further enhance purification while preserving semantic information by masking clean patch predictions.

4.1 Objective Function Design

We study diverse objective function designs [11,14,19,30,43], all geared towards enhancing robust accuracy. These designs include TRADES [43], MLM [11,19], and reconstruction loss [14,30], where TRADES is a popular choice for training robust classifiers, MLM showcases effectiveness in vision tasks, and reconstruction loss is often used for image reconstruction. We will delve into a discussion of the results produced by these designs, emphasizing the superior performance of MAEP among these alternatives. In the following, the distance measure D is chosen to be ℓ_1-norm as we find it performs better than mean square error. The descriptions regarding reconstruction loss and extensions based on TRADES and MLM will be shown in Sec. 8 of Appendix. In addition, the performance comparison of all loss designs can also be found in Table 10 of Sec. 8.

4.2 Total Loss in MAEP

Unlike the aforementioned design, MAEP is investigated to integrate both the reconstruction loss of MLM and purification loss in Eq. (7) to boost adversarial robustness and maintain the semantic of the image. We also show that solely utilizing MAE/MLM does not yield satisfactory performance in Sec. 8 of Appendix.

The entire loss of MAEP can be separated into two parts. Part 1). The purification loss, adapted from Eq. (7), addresses the purification task focusing solely on the unmasked region within an image. Several reasons support this decision to exclusively handle the unmasked region. First, training with partial image information can generalize to the entire image via position embedding, as outlined in MAE [11]. Second, the purifier is designed to operate on the entire image rather than the masked region. Third, the incorporation of MLM discussed

in Part 2) below can effectively address the issue of dealing with the masked region. Part 2). Unlike Part 1), which utilizes the unmasked region for purification, MLM reconstructs the masked region of an image using the unmasked portions. This masking mechanism helps the model learn adversarial representations and identify adversarial perturbations, thereby enhancing performance. Additionally, this loss ensures that MAEP preserves the semantic integrity of the image while maintaining the benefits of purification loss.

Based on the above concerns, it is ready to define the MAEP loss. First, we define the purification loss. Recall that $g \circ f$ is called the purifier in MAE. To ensure the clean accuracy and robust accuracy, as described in Sec. 3.3, we adopt Eq. (7) for masked region and the purification loss of unmasked region in MAEP is defined as:

$$L_{purify}^{MAEP} = \|M \odot x - M \odot g \circ f(M \odot x_a)\|. \tag{14}$$

Note that L_{purify}^{MAEP} reconstructs the clean image x based on the adversarial image x_a, which is consistent with Part 1) and is different from the traditional MAE, as shown in Eq. (4).

Second, following the reconstruction loss ($L_{pre-train}$ in Eq. (17) of Appendix) in MAE, the reconstruction loss L_{recon}^{MAEP} of masked region in MAEP is defined as $L_{recon}^{MAEP} = L_{pre-train}$. Thus, the entire loss of MAEP is derived as:

$$\begin{aligned} L^{MAEP} &= L_{purify}^{MAEP} + L_{recon}^{MAEP} \\ &= \|M \odot x - M \odot g \circ f(M \odot x_a)\| + \|(\mathbf{1} - M) \odot x - (\mathbf{1} - M) \odot g \circ f(M \odot x_a)\| \\ &\geq \|x - g \circ f(M \odot x_a)\|, \end{aligned} \tag{15}$$

where the equal sign holds when $\|\cdot\|$ is ℓ_1-norm, which is adopted as the distance measure in MAEP. The masking ratio $r \in (0, 1)$ controls the image mask M. Eq. (15) will degenerate to the purification loss in Eq. (7) when $r = 0$.

5 Experiments

In Sec. 3, we discuss the differences between diffusion-based and purification loss-based purifiers. To better preserve semantic information, we introduce MAEP, which enhances the training of purification-loss-based purifiers using a masked language modeling (MLM)-inspired objective, in Sec. 4. In this section, we first present robust accuracy results to demonstrate their competitive performance. We then conduct performance analysis under varying levels of discrepancy between training and inference data, from minor differences (*e.g.*, color variations) to more significant shifts (*e.g.*, different datasets), providing a clear understanding of the distinctions between non-diffusion-based purifiers and diffusion-based purifiers.

