

LoRID: Low-Rank Iterative Diffusion for Adversarial Purification

Geigh Zollicoffer^{1*}, Minh Vu^{1*}, Ben Nebgen¹, Juan Castorena², Boian Alexandrov¹, Manish Bhattarai¹

¹ Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM

² Computational Sciences, Los Alamos National Laboratory, Los Alamos, NM

gzollicoffer@lanl.gov, mvu@lanl.gov, bnebg@lanl.gov, jcastorena@lanl.gov, boian@lanl.gov, ceodsppectrum@lanl.gov

Abstract

This work presents an information-theoretic examination of diffusion-based purification methods, the state-of-the-art adversarial defenses that utilize diffusion models to remove malicious perturbations in adversarial examples. By theoretically characterizing the inherent purification errors associated with the Markov-based diffusion purifications, we introduce LoRID, a novel Low-Rank Iterative Diffusion purification method designed to remove adversarial perturbation with low intrinsic purification errors. LoRID centers around a multi-stage purification process that leverages multiple rounds of diffusion-denoising loops at the early time-steps of the diffusion models, and the integration of Tucker decomposition, an extension of matrix factorization, to remove adversarial noise at high-noise regimes. Consequently, LoRID increases the effective diffusion time-steps and overcomes strong adversarial attacks, achieving superior robustness performance in CIFAR-10/100, CelebA-HQ, and ImageNet datasets under both white-box and black-box settings.

1 Introduction

Despite their widespread adoption, neural networks are vulnerable to small malicious input perturbations, leading to unpredictable outputs, known as *adversarial attacks* (Szegedy et al. 2014; Goodfellow, Shlens, and Szegedy 2015). Various defense methods have been developed to protect these models (Qiu et al. 2019), including *adversarial training* (Madry et al. 2019; Bai et al. 2021; Zhang et al. 2019) and *adversarial purification* (Salakhutdinov 2015; Shi, Holtz, and Mishne 2021; Song et al. 2018; Nie et al. 2022; Wang et al. 2022, 2023). With the introduction of *diffusion models* (Ho, Jain, and Abbeel 2020; Song et al. 2021) as a powerful class of generative models, diffusion-based adversarial purifications have overcome training-based methods and achieve state-of-the-art (SOTA) robustness performance (Blau et al. 2022; Wang et al. 2022; Nie et al. 2022; Xiao et al. 2022). In principle, the diffusion-based purification first diffuses the adversarial inputs with Gaussian noises in t time-steps and utilizes the diffusion’s denoiser to remove the adversarial perturbations along with the added Gaussian noises. While it is computationally challenging to attack diffusion-based purification due to vanishing/exploding gradient problems, high memory costs, and substantial randomness (Kang, Song, and

Table 1: Performance of SOTA score-based purification versus our proposed LoRID, a Markov-based purification, in CIFAR-10 ($\epsilon = 8/255$) and ImageNet ($\epsilon = 4/255$) under L_∞ white-box AutoAttack in WideResNet-28-10.

Purification	Score-based	LoRID
Standard Acc	89.02 / 71.16	84.20 / 73.98
Robust Acc	46.88 / 44.39	54.14 / 56.54
Inference Run-time Speedup	$\times 1 / \times 1$	$\times 2.3 / \times 4.6$

Li 2024), recent work has been proposing efficient attacks against diffusion-based purification (Nie et al. 2022; Kang, Song, and Li 2024), which can degrade the model robustness significantly. A naive way to prevent such attacks is to increase the diffusion time-step t as it will remarkably raise both the time and memory complexity for the attackers (Kang, Song, and Li 2024). However, increasing t would not only introduces additional computational cost of purification (Nie et al. 2022; Lee and Kim 2023), but also inevitably damages the purified samples (see Theorem 2 or Fig. 3), and significantly degrade the classification accuracy.

Our work aims to develop a more robust and efficient diffusion-based purification method to counter emerging adversarial attacks. We first introduce an information-theoretic viewpoint on the diffusion-based purification process, in which the purified signal is considered as the recovered signal from a noisy communication channel. Different from the previous purification (Nie et al. 2022) centered on the Score-based diffusion (Song et al. 2021), our work is the first theoretical analysis of the inherent error induced by Markov-based purifications (Blau et al. 2022; Wang et al. 2022; Xiao et al. 2022), which are purifications relying on the Denoising Diffusion Probabilistic Model (DDPM) (Ho, Jain, and Abbeel 2020). Our theoretical foundation for DDPM (**Theorem 1, 2 and 3**) are essential as they validate the usage of DDPM for purification and leverage its substantial advantage in terms of running time compared to the Score-based (as shown in Table 1). Our analysis further points out an interesting finding: the purification error (Corollary 1) can be reduced significantly by conducting multiple iterations at the early time-steps of the DDPM (**Theorem 4**). Particularly, the application of a single purification with a time-step t is theoretically shown to be less beneficial than the looping of L iterations of diffusion-denoising with a time-step of t/L

*These authors contributed equally.

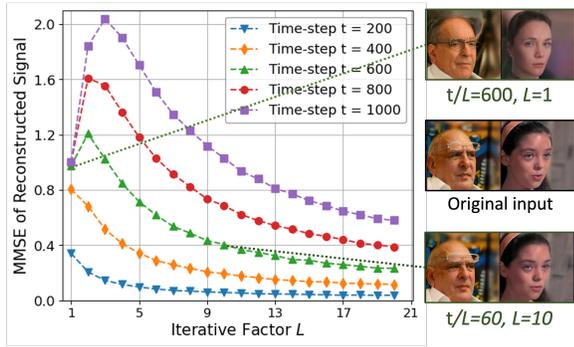


Figure 1: The MMSEs induced by Markov-based purification against the iterative factor L (Corollary 1): each point is the MMSE of the reconstructed data from a normalized Gaussian through L iterative loops of t/L diffusion-denosing calls. Thus, points on a line share the same effective denoising step $t = (t/L) \times L$. The key observation is the purification error generally decreases as L increases. The right samples compare clean samples, purified samples with a single large time-step $t/L = 600$, and those with the same effective denoising step t but with a larger iterative factor $L = 10$ (Details in Appx. B.1).

(Fig. 1). Our study additionally suggests the usage of *Tucker decomposition* (Bergqvist and Larsson 2010), a higher-order extension of matrix factorization, to attenuate adversarial noise at the high-noise regime (**Theorem 5**). We realize the advantages of those findings and propose LoRID, a Low-Rank Iterative Diffusion purification method designed to mitigate the purification errors (Fig. 2). By controlling the purification error, LoRID can effectively increase the diffusion time-step and beat the SOTA robustness benchmark, in both white-box and black-box settings (Table. 1 highlights LoRID’s performance in CIFAR-10 (Rabanser, Shchur, and Gunnemann 2017), and Imagenet (Deng et al. 2009)). The main contributions of this work are:

- We establish theoretical bounds on the purification errors of Markov-based purifications. In particular, Theorem 1 show that the adversarial noise will be removed at a distribution-level as the purification time-steps increases. On the other hand, Theorem 2, and 3 point out the purification at the sample-level.
- We show theoretical justifications for looping the early-stages of DDPM (Theorem 4), and the usage of Tucker decomposition (Theorem 5) for adversarial purification.
- We introduce a Markov-based purification algorithm, called LoRID (Alg. 1), utilizing early looping and Tucker decomposition and demonstrate rigorously its effectiveness and high performance in three real-world datasets: CIFAR-10/100, CelebA-HQ, and Imagenet.

Our paper is organized as follows. Sect. 2 provides the background and related work of this study. Sect. 3 consists of our theoretical analysis and the description of our proposed purification LoRID. Sect. 4 provides our experimental results, and Sect. 5 concludes this paper.

2 Background and Related Work

This section first briefly reviews the Denoising Diffusion Probabilistic Model (Ho, Jain, and Abbeel 2020), which is the backbone of our diffusion purifications. Then, the related work about the usage of diffusion models as adversarial purifiers is discussed. Finally, we briefly discuss the Tucker decomposition, which is a component utilized by our method.