5.1 Datasets, Model Settings, and Implementation Details

Four commonly used datasets, CIFAR10 [21], CIFAR100 [21], ImageNet [7], and ImageNet-C [12] were adopted. For ColoredImageNet used here, it was generated

Defense Methods	Clean Accuracy (%)	Robust Accuracy (%)	Average Accuracy (%)	Attacks
No defense	94.78	0	47.39	PGD-ℓ_∞/AutoAttack (Standard)
AWP [36]*	88.25	60.05	74.15	AutoAttack (Standard)
Anti-Adv [1]* + AWP	88.25	79.21	83.73	AutoAttack (Standard)
DISCO [14]	89.26	82.99	86.12	PGD-ℓ_∞
DISCO [14]	89.26	85.33	87.29	AutoAttack (Standard)
DiffPure [25]	88.62	87.12	87.87	PGD-ℓ_∞
DiffPure [25]	88.15	87.29	87.72	AutoAttack (Standard)
ScoreOpt-N [42]	91.03	80.04	85.53	PGD-ℓ_∞
ScoreOpt-O [42]	89.16	89.15	89.15	PGD-ℓ_∞
ScoreOpt-N [42]	91.31	81.79	86.55	AutoAttack (Standard)
ScoreOpt-O [42]	89.18	89.01	89.09	AutoAttack (Standard)
SOAP* [31]	96.93	63.10	80.01	PGD-ℓ_∞
Hill *et al.* [13]*	84.12	78.91	81.51	PGD-ℓ_∞
ADP ($\sigma = 0.1$) [38]*	93.09	85.45	**89.27**	PGD-ℓ_∞
MAEP	92.31	86.19	89.25	PGD-ℓ_∞
MAEP	92.30	88.73	**90.52**	AutoAttack (Standard)

Table 3: Robustness evaluation and comparison. Classifier: WRN-28-10. Testing dataset: CIFAR-10. Asterisk (*) indicates that the results were excerpted from the papers.

Defense Methods	Clean Accuracy (%)	Robust Accuracy (%)	Average Accuracy (%)	Attacks
No defense	81.66	0	40.83	PGD-ℓ_∞/AutoAttack (Standard)
Rebuffi *et al.* [27]	62.41	32.06	47.23	AutoAttack (Standard)
Wang *et al.* [35]	72.58	38.83	55.70	AutoAttack (Standard)
Cui *et al.* [6]	73.85	39.18	56.51	AutoAttack (Standard)
DISCO [14]	69.78	73.36	**71.57**	PGD-ℓ_∞
DISCO [14]	69.78	76.91	73.34	AutoAttack (Standard)
DiffPure [25]	61.96	59.27	60.61	PGD-ℓ_∞
DiffPure [25]	61.98	61.19	61.58	AutoAttack (Standard)
MAEP	73.67	70.96	**71.57**	PGD-ℓ_∞
MAEP	73.67	76.22	**74.95**	AutoAttack (Standard)

Table 4: Robustness evaluation and comparison.Classifier: WRN-28-10. Testing dataset: CIFAR-100.

by applying the method [28] to transform ImageNet images to match the color styles of target images. The target images consist of 20 samples from the ImageNet test set, resulting in a dataset that is 20 times the size of the original ImageNet. All models were trained using NVIDIA V100 GPUs.

For model architectures, we followed previous studies to employ WRN-28-10 [40] and its corresponding model weights provided by RobustBench for CIFAR10. However, for CIFAR100 and ImageNet, due to the absence of model weights from RobustBench, we adopted them from DISCO [14] and PyTorch, respectively. For the attacks, we consider PGD-ℓ_∞ and AutoAttack, and set the permissible perturbation ϵ such that $|\epsilon|_\infty \leq 8/255$.

To train the purifier, we first pre-trained MAEP from scratch using the loss function defined in Eq. (15), with a masking ratio of $r = 0.5$ and a patch size of 2. During inference, the masking ratio was set to $r = 0$ to fully utilize the input for downstream tasks. The clean and robust accuracy results were averaged over five runs with different random seeds and followed the non-adaptive setting of DiffPure [25].

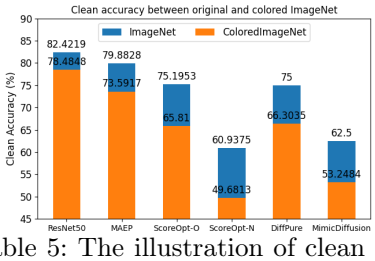


Table 5: The illustration of clean accuracy drop across different purifiers on ColorImageNet.

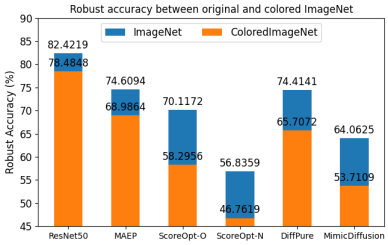


Table 6: The illustration of robust accuracy drop across different purifiers on ColorImageNet.

5.2 Evaluation of Adversarial Defense

We adopted SOTA adversarial purifiers for comparison, including diffusion model-based approaches [25,42,38] and non-diffusion-based approaches [1,13,14,31,36]. For DiffPure [25], we used the official code and tested the purifier under the same experimental setup mentioned in Sec. 5.1. For ScoreOpt [42], the classifier was trained by the authors and not from RobustBench. For a fair comparison, we used the official code and only replaced the default classifier of ScoreOpt with the WRN-28-10 model provided by RobustBench.

Table 3 shows the comparison results with dataset CIFAR-10. We have observations as follows: (1) For robust accuracy, MAEP and ScoreOpt-O [42] are comparable but better than DiffPure [25]. However, for clean accuracy, MAEP performs better than [25][42]. (2) MAEP and diffusion-based defenses are generally better than other methods in terms of average accuracy.

For CIFAR100, the robustness comparison results are shown in Table 4. Different from CIFAR-10, under CIFAR-100, DISCO performs better than DiffPure for both clean and robust accuracy, and MAEP outperforms DISCO and DiffPure with a large gap. It is noteworthy that for both MAEP and DISCO, their robust accuracy is higher than clean accuracy. One possible explanation is that they primarily learn the mapping from the adversarial image x_{adv} to the clean image x. This conforms to the verification in Table 2 and depicts that the theoretical optimal situation, in which the robust accuracy is higher than clean accuracy, may exist.

5.3 Sensitivity to Color Transform in Diffusion-based Purifiers

As we argue in Sec. 3.1, the purifier should not degrade the classifier's accuracy. In this section, we aim to demonstrate that if a classifier learns the texture features of a class—such as a cat from a training dataset of brown cats, does it still correctly classify cats of different colors? To the best of our knowledge, however, no such dataset currently exists. Therefore, we generate "ColoredImageNet" using a color transfer technique [28], as described in Sec. 5.1. In Sec. 5.2, we have already demonstrated that non-diffusion and diffusion-based purifiers achieve

Model	Training Data		Test Data		Clean Acc. (%)	Robust Acc. (%)	Avg. Acc. (%)
	CIFAR10	CIFAR100	CIFAR10	CIFAR100			
WRN28-10	v		v		94.78	0	47.39
+ DiffPure [25]	v		v		89.58	89.45	89.51
+ DISCO [14]	v		v		89.26	85.33	87.29
+ MAEP	v		v		92.30	88.73	**90.51**
+ DiffPure [25]		v	v		94.50	69.00	81.75
+ DISCO [14]		v	v		89.78	87.44	**88.61**
+ MAEP		v	v		91.58	84.73	88.16

Table 7: Transferability of adversarial defenses (DISCO, Diffpure, and MAEP) from CIFAR100 [20] to CIFAR10 [20] (trained on CIFAR100 but tested on CIFAR10) under WRN28-10 [41] and AutoAttack [5] with attack budget $\epsilon_\infty = 8/255$. Avg. Acc. is the average of clean acc. and robust acc. and used to show overall performance.