Denoising Diffusion Probabilistic Models (DDPMs) are a class of generative models that, during training, iteratively adding noise to input signals, then learning to denoise from the resulting noisy signal. Formally, given a data point \mathbf{x}_0 sampled from the data distribution $q(\mathbf{x}_0)$, a *forward diffusion process* from clean data \mathbf{x}_0 to \mathbf{x}_T is a Markov-chain that gradually adds Gaussian noise, denoted by \mathcal{N} , to the data according to a variance schedule $\{\beta_t \in (0, 1)\}_{t=1}^T$: $(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})$, where

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad (1)$$

The objective of DDPM is to learn the joint distribution $p_\theta(\mathbf{x}_{0:T})$, called the reverse process, which is defined as another Markov-chain with learned Gaussian transitions $p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, where

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \quad (2)$$

starting with $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$. The mean $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ is a neural network parameterized by θ , and the variance $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ can be either time-step dependent constants (Ho, Jain, and Abbeel 2020) or learned by a neural network (Nichol and Dhariwal 2021). A notable property of the forward process is that it admits sampling \mathbf{x}_t at an arbitrary time-step t in closed form: using the notation $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$, we have

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I})$$

Using the reparameterize trick, we can define the forward diffusion process to the time-step t as f_t :

$$\mathbf{x}_t = f_t(\mathbf{x}_0) := \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}_0 \quad (3)$$

where $\boldsymbol{\epsilon}_0$ is a standard Gaussian noise.

For the reverse process, the recovered signal from the time-step t can be written as (Ho, Jain, and Abbeel 2020):

$$\tilde{\mathbf{x}}_0(t) = \frac{1}{\sqrt{\bar{\alpha}_t}}\mathbf{x}_t - \frac{\sqrt{1 - \bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \quad (4)$$

where $\boldsymbol{\epsilon}_\theta$ is a function approximator predicting $\boldsymbol{\epsilon}$ from \mathbf{x}_t , i.e., the *noise matching term*. Given that, we have

$$\tilde{\mathbf{x}}_0(t) - \mathbf{x}_0 = \frac{\sqrt{1 - \bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{\epsilon}_0 - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)) \quad (5)$$

Thus, the approximator $\boldsymbol{\epsilon}_\theta$ can be trained using MSE loss:

$$L(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2 \right] \quad (6)$$

Diffusion models as adversarial purifiers. Diffusion-based purification schemes can be categorized into Markov-based purification (or DDPM-based), and Score-based purification, which utilize DDPM (Ho, Jain, and Abbeel 2020)

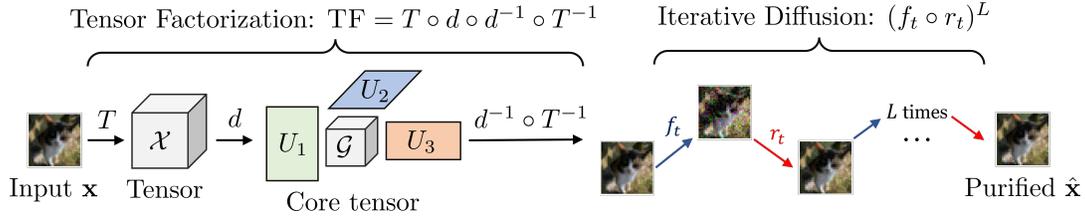


Figure 2: The overall purification process of LoRID: given an input image \mathbf{x} , LoRID first transforms the image to a tensor and conducts tensor factorization to eliminate some adversarial perturbation. Then, multiple loops of diffusion-denoising, denoted by f_t and r_t , at the early stages of the diffusion models are applied to obtain the final purified image $\hat{\mathbf{x}}$.

and Score-based diffusion model (Song et al. 2021) to purify the adversarial examples, respectively. In this work, we focus on the Markov-based methods, which typically diffuse the adversarial input \mathbf{x}_a to a certain time-step t , then utilize the DDPMs to iteratively solve the reverse process as given in (Ho, Jain, and Abbeel 2020):

$$\hat{\mathbf{x}}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\hat{\mathbf{x}}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_{\theta}(\mathbf{x}_t, t) \right) + \frac{\beta_t(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \epsilon \quad (7)$$

We denote the process of running (7) iteratively, starting from $\hat{\mathbf{x}}_t := \mathbf{x}_t$ to finally obtain $\hat{\mathbf{x}}_0$ by r_t and write $\hat{\mathbf{x}}_0 = r_t(\mathbf{x}_t)$. The whole process of purification is, therefore, can be referred by the composition $r_t \circ f_t$.

Recent Markov-based purifications often apply a modified version of (7). The work (Blau et al. 2022) uses a rescaled version of (7) with a larger noise term for purification. The Guided-DDPM purification (Wang et al. 2022) introduces a guided term encouraging the purified image to be close to the adversarial image in the reverse process to protect the sample’s semantics. On the other hand, DensePure (Xiao et al. 2022) computes multiple reversed samples using (Eq. 7) and determines final predictions by majority voting. We find that the recent findings of Lee and Kim (2023) are the most closely related to this work: they observe that the *gradual noise-scheduling strategy*, which involves looping several times during the early stages of the DDPM, can enhance defense mechanisms. However, there is no theoretical justification for this strategy. Furthermore, as the attackers in their threat models are unaware of this defense, it is unclear whether gradual noise-scheduling truly offer better robustness against practical white-box attackers.

Despite the differences, all methods emphasize a too-large t would damage the global label semantics from the purified sample. While the theoretical statement for this is stated in the case of Score-based purifications (Nie et al. 2022), the counterpart for Markov-based is lacking. One of our contributions is the theoretical statement for the Markov-based in Theorem 1 and Theorem 2. Another aspect that distinguishes the technicality our method, LoRID, from previous work is the use of a large number of loops (typically between 10 and 40) in the early stages (about 1% – 10% of the total time-step) of the DDPM, along with the implementation of Tucker decomposition.

Tucker Decomposition, also known as higher-order singular value decomposition (HOSVD) (Bergqvist and Larsson 2010), is a mathematical technique used in multilinear

algebra and data analysis, and can be viewed as an extension of the concept of singular value decomposition (SVD) for higher-dimensional data arrays or tensors (Kolda and Bader 2009). Computing the Tucker decomposition of a tensor can encode the essential information and structure of the tensor into a set of core tensor and factor matrices. It’s widely used in various fields such as signal processing, image processing, neuroscience, data compression, and in the line of the proposed work, feature extraction from high-dimensional data (Kolda and Bader 2009).

In our purification context, as the latent tensor \mathcal{X} is obtained from an original image \mathbf{x} via a tensorization process (Bhattarai et al. 2023), denoted by $\mathcal{X} := T(\mathbf{x})$, the overall tensor-factorization denoising process (Fig. 2) can be referred by the following:

$$\hat{\mathbf{x}} = \text{TF}(\mathbf{x} + \epsilon) \quad (8)$$

where T^{-1} denotes the recover of the image from the latent space, and TF is defined as $\text{TF} := T^{-1} \circ d^{-1} \circ d \circ T$. The details of this operation is provided in Appx. B.2.

As both Tucker and tensorization are linear transformations, the denoising-reconstruction error can be bounded as:

$$\|\mathbf{x} - \text{TF}(\mathbf{x} + \epsilon)\| \leq \|\mathbf{x} - \text{TF}(\mathbf{x})\| + \|\text{TF}(\epsilon)\|. \quad (9)$$

Here, the first term, denoted as $E_{\text{TUCKER}} := \|\mathbf{x} - \text{TF}(\mathbf{x})\|$ represents the error introduced by not capturing the full variance in each mode of the data through the Tucker decomposition. The second term $\|\text{TF}(\epsilon)\|$ represents the error caused by the original noise on \mathbf{x} remained after the denoising process. E_{TUCKER} can be bounded further by ((De Lathauwer, De Moor, and Vandewalle 2000) Property 10; (Hackbusch 2012) Theorem 10.2): $E_{\text{TUCKER}} \leq \sum_{n=1}^N \sum_{i_n=r_n+1}^{I_n} (\sigma_{i_n}^{(n)})^2$, where $\{\sigma_{i_n}^{(n)}\}_{i_n=1}^{I_n}$ is the singular values of the mode- n unfolding of the tensor \mathcal{X} .

3 Method

This section provides theoretical results on different aspects of Markov-based purification (7) and the details for our proposed adversarial purification algorithm LoRID.

- Subsect. 3.1 provides Theorem 1 about the theoretical removal of the adversarial noise as the diffusion time-step t in the Markov-based diffusion model increases at the distribution-level. It is the counterpart of Theorem 3.1 in (Nie et al. 2022) for Score-based purification.