Model	Training Data		Test Data		Clean Acc. (%)	Robust Acc. (%)	Avg. Acc. (%)
	CIFAR10	CIFAR100	CIFAR10	CIFAR100			
WRN28-10		v		v	81.66	0	40.83
+ DiffPure [25]		v		v	61.98	61.19	61.58
+ DISCO [14]		v		v	69.78	76.91	73.34
+ MAEP		v		v	73.67	76.22	**74.95**
+ ScoreOpt-O [42]	v			v	57.55	42.83	50.19
+ ScoreOpt-N [42]	v			v	54.87	54.37	54.62
+ DiffPure [25]	v			v	81.00	40.00	60.50
+ DISCO [14]	v			v	72.50	69.22	70.86
+ MAEP	v			v	75.37	68.75	**72.06**

Table 8: Transferability of adversarial defenses (ScoreOpt, Diffpure, DISCO, and MAEP) from CIFAR10 to CIFAR100 (trained on CIFAR10 but tested on CIFAR100) under WRN28-10 and AutoAttack with attack budget $\epsilon_\infty = 8/255$.

comparable robustness. Here, we extend our verification to scenarios, where test images differ only in color, as illustrated in Tables 5 and 6. The results reveal that diffusion-based purifiers are more sensitive to color variations, as indicated by the difference between the blue bars for ImageNet and the orange bars for ColoredImageNet. Specifically, diffusion-based approaches—ScoreOpt [42], DiffPure [25], and MimicDiffusion [33]—experience an accuracy drop approximately twice as large as that of MAEP.

5.4 Defense Transferability

In this section, we examine diffusion-based purifiers [25,33,42] under a more challenging setting, where the purifier is applied to a dataset different from the one it was trained. As shown in Tables 7 and 8, these approaches exhibit limited transferability across datasets.

Table 7 demonstrates that while DiffPure maintains a small gap between clean and robust accuracy on CIFAR10, its robust accuracy drops significantly from

89.45% to 69.0% when applied to CIFAR100. A similar performance degradation is observed in the reverse transfer setting (Table 8). We highlight DiffPure as a representative method of diffusion-based defenses, as other approaches follow a similar paradigm by modifying the reverse diffusion process.

Crucially, in real-world scenarios, access to a well-trained diffusion model for every potential dataset is often infeasible, and training such models from scratch is computationally expensive—especially for small or diverse datasets. Furthermore, even when the training and testing datasets are closely related, as in the case of CIFAR10 and CIFAR100, diffusion-based defenses [25,42] suffer from a notable decline in robust accuracy. This highlights a key limitation: a lack of generalization and resilience to slight variations in image distributions.

5.5 Transferability to High-Resolution Dataset

In addition to Tables 7 and 8, we evaluated the transferability of purifiers from a low-resolution dataset to a high-resolution dataset in Table 9.

When transferring from CIFAR-10 to ImageNet, MAEP achieves approximately 75% clean accuracy, outperforming both DiffPure (68.60%) and ScoreOpt (68.05%) at attack budget $\epsilon_\infty = 4/255$, even though both baselines are trained directly on ImageNet. When attack budget was increased to $\epsilon_\infty = 8/255$, MAEP still maintains promising accuracy. We also evaluated the robust accuracy of DDA, which is specifically designed to maintain classification performance under image corruptions. However, our results show that DDA fails to preserve robustness in this setting.

Additionally, MAEP incurs only a 3% drop in clean accuracy compared to the original classifier accuracy of 80.85% without any defense, whereas diffusion-based methods suffer a larger degradation of around 10%. This difference attributes to the fact that diffusion models introduce noises to remove adversarial perturbations, thereby reducing clean accuracy.

5.6 More Result with Common Color Corruptions

We further investigate the impact of additional color-related image corruptions, which are common in real-world settings. As illustrated in Fig. 6, diffusion-based methods exhibit heightened sensitivity to image corruptions. In particular, ScoreOpt demonstrates significant performance degradation. As discussed in Sec. 3.2, the stochasticity and approximation involved in ScoreOpt's reverse diffusion process amplify semantic loss, contributing to reduced robustness. Representative examples of purified images are shown in Fig. 3.

While DDA [9] is specifically designed to maintain prediction accuracy under corruption, we exclude its results here due to its vulnerability to adversarial perturbations, as evidenced in Table 9.