- The purification error between the clean and the purified images at the sample-level are further characterized in Theorem 2 and 3 in Subsect. 3.1. While Theorem 3 can be viewed as an adaptation of Theorem 3.2 from Score-based to Markov-based purification, to the best of our knowledge, the lower bound on the reconstruction error in Theorem 2 has not been previously established for any diffusion-based purification methods.
- Subsect. 3.2 demonstrates how we realize our theoretical analysis into practical measurement. Particularly, We analyze the intrinsic purification error arising from the Markov-based purification process (Corollary 1) and identify the advantage of looping the early time-steps of the diffusion models for the purification task (Theorem 4). The result suggests that, with the same effective diffusion-denoising steps, looping can reduce the intrinsic purification error significantly.
- Subsect. 3.2 also studies and validates the usage of Tucker Decomposition combined with Markov-based purification at the high-noise regime (Theorem 5).
- Based on the theoretical analysis, we design LoRID, the Low-Rank Iterative Diffusion method to purify adversarial noise. Its description is provided in Subsect. 3.3.

3.1 Markov-based Purification

Intuitively, the diffusion time-step t need be large enough to remove adversarial perturbations; however, the image’s semantics will also be removed as t increases. That observation is captured in the following Theorem 1, which states that the KL-divergence between the distributions of the clean images and the adversarial images converges as t increases:

Theorem 1. Let $\{\mathbf{x}_t^{(i)}\}_{t \in \{0, \dots, T\}}$, $i \in \{1, 2\}$ be two diffusion processes given by the forward equation (1) of a DDPM. Denote $q_t^{(1)}$ and $q_t^{(2)}$ the distributions of $\mathbf{x}_t^{(1)}$ and $\mathbf{x}_t^{(2)}$, respectively. Then, for all $t \in \{0, \dots, T-1\}$, we have

$$D_{KL}(q_t^{(1)} || q_t^{(2)}) \geq D_{KL}(q_{t+1}^{(1)} || q_{t+1}^{(2)})$$

Sketch of proof (proof in Appx A.1). While Theorem 1 resembles that stated for the Score-based purification (Nie et al. 2022), its proof is greatly different since the DDPM’s diffusion is not controlled by an Stochastic Differential Equation. Instead, we leverage the underlying Markov process governing the forward diffusion of DDPM (1), and show $D_{KL}(q_{t+1}^{(1)} || q_{t+1}^{(2)}) + D_{KL}(q^{(1)}(\mathbf{x}_t | \mathbf{x}_{t+1}) || q^{(2)}(\mathbf{x}_t | \mathbf{x}_{t+1})) = D_{KL}(q_t^{(1)} || q_t^{(2)})$ by expanding the KL-divergence between $q^{(1)}(\mathbf{x}_{t+1}, \mathbf{x}_t)$ and $q^{(2)}(\mathbf{x}_{t+1}, \mathbf{x}_t)$. Then, due to the non-negativity of the KL-divergence, we have the Theorem.

Note that Theorem 1 captures the purification at the distribution level. Similar to the Score-based purification (Nie et al. 2022), we are also interested in the purification of the DDPM at the instance level. In fact, the variational bound (Eq. 6) suggests that the reconstruction error $\|\hat{\mathbf{x}}_0(t) - \mathbf{x}_0\|$ is directly proportional to the DDPM’s training objective.

However, that objective, $L(\theta)$, is for all time-steps, while the purification error only depends on the one time-step, at which, the reverse process is applied to recover $\hat{\mathbf{x}}_0(t)$. Intuitively, as t increases, the argument of the approximator $\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$ contains less information about the noise ϵ , thus, results in a higher error. The following two Theorems formalize that intuition:

Theorem 2. Let $\{\mathbf{x}_t\}_{t \in \{0, \dots, T\}}$ be a diffusion process defined by the forward equation (1) where \mathbf{x}_0 is the adversarial sample. i.e, $\mathbf{x}_0 = \mathbf{x}_{clean} + \epsilon_a$. For any time t , we have

$$\mathbb{E} [\|\hat{\mathbf{x}}_0(t) - \mathbf{x}_{clean}\|] \geq \text{MMSE} \left(\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \right) - \|\epsilon_a\| \quad (10)$$

where the expectation is taken over the distribution of \mathbf{x}_{clean} and $\text{MMSE}(\text{SNR})$ is the minimum mean-square error achievable by optimal estimation of the input given the output of Gaussian channel with a signal-to-noise ratio of SNR. The function $\text{MMSE}(\text{SNR})$ has the following form (Guo, Shamai, and Verdu 2005):

$$1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-y^2/2} \tanh(\text{SNR} - \sqrt{\text{SNR}} y) dy. \quad (11)$$

Theorem 3. Additionally to the conditions stated in Theorem 2, if the DDPM is able to recover the original signal \mathbf{x}_0 within an error $\delta_{\text{DDPM}}(t)$ in the expectation, i.e., for all t ,

$$\mathbb{E} [\|\hat{\mathbf{x}}_0(t) - \mathbf{x}_0\|] \leq \mathbb{E} [\|\hat{\mathbf{x}}_0^*(\mathbf{y}_t) - \mathbf{x}_0\|] + \delta_{\text{DDPM}}(t) \quad (12)$$

where $\hat{\mathbf{x}}_0^*(\mathbf{y}_t)$ is the best estimator of \mathbf{x}_0 given $\mathbf{y}_t = (\sqrt{\bar{\alpha}_t}/\sqrt{1 - \bar{\alpha}_t})\mathbf{x}_0 + \epsilon_0$, then, we have the reconstructed error $\mathbb{E} [\|\hat{\mathbf{x}}_0(t) - \mathbf{x}_{clean}\|]$ is upper-bounded by:

$$\text{MMSE} \left(\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \right) + \delta_{\text{DDPM}}(t) + \|\epsilon_a\| \quad (13)$$

Sketch of proofs (proofs in Appx A.2 and A.3). The proofs of both theorems consider the forwarding diffusion of the DDPM as a Gaussian channel, and the purification task is equivalent to the reconstruction of the channel’s input. Given a purification time-step t , i.e., the time-step we decide to start the denoising/purification process, the equivalent Gaussian channel would have an effective signal-to-noise (SNR) ratio of $\bar{\alpha}_t/(1 - \bar{\alpha}_t)$. Intuitively, the higher the time-step, the smaller the value of $\bar{\alpha}_t/(1 - \bar{\alpha}_t)$, and, even with an optimal denoiser, the more inherent error are introduced to the purified sample. In fact, the expression $\text{MMSE}(\bar{\alpha}_t/(1 - \bar{\alpha}_t))$ appearing in both theorems capture that intrinsic error. Unfortunately, there is currently no closed-form for that expression. We follow previous work studying noisy Gaussian channel (Guo, Shamai, and Verdu 2005) and provide its integral form in expression (11).

Remark. Regarding $\delta_{\text{DDPM}}(t)$, it captures how well the trained-DDPM can recover the input given its noisy signal at time-step t . The assumption that $\delta_{\text{DDPM}}(t)$ bounds the reconstruction error (12) is a weaker version of the assumption made by (Song et al. 2021) in the analysis of the Score-based diffusion, which is also utilized to upper-bound the error induced by Score-based purification (Ho, Jain, and Abbeel 2020). In fact, both works assume the Score-based diffusion model can **perfectly** learn the score function $\nabla_x \log p(\mathbf{x}_0)$ to establish their theoretical results.

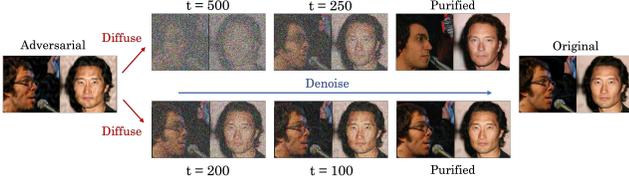


Figure 3: Illustration of adversarial purification using DDPM. The adversarial samples (left) is purified from the time-step $t = 200$ (bottom) and $t = 500$ (top) to recover the original samples (right). The middles show $\hat{\mathbf{x}}_t$ (Equation (7)) obtained by iteratively denoising to the indicated intermediate time-steps. The top purification with a too large time-step induces unavoidable error (Theorem 2).

To conclude this subsection, we illustrate the impact of the time-step t on the purification process based on DDPM in Fig. 3. When t is chosen appropriately, all the terms on the right-hand-side of (13) of Theorem 3 are controlled, which enforces a small difference between the clean input and and recovered signal. This means not only the adversarial noises are removed but also the purified images maintain the semantic of the original data. This is reflected in the purified images at the bottom of Fig. 3) However, when t is too large as illustrated at the top of Fig. 3, the diffusion-based purification induces an intrinsic error reflected in the term $\text{MMSE}(\bar{\alpha}_t/(1 - \bar{\alpha}_t))$ of Theorem 2. This error makes the purified images inevitably different from the original signal.