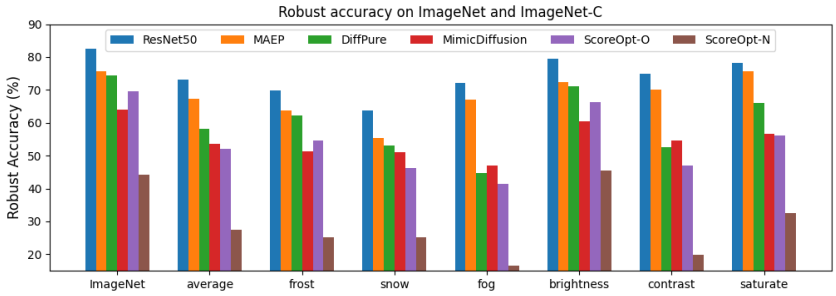


Fig. 6: Drop in robust accuracy across different purifiers on ImageNet-C. "ImageNet" refers to the original, unmodified images, while the other categories represent corrupted versions of ImageNet. The term "average" denotes the mean accuracy across various corruptions, excluding the original images. The other terms in the X-asix denote the types of image corruptions.

Model	Train		Test		Clean Acc. (%)	Robust Acc. (%)	Avg. Acc. (%)	Attacks
	CIFAR10	ImageNet	CIFAR10	ImageNet				
ResNet50	-	v	-	v	80.85	0	40.42	$\epsilon_\infty = 4/255$
+ MAEP	v	-	-	v	77.84	70.62	**74.23**	$\epsilon_\infty = 4/255$
+ MAEP	v	-	-	v	77.62	66.19	**71.91**	$\epsilon_\infty = \mathbf{8/255}$
+ DISCO [14]	v	-	-	v	76.61	69.12	72.86	$\epsilon_\infty = 4/255$
+ MAEP	-	v	-	v	77.97	73.94	**75.96**	$\epsilon_\infty = 4/255$
+ DISCO	-	v	-	v	77.54	70.44	73.99	$\epsilon_\infty = 4/255$
+ DiffPure* [25]	-	v	-	v	70.01	67.11	68.60	$\epsilon_\infty = 4/255$
+ ScoreOpt* [42]	-	v	-	v	70.07	66.02	68.05	$\epsilon_\infty = 4/255$
+ DDA [9]	-	v	-	v	77.92	1.37	39.65	$\epsilon_\infty = 4/255$

Table 9: Transferability of adversarial defenses (DISCO and MAEP) from CIFAR10 to ImageNet (trained on CIFAR10 but tested on ImageNet) under ResNet50 (pre-trained model weight is from official PyTorch) and AutoAttack[5]. sterisk (*) indicates that the results were excerpted from the papers.

6 Conclusion

Although diffusion models have demonstrated strong capabilities as adversarial purifiers in prior studies, their limitations remain underexplored. In this paper, we reveal that diffusion-based purification can impair classifier generalization, particularly in scenarios involving color-related variations. Moreover, we explore the generalization loss in classifiers caused by diffusion models and propose Masked AutoEncoder Purifier (MAEP), incorporating masked autoencoder (MAE) and purification loss, as a non-diffusion-based purifier.

7 Acknowledgement

This work was supported by the National Science and Technology Council (NSTC) with Grants NSTC 112-2221-E-001-011-MY2 and 114-2221-E-001 -010 -MY2.