3.2 Controlling Purification Error

This subsection studies the inherent error introduced by the purification process and demonstrates why it instigates a better purification scheme based on looping the early stage of the DDPM and the utilization of Tucker decomposition.

Our analysis starts with the consideration of the trivial case in which there is no adversarial noise. By combining the two Theorems 2 and 3, we have the following corollary:

Corollary 1. *Given the assumptions in Theorem 3, the intrinsic purification error on a clean purification input $\mathbf{x}_0 = \mathbf{x}_{clean}$, i.e., $\epsilon_a = 0$, is bounded by*

$$\begin{aligned} \text{MMSE} \left(\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \right) &\leq \mathbb{E} [\|\hat{\mathbf{x}}_0(t) - \mathbf{x}_{clean}\|] \\ &\leq \text{MMSE} \left(\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \right) + \delta_{\text{DDPM}}(t) \end{aligned} \quad (14)$$

The corollary reflects the strong connection between the purification error and the MMSE term. Especially, when the DDPM is well-trained, the gap $\delta_{\text{DDPM}}(t)$ between the lower and upper bounds becomes small, and $\mathbb{E} [\|\hat{\mathbf{x}}_0(t) - \mathbf{x}_{clean}\|]$ becomes more similar to $\text{MMSE}(\bar{\alpha}_t/(1 - \bar{\alpha}_t))$. This observation motivates us to investigate purification schemes that minimize the MMSE.

Looping at early time-steps. Several recent work observed that repetitive usage of the diffusion-denoising steps in parallel (Wang et al. 2022; Nie et al. 2022) or sequential (Lee and Kim 2023) can enhance system robustness against

adversarial attacks. However, too many diffusion-denoising calls would not only diminish robustness gain but also degrade the clean accuracy significantly. Hence, we tackle the following question: *Given a fixed number of denoiser’s call, i.e., total number of diffusion-denoising steps, for the sake of adversarial purification, should we diffusion-denoising multiple loops of the DDPM at the earlier time-steps or utilize a few loops with large time-steps?*

We now provide theoretical justification for the usage of multiple loops in purification. Specifically, we want to compare the purification to the time-step t , i.e., denoted by $r_t \circ f_t$, and the purification of L loops to the time-step t/L , i.e., $(r_{t/L} \circ f_{t/L})^L$. By denoting the output of l times DDPM-purification to time-step t , $\hat{\mathbf{x}}_0^l(t) := (r_{t/L} \circ f_{t/L})^l(\mathbf{x}_0)$, we formalize the impact of looping purification via the following Theorem 4:

Theorem 4. *Let $\{\mathbf{x}_t\}_{t=0}^T$ be a diffusion process defined by the forward (1) where \mathbf{x}_0 is the adversarial sample. i.e., $\mathbf{x}_0 = \mathbf{x}_{clean} + \epsilon_a$. For any given time t , we have the reconstructed error $\mathbb{E} [\|\hat{\mathbf{x}}_0^L(t/L) - \mathbf{x}_{clean}\|]$ is upper-bounded by:*

$$L \times \left(\text{MMSE} \left(\frac{\bar{\alpha}_{t/L}}{1 - \bar{\alpha}_{t/L}} \right) + \delta_{\text{DDPM}} \left(\frac{t}{L} \right) \right) + \|\epsilon_a\| \quad (15)$$

where the expectation is taken over the distribution of \mathbf{x}_{clean} (Proof in Appx. A.4).

Note that the upper-bound on the reconstruction error of $(r_{t/L} \circ f_{t/L})^L$ is controlled by $L \times \text{MMSE}(\bar{\alpha}_{t/L}/(1 - \bar{\alpha}_{t/L}))$, instead of $\text{MMSE}(\bar{\alpha}_t/(1 - \bar{\alpha}_t))$ as in the vanilla purification scheme $r_t \circ f_t$. For an illustration of the impact of looping the diffusion-denoising, we consider the input to compute the MMSE as standard Gaussian. The MMSE is then given by $\text{MMSE}(\text{SNR}) = 1/(1 + \text{SNR})$ (instead of the integral form (11)). We further take the values of $\bar{\alpha}_t$ in DDPM (Ho, Jain, and Abbeel 2020) and plot $L \times \text{MMSE}(\bar{\alpha}_{t/L}/(1 - \bar{\alpha}_{t/L}))$ as a function of L in Fig. 1. The result shows that purification at a small time-step with a large number of iteration is greatly beneficial for the purification error.

Tucker Decomposition for High-noise Regime. We now study the utilization of DDPM and Tucker Decomposition to purify the adversarial samples, which is characterized by the operations $r_t \circ f_t$ and $\text{TF} = T^{-1} \circ d^{-1} \circ d \circ T$. From the previous analysis, the reconstruction error induced by the two methods are bounded by:

$$\text{MSE}_{r_t \circ f_t}(\epsilon_a) \leq \text{MMSE} \left(\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \right) + \delta_{\text{DDPM}}(t) + \|\epsilon_a\| \quad (16)$$

$$\text{MSE}_{\text{TF}}(\epsilon_a) \leq \text{E}_{\text{TUCKER}} + \|\text{TF}(\epsilon_a)\| \quad (17)$$

where (16) is from Theorem 3 and (17) is from (9). Here, $\text{MSE}_{r_t \circ f_t}(\epsilon_a)$ and $\text{MSE}_{\text{TF}}(\epsilon_a)$ denote the reconstruction error of the $r_t \circ f_t$ and TF purification schemes (stated in (13) and (9), respectively). We now provide the upper-bounds of an integration of Tucker Decomposition into DDPM purification in the following Theorem 5.

Theorem 5. *The reconstruction errors introduced of the pu-*

purification $r_t \circ f_t \circ \text{TF}$ is bounded by:

$$\begin{aligned} \text{MSE}_{r_t \circ f_t \circ \text{TF}}(\epsilon_a) &\leq \text{MMSE} \left(\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \right) + \delta_{\text{DDPM}}(t) \\ &\quad + E_{\text{TUCKER}} + \|\text{TF}(\epsilon_a)\| \end{aligned} \quad (18)$$

(Proof in Appx. A.5)

Intuitively, comparing to the purification $r_t \circ f_t$, this purification process $r_t \circ f_t \circ \text{TF}$ have a better upper bound when the Tucker Decomposition can reduce the adversarial noise before forwarding the signal to the DDPM, i.e., when $E_{\text{TUCKER}} + \|\text{TF}(\epsilon_a)\| < \|\epsilon_a\|$, which suggests the usage of Tucker Decomposition at a high-adversarial-noise regime.

3.3 LoRID: Low-Rank Iterative Diffusion for Adversarial Purification

Based on the above analysis, we propose LoRID, Low-Rank Iterative Diffusion algorithm for adversarial purification. Generally, LoRID consists of two major steps: Tensor factorization, and diffusion-denoising. So far, our manuscript has considered four different configurations of LoRID, depending on the usage of looping and on how the TF and diffusions are coupled: Tensor-factorization TF, diffusion-denoising $r_t \circ f_t$, looping $(r_{t/L} \circ f_{t/L})^L$, and Tensor-factorization with diffusion-denoising $\text{TF} \circ r_t \circ f_t$. However, the default configuration that we refer to with LoRID would utilize both Tucker Decomposition (step 1) and multiple loops of diffusion-denoising (step 2), which can be described by the expression $\text{TF} \circ (r_{t/L} \circ f_{t/L})^L$. The pseudocode of LoRID is described in Appendix. B.5.

4 Experiments

This section is about our experimental setting and robustness results: Subsect. 4.1 highlights the experimental settings and Subsect. 4.2 reports our experimental results.

4.1 Experimental Setting

Datasets and attacked architectures. We evaluate LoRID on CIFAR-10/100 (Rabanser, Shchur, and Günnemann 2017), CelebA-HQ (Karras et al. 2018), and ImageNet (Deng et al. 2009). Comparisons are made against SOTA defense methods reported by RobustBench (Croce et al. 2021) on CIFAR-10 and ImageNet, and against DiffPure (Nie et al. 2022), a score-based diffusion purifier, on CIFAR-10, ImageNet, and CelebA-HQ. We use the standard WideResNet (Zagoruyko and Komodakis 2017) architecture for classification, evaluating defenses using standard accuracy (pre-perturbation) and robust accuracy (post-perturbation). When the gradients is not needed (black-box setting) in CIFAR-10, all methods are evaluated 10000 test images. On the other hand, due to the high computational cost of computing gradients for adaptive attacks against diffusion-based defenses, we assess the methods on a fixed subset of 512 randomly sampled test images, consistent with previous studies (Nie et al. 2022; Lee and Kim 2023). Further experimental details are provided in Appx. B with EOT=20.

Attacker settings. We consider two common threat models: black-box and white-box. In both scenarios, the attacker has full knowledge of the classifier. However, only in the white-box setting, the attacker also knows about the purification scheme.¹ For black-box, we adapt (Nie et al. 2022; Lee and Kim 2023) and evaluate defense methods against AutoAttack (Croce and Hein 2020) in CIFAR-10/100 and BPDA+EOT (Ferrari et al. 2023) in CelebA-HQ. For white-box, we also follow the literature and consider AutoAttack and PGD+EOT (Zimmermann 2019).

However, white-box attacks require gradient backpropagation through the diffusion-denoising path, causing memory usage to increase linearly with diffusion step t . This makes exact gradient attacks infeasible on larger datasets like CelebA-HQ and ImageNet (Kang, Song, and Li 2024). Therefore, all existing work rely on some approximations of the gradients to conduct white-box attacks on those dataset (Nie et al. 2022; Lee and Kim 2023).² To the best of our knowledge, The strongest approximation to date is the *surrogate* method (Lee and Kim 2023), which denoises noisy signals using fewer denoising steps (Song, Meng, and Ermon 2020). This approach reduces the number of denoiser calls while effectively simulating the original process (details in Appendix B.4). In summary, we use exact gradients for CIFAR-10 and the surrogate method for CelebA-HQ and ImageNet in our white-box attacks.

LoRID settings. LoRID requires the specification of both the time-step t and the looping number L , which are crucial for its iterative process. These hyperparameters are generally selected by evaluating the classifier’s performance on the clean dataset, with t and L chosen to maintain acceptable clean accuracy. Further details on this parameter selection process are provided in Appx. B.6. We report those parameters as a tuple (t, L) next to the name of our method. Additionally, obtaining an accurate Tucker decomposition for large datasets can be computationally intensive. Therefore, in such cases, LoRID is applied solely with Markov-based purification. In our results, the use of Tucker decomposition is denoted by TF next to the method’s name, e.g. (TF, t, L).

4.2 Robustness Results

We compare LoRID with the SOTA adversarial training methods documented by RobustBench (Croce et al. 2021), as well as leading adversarial purification techniques, against strong L_∞ and L_2 attacks.

CIFAR-10. Tables 2 and 3 show the defense’s performance under $L_\infty(\epsilon = 8/255)$ and $L_2(\epsilon = 0.5)$ AutoAttack on CIFAR-10. Our method achieves significant im-

¹In our white-box setting, the attacker is aware of both t and L in our LoRID framework and can fully backpropagate through the DDPM, making this scenario even stronger than the white-box assumption used by Lee and Kim (2023).

²While the *adjoint* (Nie et al. 2022) against the Score-based purification is claimed to be exact, it relies on underlying numerical solvers and they can introduce significant error. We observe that using adjoint-gradients results in significantly weaker attack than using surrogate, which is also observed and reported by Kang, Song, and Li (2024); Lee and Kim (2023).

Table 2: Standard accuracy and robust accuracy against AutoAttack L_∞ ($\epsilon = 8/255$) on CIFAR-10. * indicates the usage of extra data. The gray and white boxes indicate the black-box and white-box attacks.

Method	Standard Acc	Robust Acc
WideResNet-28-10		
(Zhang et al. 2020)*	89.36	59.96
(Wu, Xia, and Wang 2020)*	88.25	62.11
(Gowal et al. 2020)*	89.48	62.70
(Wu, Xia, and Wang 2020)	85.36	59.18
(Rebuffi et al. 2021)	87.33	61.72
(Gowal et al. 2021)	87.50	65.24
LoRID (39, 5)	90.41	88.39
(Wang et al. 2022)	85.66	33.48
(Nie et al. 2022)	89.02	46.88
LoRID (20, 24)	84.20	59.14
WideResNet-70-16		
(Gowal et al. 2020)*	91.10	66.02
(Rebuffi et al. 2021)*	92.23	68.56
(Gowal et al. 2020)	85.29	59.57
(Rebuffi et al. 2021)	88.54	64.46
(Gowal et al. 2021)	88.74	66.60
LoRID (50, 10)	85.30	69.34
LoRID (60, 10)	85.10	70.87
(Wang et al. 2022)	86.76	37.11
(Nie et al. 2022)	90.07	45.31
LoRID (25, 20)	84.60	66.40
LoRID (10, 40)	86.90	59.20

Table 3: Standard accuracy and robust accuracy against AutoAttack L_2 ($\epsilon = 0.5$) on CIFAR-10. * indicates the usage of extra data. The gray and white boxes indicate the black-box and white-box attacks.

Method	Standard Acc	Robust Acc
WideResNet-28-10		
(Pang et al. 2022)*	90.83	78.10
(Rebuffi et al. 2021)*	91.79	78.69
(Wang et al. 2023)*	95.16	83.68
LoRID (39, 4)	90.34	89.69
(Wang et al. 2022)	85.66	73.32
(Nie et al. 2022)	91.03	64.06
LoRID (15, 30)	85.4	77.9
LoRID (20, 24)	84.2	73.6

improvements in both standard and robust accuracy compared to previous SOTA in both black-box and white-box settings. Particularly, LoRID improves black-box robust accuracy by 23.15% on WideResNet-28-10 and by 4.27% on WideResNet-70-16. Additionally, our method surpasses baseline robust accuracy in the white-box by 12.26% on WideResNet-28-10 and by 21.09% on WideResNet-70-16.

ImageNet. Table 4 shows the robustness performance against L_∞ ($\epsilon = 4/255$) AutoAttack on WideResNet-28-10. Our method significantly outperforms SOTA baselines in both standard and robust accuracies.

CelebA-HQ. For large datasets like CelebA-HQ, attackers often use the BPDA+EOT attack (Tramer et al. 2020; Hill, Mitchell, and Zhu 2021), which substitutes exact gradients with classifier gradients. We evaluated our approach against

Table 4: Standard accuracy and robust accuracy against white-box PGD+EOT L_∞ ($\epsilon = 4/255$) on ImageNet.

Method	Standard Acc	Robust Acc
WideResNet-28-10		
(Wong, Rice, and Kolter 2020)	53.83	28.04
(Engstrom et al. 2019)	62.42	33.20
(Salman et al. 2020)	68.46	39.25
(Nie et al. 2022)	71.16	44.39
(Lee and Kim 2023)	70.74	42.15
LoRID (5, 30)	73.98	56.54

Table 5: Standard accuracy and robust accuracy against BPDA+EOT L_∞ on Celeb HQ-Eyeglasses attribute classifier, with $\epsilon = 16/255$.

Method	Standard Acc	Robust Acc
Eyeglasses attribute classifier for CelebA-HQ		
(Chai et al. 2021)	99.37	26.37
(Richardson et al. 2021)	93.95	75.00
(Nie et al. 2022)	93.77	90.63
LoRID (100, 15)	98.91	97.80

baseline methods under this attack, as shown in Table 5. Our method outperforms the best baseline in robust accuracy by 7.17%, while also maintaining high standard accuracy.

Table 6: Performance of LoRID against black-box AutoAttack on CIFAR-10 at high-noise regime.

Method	ϵ	Standard Acc	Robust Acc
WideResNet-28-1 L_∞ attacks			
(Gowal et al. 2021)	8/255	87.50	65.24
LoRID (39, 5)	8/255	90.41	88.39
LoRID (TF, 40, 2)	8/255	89.32	88.12
LoRID (49, 8)	16/255	89.00	85.86
LoRID (TF, 42, 5)	16/255	88.66	86.23
LoRID (49, 12)	32/255	89.20	69.87
LoRID (TF, 48, 9)	32/255	88.35	78.04

High-noise regime. We demonstrate the effectiveness of Tucker decomposition in high-noise settings, as shown in Table 6. Specifically, we compare LoRID to the best known robustness results from Gowal et al. (2021) under black-box L_∞ AutoAttack. The results indicate that Tucker decomposition becomes increasingly beneficial as noise levels rise, as supported by Theorem A.5. Notably, with Tucker decomposition, LoRID’s robustness at a very high noise level ($\epsilon = 32/255$) surpasses SOTA performance at the standard noise level ($\epsilon = 8/255$) by 12.8%.