References

1. Alfarra, M., Pérez, J.C., Thabet, A., Bibi, A., Torr, P.H., Ghanem, B.: Combating adversaries with anti-adversaries. In: AAAI. vol. 36, pp. 5992–6000 (2022)
2. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: ICML. pp. 274–283. PMLR (2018)
3. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: CVPR. pp. 5939–5948 (2019)
4. Croce, F., Hein, M.: Minimally distorted adversarial examples with a fast adaptive boundary attack. In: ICML. pp. 2196–2205. PMLR (2020)
5. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: ICML. pp. 2206–2216. PMLR (2020)
6. Cui, J., Tian, Z., Zhong, Z., Qi, X., Yu, B., Zhang, H.: Decoupled kullback-leibler divergence loss. arXiv preprint arXiv:2305.13948 (2023)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255. Ieee (2009)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2020)
9. Gao, J., Zhang, J., Liu, X., Darrell, T., Shelhamer, E., Wang, D.: Back to the source: Diffusion-driven adaptation to test-time corruption. In: CVPR. pp. 11786–11796 (2023)
10. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. stat **1050**, 20 (2015)
11. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: CVPR. pp. 16000–16009 (2022)
12. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261 (2019)
13. Hill, M., Mitchell, J.C., Zhu, S.C.: Stochastic security: Adversarial defense using long-run dynamics of energy-based models. In: ICLR (2020)
14. Ho, C.H., Vasconcelos, N.: Disco: Adversarial defense with local implicit functions. NIPS **35**, 23818–23837 (2022)
15. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NIPS **33**, 6840–6851 (2020)
16. Hsiung, L., Tsai, Y.Y., Chen, P.Y., Ho, T.Y.: Towards compositional adversarial robustness: Generalizing adversarial training to composite semantic perturbations. In: CVPR. pp. 24658–24667 (2023)
17. Huang, B., Chen, M., Wang, Y., Lu, J., Cheng, M., Wang, W.: Boosting accuracy and robustness of student models via adaptive adversarial distillation. In: CVPR. pp. 24668–24677 (2023)
18. Huang, Q., Dong, X., Chen, D., Chen, Y., Yuan, L., Hua, G., Zhang, W., Yu, N.: Improving adversarial robustness of masked autoencoders via test-time frequency-domain prompting. In: ICCV. pp. 1600–1610 (2023)
19. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT. pp. 4171–4186 (2019)
20. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
21. Krizhevsky, A., et al.: Learning multiple layers of features from tiny images (2009)

22. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: Artificial intelligence safety and security, pp. 99–112. Chapman and Hall/CRC (2018)
23. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: CVPR workshops. pp. 136–144 (2017)
24. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018)
25. Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., Anandkumar, A.: Diffusion models for adversarial purification. In: ICML. pp. 16805–16827. PMLR (2022)
26. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763. PMLR (2021)
27. Rebuffi, S.A., Gowal, S., Calian, D.A., Stimberg, F., Wiles, O., Mann, T.A.: Data augmentation can improve robustness. NIPS **34**, 29935–29948 (2021)
28. Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. IEEE Computer graphics and applications **21**(5), 34–41 (2001)
29. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
30. Shafahi, A., Najibi, M., Ghiasi, M.A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! NIPS **32** (2019)
31. Shi, C., Holtz, C., Mishne, G.: Online adversarial purification based on self-supervision. arXiv preprint arXiv:2101.09387 (2021)
32. Somepalli, G., Singla, V., Goldblum, M., Geiping, J., Goldstein, T.: Diffusion art or digital forgery? investigating data replication in diffusion models. In: CVPR. pp. 6048–6058 (2023)
33. Song, K., Lai, H., Pan, Y., Yin, J.: Mimicdiffusion: Purifying adversarial perturbation via mimicking clean diffusion model. In: CVPR. pp. 24665–24674 (2024)
34. Tsai, Y.Y., Chao, J.C., Wen, A., Yang, Z., Mao, C., Shah, T., Yang, J.: Test-time detection and repair of adversarial samples via masked autoencoder. In: CVPR Workshops (2023)
35. Wang, Z., Pang, T., Du, C., Lin, M., Liu, W., Yan, S.: Better diffusion models further improve adversarial training. In: ICML. pp. 36246–36263. PMLR (2023)
36. Wu, D., Xia, S.T., Wang, Y.: Adversarial weight perturbation helps robust generalization. NIPS **33**, 2958–2969 (2020)
37. Wu, Q., Ye, H., Gu, Y., Zhang, H., Wang, L., He, D.: Denoising masked autoencoders help robust classification. In: ICLR (2023)
38. Yoon, J., Hwang, S.J., Lee, J.: Adversarial purification with score-based generative models. In: ICML. pp. 12062–12072. PMLR (2021)
39. You, Z., Liu, D., Han, B., Xu, C.: Beyond pretrained features: Noisy image modeling provides adversarial defense. NIPS **36** (2023)
40. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: British Machine Vision Conference 2016. British Machine Vision Association (2016)
41. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2017)
42. Zhang, B., Luo, W., Zhang, Z.: Enhancing adversarial robustness via score-based optimization. NIPS **36** (2023)
43. Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., Jordan, M.: Theoretically principled trade-off between robustness and accuracy. In: ICML. pp. 7472–7482. PMLR (2019)

Appendix

8 More Objective Function Designs

1) Reconstruction Loss. Intuitively, to enhance both the clean accuracy and robust accuracy of an NN model, the purified/reconstructed image should closely resemble its clean version by minimizing the loss as:

$$L = D(\mathcal{P}(x_a), x) + D(\mathcal{P}(x), x), \tag{16}$$

where the first term denotes the distance between the purified adversarial image and its corresponding clean image, and the second term measures the distance between the purified clean image and true clean one.

2)Masked Language Modeling (MLM). We leverage MLM directly to train the purifier through a two-step process. Initially, it aims to learn adversarial embeddings during the pre-training stage, as indicated in Eq. (4), and subsequently finetunes to purify an adversarial image. The loss function is described as follows with respect to the two-step process:

$$L_{pre-train} = L_{MAE}(x_a, x) = \|(\mathbf{1} - M) \odot x - (\mathbf{1} - M) \odot (g \circ f(M \odot \ x_a))\|. \tag{17}$$

$$L_{finetune} = D(\mathcal{P}_{MAE}(x_a), x). \tag{18}$$

3) TRADES [43]. TRADES proposed to train a robust classifier with the loss function defined as:

$$L = CrossEntropy(c(x), y) + KL(c(x), c(x_a))/\lambda, \tag{19}$$

where the first term maintains the clean accuracy while the second term focuses on improving the robust accuracy by making logits of adversarial sample similar to those of clean acccuracy, and c is a classifier.

3.1) TRADES in pixel domain. To replicate the success of TRADES in adversarial training, we adapt its concept from training a robust classifier to training an adversarial purifier. The main difference is that the purifier needs to process the image instead of class prediction. Therefore, we replace the cross-entropy loss and KL divergence loss in Eq. (19) with a reconstruction loss in the image pixel domain to meet the purifier's requirement as:

$$L = D(\mathcal{P}(x), x) + D(\mathcal{P}(x), \mathcal{P}(x_a))/\lambda, \tag{20}$$

where the first term maintains the purified clean image quality and the second term tries to purify the adversarial image by mimicking the clean image in a sense similar to KL divergence loss in Eq. (19).

3.2) TRADES in latent domain. Unlike the methods proposed to concentrate on the image pixel domain, several works, such as Latent Diffusion

[29] and CLIP [26], have achieved notable success by processing image latent representations. In our approach, as indicated in Eq. (21) below, we maintain clean image purification (1st term) while enforcing constraints on adversarial perturbations within the latent space (2nd term) as:

$$L = D(\mathcal{P}(x), x) + D(f(x), f(x_a))/\lambda. \tag{21}$$

In Table 10, we provide a comparison of the performance of all loss designs discussed here to verify the design of MAEP. We have the following observations: (1) Although reconstruction loss concurrently learns the reconstruction of both clean and adversarial images, its performance falls short of DISCO, which concentrates solely on reconstructing adversarial images. (2) MLM and DISCO are closely associated with MAEP. Directly applying MLM appears ineffective, while MAEP demonstrates performance enhancement over DISCO. (3) Exploiting the concept of TRADES does not aid in learning an adversarial purifier. Thus, our validation shows that MAEP significantly outperforms other approaches.

Defenses	Clean Acc. (%)	Robust Acc. (%)	Avg. Acc.
WRN28-10	94.78	0	47.39
+ DISCO [14]	89.26	85.33	87.29
+ Reconstruction (Eq. (16))	94.74	82.73	88.73
+ TRADES (pixel, Eq. (20))	94.75	0.85	47.81
+ TRADES (latent, Eq. (21))	94.64	38.16	66.40
+ MLM [11]	92.85	61.46	77.15
+ MAEP	92.30	88.73	**90.52**

Table 10: Performance comparison of different objective functions on CIFAR10 dataset under AutoAttack with attack budget $\epsilon_\infty = 8/255$.