5 Conclusion

We introduced LoRID, a defense strategy that uses multiple looping in the early stages of diffusion models to purify adversarial examples. To enhance robustness in high noise regimes, we integrated Tucker decomposition. Our approach, validated by theoretical analysis and extensive experiments on CIFAR-10/100, ImageNet, and CelebA-HQ, significantly outperforms state-of-the-art methods against strong adaptive attacks like AutoAttack, PGD+EOT and BPDA+EOT.

References

- Bai, T.; Luo, J.; Zhao, J.; Wen, B.; and Wang, Q. 2021. Recent Advances in Adversarial Training for Adversarial Robustness. *arXiv:2102.01356*.
- Bergqvist, G.; and Larsson, E. G. 2010. The Higher-Order Singular Value Decomposition: Theory and an Application [Lecture Notes]. *IEEE Signal Processing Magazine*, 27(3): 151–154.
- Bhattacharai, M.; Kaymak, M. C.; Barron, R.; Nebgen, B.; Rasmussen, K.; and Alexandrov, B. S. 2023. Robust Adversarial Defense by Tensor Factorization. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, 308–315. IEEE.
- Blahut, R. E. 1987. *Principles and practice of information theory*. USA: Addison-Wesley Longman Publishing Co., Inc. ISBN 0201107090.
- Blau, T.; Ganz, R.; Kawar, B.; Bronstein, A.; and Elad, M. 2022. Threat model-agnostic adversarial defense using diffusion models. *arXiv preprint arXiv:2207.08089*.
- Chai, L.; Zhu, J.-Y.; Shechtman, E.; Isola, P.; and Zhang, R. 2021. Ensembling with Deep Generative Views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- Cover, T. M.; and Thomas, J. A. 2006. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience. ISBN 0471241954.
- Croce, F.; Andriushchenko, M.; Sehwag, V.; Debenedetti, E.; Flammarion, N.; Chiang, M.; Mittal, P.; and Hein, M. 2021. RobustBench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Croce, F.; and Hein, M. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*.
- De Lathauwer, L.; De Moor, B.; and Vandewalle, J. 2000. A Multilinear Singular Value Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4): 1253–1278.
- Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.
- Engstrom, L.; Ilyas, A.; Salman, H.; Santurkar, S.; and Tsipras, D. 2019. Robustness (Python Library).
- Ferrari, C.; Becattini, F.; Galteri, L.; and Bimbo, A. D. 2023. (Compress and Restore)N: A Robust Defense Against Adversarial Attacks on Image Classification. *ACM Trans. Multimedia Comput. Commun. Appl.*, 19(1s).
- Gallager, R. G. 1968. *Information Theory and Reliable Communication*. USA: John Wiley & Sons, Inc. ISBN 0471290483.
- Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.
- Gowal, S.; Qin, C.; Uesato, J.; Mann, T.; and Kohli, P. 2020. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*.
- Gowal, S.; Rebuffi, S.-A.; Wiles, O.; Stimberg, F.; Calian, D. A.; and Mann, T. A. 2021. Improving Robustness using Generated Data. *Advances in Neural Information Processing Systems*, 34.
- Guo, D.; Shamai, S.; and Verdu, S. 2005. Mutual information and minimum mean-square error in Gaussian channels. *IEEE Transactions on Information Theory*, 51(4): 1261–1282.
- Hackbusch, W. 2012. Tensor Spaces and Numerical Tensor Calculus. *Springer Series in Computational Mathematics*.
- Hill, M.; Mitchell, J. C.; and Zhu, S.-C. 2021. Stochastic Security: Adversarial Defense Using Long-Run Dynamics of Energy-Based Models. In *International Conference on Learning Representations*.
- Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. In *Neural Information Processing Systems (NeurIPS)*.
- Kang, M.; Song, D.; and Li, B. 2024. DiffAttack: Evasion Attacks Against Diffusion-Based Adversarial Purification. *arXiv:2311.16124*.
- Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv:1710.10196*.
- Kim, H. 2020. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*.
- Kolda, T. G.; and Bader, B. W. 2009. Tensor decompositions and applications. *SIAM review*, 51(3): 455–500.
- Lee, M. J.; and Kim, D. 2023. Robust Evaluation of Diffusion-Based Adversarial Purification. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 134–144.
- Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2019. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv:1706.06083*.
- Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, 8162–8171. PMLR.
- Nie, W.; Guo, B.; Huang, Y.; Xiao, C.; Vahdat, A.; and Anandkumar, A. 2022. Diffusion Models for Adversarial Purification. In *International Conference on Machine Learning (ICML)*.
- Pang, T.; Lin, M.; Yang, X.; Zhu, J.; and Yan, S. 2022. Robustness and Accuracy Could Be Reconcilable by (Proper) Definition. In *International Conference on Machine Learning*.
- Qiu, S.; Liu, Q.; Zhou, S.; and Wu, C. 2019. Review of Artificial Intelligence Adversarial Attack and Defense Technologies. *Applied Sciences*, 9(5).
- Rabanser, S.; Shchur, O.; and Günnemann, S. 2017. Introduction to Tensor Decompositions and their Applications in Machine Learning. *arXiv:1711.10781*.
- Rebuffi, S.-A.; Gowal, S.; Calian, D. A.; Stimberg, F.; Wiles, O.; and Mann, T. 2021. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*.

- Richardson, E.; Alaluf, Y.; Patashnik, O.; Nitzan, Y.; Azar, Y.; Shapiro, S.; and Cohen-Or, D. 2021. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Salakhutdinov, R. 2015. Learning deep generative models. *Annual Review of Statistics and Its Application*, 2: 361–385.
- Salman, H.; Ilyas, A.; Engstrom, L.; Kapoor, A.; and Madry, A. 2020. Do Adversarially Robust ImageNet Models Transfer Better? In *Advances in Neural Information Processing Systems*.
- Shi, C.; Holtz, C.; and Mishne, G. 2021. Online Adversarial Purification based on Self-Supervision. *arXiv:2101.09387*.
- Song, J.; Meng, C.; and Ermon, S. 2020. Denoising Diffusion Implicit Models. *arXiv:2010.02502*.
- Song, Y.; Kim, T.; Nowozin, S.; Ermon, S.; and Kushman, N. 2018. PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples. In *International Conference on Learning Representations*.
- Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. *arXiv:2011.13456*.
- Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.
- Tramer, F.; Carlini, N.; Brendel, W.; and Madry, A. 2020. On Adaptive Attacks to Adversarial Example Defenses. *Advances in Neural Information Processing Systems*, 33.
- Wang, J.; Lyu, Z.; Lin, D.; Dai, B.; and Fu, H. 2022. Guided Diffusion Model for Adversarial Purification. *arXiv:2205.14969*.
- Wang, Z.; Pang, T.; Du, C.; Lin, M.; Liu, W.; and Yan, S. 2023. Better Diffusion Models Further Improve Adversarial Training. In *International Conference on Machine Learning (ICML)*.
- Wong, E.; Rice, L.; and Kolter, J. Z. 2020. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*.
- Wu, D.; Xia, S.-T.; and Wang, Y. 2020. Adversarial weight perturbation helps robust generalization. *arXiv preprint arXiv:2004.05884*.
- Xiao, C.; Chen, Z.; Jin, K.; Wang, J.; Nie, W.; Liu, M.; Anandkumar, A.; Li, B.; and Song, D. 2022. DensePure: Understanding Diffusion Models towards Adversarial Robustness. *arXiv preprint arXiv:2211.00322*.
- Zagoruyko, S.; and Komodakis, N. 2017. Wide Residual Networks. *arXiv:1605.07146*.
- Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; and Jordan, M. 2019. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, 7472–7482. PMLR.
- Zhang, J.; Zhu, J.; Niu, G.; Han, B.; Sugiyama, M.; and Kankanhalli, M. 2020. Geometry-aware Instance-reweighted Adversarial Training. In *International Conference on Learning Representations*.
- Zimmermann, R. S. 2019. Comment on "Adv-BNN: Improved Adversarial Defense through Robust Bayesian Neural Network". *CoRR*, abs/1907.00895.

A Appendix

A.1 Proof of Theorem 1

In the following, we provide the proof of Theorem 1. We restate the Theorem below:

Theorem. Let $\{\mathbf{x}_t^{(i)}\}_{t \in \{0, \dots, T\}}$, $i \in \{1, 2\}$ be two diffusion processes given by the forward equation (1) of a DDPM. Denote $q_t^{(1)}$ and $q_t^{(2)}$ the distributions of $\mathbf{x}_t^{(1)}$ and $\mathbf{x}_t^{(2)}$, respectively. Then, for all $t \in \{0, \dots, T-1\}$, we have

$$D_{KL}(q_t^{(1)} \| q_t^{(2)}) \geq D_{KL}(q_{t+1}^{(1)} \| q_{t+1}^{(2)})$$

Proof. We start by restating the chain rule for relative entropy (Cover and Thomas 2006):

$$\begin{aligned} & D_{KL}(p(\mathbf{z}_1, \mathbf{z}_2) \| p'(\mathbf{z}_1, \mathbf{z}_2)) \\ &= D_{KL}(p(\mathbf{z}_1) \| p'(\mathbf{z}_1)) + D_{KL}(p(\mathbf{z}_2 | \mathbf{z}_1) \| p'(\mathbf{z}_2 | \mathbf{z}_1)) \end{aligned} \quad (19)$$

Then, by denoting $q^{(i)}(\mathbf{x}_{t+1}, \mathbf{x}_t)$ the joint distribution of $\mathbf{x}_{t+1}^{(i)}$ and $\mathbf{x}_t^{(i)}$, the chain rule gives us:

$$\begin{aligned} & D_{KL}(q^{(1)}(\mathbf{x}_{t+1}, \mathbf{x}_t) \| q^{(2)}(\mathbf{x}_{t+1}, \mathbf{x}_t)) \quad (20) \\ &= D_{KL}(q^{(1)}(\mathbf{x}_{t+1} | \mathbf{x}_t) \| q^{(2)}(\mathbf{x}_{t+1} | \mathbf{x}_t)) + D_{KL}(q^{(1)}(\mathbf{x}_t | \mathbf{x}_{t+1}) \| q^{(2)}(\mathbf{x}_t | \mathbf{x}_{t+1})) \quad (21) \\ &= D_{KL}(q^{(1)}(\mathbf{x}_{t+1} | \mathbf{x}_t) \| q^{(2)}(\mathbf{x}_{t+1} | \mathbf{x}_t)) + D_{KL}(q^{(1)}(\mathbf{x}_t | \mathbf{x}_{t+1}) \| q^{(2)}(\mathbf{x}_t | \mathbf{x}_{t+1})) \quad (22) \end{aligned}$$

Note that, due to (1), we have $q^{(1)}(\mathbf{x}_{t+1} | \mathbf{x}_t) = q^{(2)}(\mathbf{x}_{t+1} | \mathbf{x}_t)$, this implies the last term of Eq. (22) $D_{KL}(q^{(1)}(\mathbf{x}_t | \mathbf{x}_{t+1}) \| q^{(2)}(\mathbf{x}_t | \mathbf{x}_{t+1})) = 0$. Thus, we have:

$$\begin{aligned} & D_{KL}(q^{(1)}(\mathbf{x}_{t+1} | \mathbf{x}_t) \| q^{(2)}(\mathbf{x}_{t+1} | \mathbf{x}_t)) \\ &= D_{KL}(q_t^{(1)} \| q_t^{(2)}) \end{aligned} \quad (23)$$

Thus, due to the non-negativity of the KL divergence $D_{KL}(q^{(1)}(\mathbf{x}_t | \mathbf{x}_{t+1}) \| q^{(2)}(\mathbf{x}_t | \mathbf{x}_{t+1}))$, we have the Theorem:

$$D_{KL}(q_t^{(1)} \| q_t^{(2)}) \geq D_{KL}(q_{t+1}^{(1)} \| q_{t+1}^{(2)}) \quad (24)$$

□

A.2 Proof of Theorem 2

Theorem. Let $\{\mathbf{x}_t\}_{t \in \{0, \dots, T\}}$ be a diffusion process defined by the forward equation (1) where \mathbf{x}_0 is the adversarial sample. i.e. $\mathbf{x}_0 = \mathbf{x}_{clean} + \epsilon_a$. For any given time t , we have

$$\mathbb{E}[\|\hat{\mathbf{x}}_0(t) - \mathbf{x}_{clean}\|] \geq \text{MMSE}\left(\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t}\right) - \|\epsilon_a\|$$

where the expectation is taken over the distribution of \mathbf{x}_{clean} and $\text{MMSE}(\text{SNR})$ is the minimum mean-square error achievable by optimal estimation of the input given the output of Gaussian channel with a signal-to-noise ratio of SNR. The function $\text{MMSE}(\text{SNR})$ has the following form (Guo, Shamai, and Verdu 2005):

$$1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-y^2/2} \tanh(\text{SNR} - \sqrt{\text{SNR}}y) dy$$

Proof. We consider (3) as a Gaussian channel $\mathbf{y}_t = \frac{\sqrt{\bar{\alpha}_t}}{\sqrt{1 - \bar{\alpha}_t}} \mathbf{x}_0 + \epsilon_0$, and $\hat{\mathbf{x}}_0(t)$ as an estimation of $\mathbf{x}_0(t)$ given \mathbf{y}_t . By denoting $\hat{\mathbf{x}}_0^*(\mathbf{y}_t)$ the best estimator of \mathbf{x}_0 given \mathbf{y}_t , we have

$$\begin{aligned} \mathbb{E}[\|\hat{\mathbf{x}}_0(t) - \mathbf{x}_0\|] &\geq \mathbb{E}[\|\hat{\mathbf{x}}_0^*(\mathbf{y}_t) - \mathbf{x}_0\|] \\ &= \text{MMSE}\left(\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t}\right) \end{aligned} \quad (25)$$

where the equality is from the definition of the $\text{MMSE}(\text{SNR})$ function. We are now ready to show (10). In fact, from the triangle inequality, we have:

$$\begin{aligned} \|\hat{\mathbf{x}}_0(t) - \mathbf{x}_{clean}\| &= \|\hat{\mathbf{x}}_0(t) - \mathbf{x}_0 - \epsilon_a\| \\ &\geq \|\hat{\mathbf{x}}_0(t) - \mathbf{x}_0\| - \|\epsilon_a\| \end{aligned}$$

Combining the above with (25) gives us:

$$\mathbb{E}[\|\hat{\mathbf{x}}_0(t) - \mathbf{x}_{clean}\|] \geq \text{MMSE}\left(\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t}\right) - \|\epsilon_a\| \quad (26)$$

Thus, we have the Theorem.

For comprehensiveness, we now highlight how to derive (11). Particularly, we use the following relation between mutual information of a channel, i.e., $I(\text{SNR})$, and the minimum mean-square error of Gaussian channel (Guo, Shamai, and Verdu 2005):

$$\frac{dI(\text{SNR})}{d\text{SNR}} = \frac{1}{2} \text{MMSE}(\text{SNR}) \quad (27)$$

where the mutual information of a channel $I(\text{SNR})$ is given as (Blahut 1987) (p. 274), and (Gallager 1968) (Problem 4.22):

$$\text{SNR} - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-y^2/2} \log \cosh(\text{SNR} - \sqrt{\text{SNR}}y) dy. \quad (28)$$

Here, the mutual information is computed in nats. Taking the derivative of (28) gives us (11). □

A.3 Proof of Theorem 3

Theorem. Additionally to the conditions stated in Theorem 2, if the DDPM is able to recover the original signal \mathbf{x}_0 within an error $\delta_{\text{DDPM}}(t)$ in the expectation, i.e., for all t ,

$$\mathbb{E}[\|\hat{\mathbf{x}}_0(t) - \mathbf{x}_0\|] \leq \mathbb{E}[\|\hat{\mathbf{x}}_0^*(\mathbf{y}_t) - \mathbf{x}_0\|] + \delta_{\text{DDPM}}(t)$$

where $\hat{\mathbf{x}}_0^*(\mathbf{y}_t)$ is the best estimator of \mathbf{x}_0 given $\mathbf{y}_t = \frac{\sqrt{\bar{\alpha}_t}}{\sqrt{1 - \bar{\alpha}_t}} \mathbf{x}_0 + \epsilon_0$, then, we have the reconstructed error $\mathbb{E}[\|\hat{\mathbf{x}}_0(t) - \mathbf{x}_{clean}\|]$ is upper-bounded by:

$$\text{MMSE}\left(\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t}\right) + \delta_{\text{DDPM}}(t) + \|\epsilon_a\|$$

Proof. From the triangle inequality, we have:

$$\begin{aligned} \|\hat{\mathbf{x}}_0(t) - \mathbf{x}_{clean}\| &= \|\hat{\mathbf{x}}_0(t) - \mathbf{x}_0 - \epsilon_a\| \\ &\leq \|\hat{\mathbf{x}}_0(t) - \mathbf{x}_0\| + \|\epsilon_a\| \end{aligned}$$

Combining the above with the assumption on the recovering error stated in the Theorem gives us:

$$\begin{aligned} & \mathbb{E} [\|\hat{\mathbf{x}}_0(t) - \mathbf{x}_0\| + \|\epsilon_a\|] = \mathbb{E} [\|\hat{\mathbf{x}}_0(t) - \mathbf{x}_0\|] + \mathbb{E} [\|\epsilon_a\|] \\ & \leq \mathbb{E} [\|\hat{\mathbf{x}}_0^*(\mathbf{y}_t) - \mathbf{x}_0\|] + \delta_{\text{DDPM}}(t) + \|\epsilon_a\| \\ & = \text{MMSE} \left(\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \right) + \delta_{\text{DDPM}}(t) + \|\epsilon_a\| \end{aligned}$$

We then have the Theorem. \square

A.4 Proof of Theorem 4

Theorem. Let $\{\mathbf{x}_t\}_{t \in \{0, \dots, T\}}$ be a diffusion process defined by the forward Eq. (1) where \mathbf{x}_0 is the adversarial sample. i.e, $\mathbf{x}_0 = \mathbf{x}_{\text{clean}} + \epsilon_a$. For any given time t , we have the reconstructed error $\mathbb{E} [\|\hat{\mathbf{x}}_0^L(t/L) - \mathbf{x}_{\text{clean}}\|]$ is upper-bounded by:

$$L \times \left(\text{MMSE} \left(\frac{\bar{\alpha}_{t/L}}{1 - \bar{\alpha}_{t/L}} \right) + \delta_{\text{DDPM}} \left(\frac{t}{L} \right) \right) + \|\epsilon_a\|$$

where the expectation is taken over the distribution of $\mathbf{x}_{\text{clean}}$.

Proof. From the triangle inequality, we have:

$$\begin{aligned} & \|\hat{\mathbf{x}}_0^L(t/L) - \mathbf{x}_{\text{clean}}\| \\ & = \left\| \sum_{l=1}^{L-1} (\hat{\mathbf{x}}_0^{l+1}(t/L) - \hat{\mathbf{x}}_0^l(t/L)) + (\hat{\mathbf{x}}_0^1(t/L) - \mathbf{x}_0) - \epsilon_a \right\| \quad (29) \\ & \leq \sum_{l=1}^{L-1} \|\hat{\mathbf{x}}_0^{l+1}(t/L) - \hat{\mathbf{x}}_0^l(t/L)\| + \|\hat{\mathbf{x}}_0^1(t/L) - \mathbf{x}_0\| + \|\epsilon_a\| \quad (30) \end{aligned}$$

Noting that, each of the signal $\hat{\mathbf{x}}_0^{l+1}(t/L)$ is the output of the purification $r_{t/L} \circ f_{t/L}$ on the input $\hat{\mathbf{x}}_0^l(t/L)$. Thus, from the condition of δ_{DDPM} (stated in Theorem 3), we have:

$$\begin{aligned} & \mathbb{E} [\|\hat{\mathbf{x}}_0^{l+1}(t/L) - \hat{\mathbf{x}}_0^l(t/L)\|] \\ & \leq \mathbb{E} \left[\left\| \hat{\mathbf{x}}_0^{l+1*}(\mathbf{y}_{t/L}^l) - \hat{\mathbf{x}}_0^l(t/L) \right\| \right] + \delta_{\text{DDPM}}(t/L), \quad (31) \end{aligned}$$

for all t , where $\mathbf{y}_{t/L}^l = \frac{\sqrt{\bar{\alpha}_{t/L}}}{\sqrt{1 - \bar{\alpha}_{t/L}}} \hat{\mathbf{x}}_0^l(t/L) + \epsilon$, and $\hat{\mathbf{x}}_0^{l+1*}(\mathbf{y}_{t/L}^l)$ is the optimal reconstruction of $\hat{\mathbf{x}}_0^l(t/L)$ given $\mathbf{y}_{t/L}^l$.

Notice that the SNR of that channel is $\frac{\sqrt{\bar{\alpha}_{t/L}}}{\sqrt{1 - \bar{\alpha}_{t/L}}}$, thus, $\mathbb{E} \left[\left\| \hat{\mathbf{x}}_0^{l+1*}(\mathbf{y}_{t/L}^l) - \hat{\mathbf{x}}_0^l(t/L) \right\| \right] = \text{MMSE} \left(\frac{\bar{\alpha}_{t/L}}{1 - \bar{\alpha}_{t/L}} \right)$. Applying that to (31) gives us

$$\begin{aligned} & \sum_{l=1}^{L-1} \|\hat{\mathbf{x}}_0^{l+1}(t/L) - \hat{\mathbf{x}}_0^l(t/L)\| + \|\hat{\mathbf{x}}_0^1(t/L) - \mathbf{x}_0\| \\ & \leq L \times \left(\text{MMSE} \left(\frac{\bar{\alpha}_{t/L}}{1 - \bar{\alpha}_{t/L}} \right) + \delta_{\text{DDPM}}(t/L) \right), \quad (32) \end{aligned}$$

since $\hat{\mathbf{x}}_0^1(t/L)$ can also be considered as the optimal reconstruction of \mathbf{x}_0 given $\mathbf{y}_{t/L}^1 = \frac{\sqrt{\bar{\alpha}_{t/L}}}{\sqrt{1 - \bar{\alpha}_{t/L}}} \mathbf{x}_0 + \epsilon$. Using the above result on (30) gives us the Theorem. \square

A.5 Proof of Theorem 5

Theorem. The reconstruction errors introduced of the purification $r_t \circ f_t \circ \text{TF}$ is bounded by:

$$\begin{aligned} \text{MSE}_{r_t \circ f_t \circ \text{TF}}(\epsilon_a) & \leq \text{MMSE} \left(\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \right) + \delta_{\text{DDPM}}(t) \\ & \quad + \text{E}_{\text{TUCKER}} + \|\text{TF}(\epsilon_a)\| \end{aligned}$$

Proof. We denote the purified signal of the adaptation of tensor-factorization into the purification process of DDPM by:

$$\hat{\mathbf{x}}_{\text{DDPM-TF}} := r_t \circ f_t \circ \text{TF}(\mathbf{x} + \epsilon_a) \quad (33)$$

By denoting $\epsilon_a^{\text{TF}} := \text{TF}(\mathbf{x} + \epsilon_a) - \mathbf{x}$, we can consider (33) as the applying of the purification $r_t \circ f_t$ onto $\mathbf{x} + \epsilon_a^{\text{TF}}$. By applying Theorem 3, we have:

$$\text{MSE}_{r_t \circ f_t \circ \text{TF}}(\epsilon_a) \leq \text{MMSE} \left(\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \right) + \delta_{\text{DDPM}}(t) + \|\epsilon_a^{\text{TF}}\| \quad (34)$$

Since ϵ_a^{TF} is the purification error of tensor-factorization on $\mathbf{x} + \epsilon_a$, (9) implies:

$$\|\epsilon_a^{\text{TF}}\| = \text{MSE}_{\text{TF}}(\epsilon_a) \leq \text{E}_{\text{TUCKER}} + \|\text{TF}(\epsilon_a)\| \quad (35)$$

which gives us

$$\begin{aligned} \text{MSE}_{r_t \circ f_t \circ \text{TF}}(\epsilon_a) & \leq \text{MMSE} \left(\frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \right) + \delta_{\text{DDPM}}(t) \\ & \quad + \text{E}_{\text{TUCKER}} + \|\text{TF}(\epsilon_a)\| \quad (36) \end{aligned} \quad \square$$

B Experimental Details

In this appendix, we provide the details of the experimental results reported in our main manuscript.

B.1 Experimental details of Fig. 1

Intuitively, Theorem 4 captures the impact of increasing the iterative factor L on the reconstruction error induced by the purification process. Experiment in Fig. 1 aims to illustrate that behavior. As the MMSE (11) depends on the actual input distribution, to visualize the lower bound of Theorem 4, we consider the input of the Gaussian channel $\mathbf{y}_t = \frac{\sqrt{\bar{\alpha}_t}}{\sqrt{1 - \bar{\alpha}_t}} \mathbf{x}_0 + \epsilon_0$ induced by (3) to be a Gaussian signal. Thus, the MMSE(SNR) is simply $1/(1 + \text{SNR})$ (Guo, Shamai, and Verdu 2005). Given that, we can plot the dominant term $L \times \left(\text{MMSE} \left(\frac{\bar{\alpha}_{t/L}}{1 - \bar{\alpha}_{t/L}} \right) \right)$ of the bound in Theorem 4 and plot it in Fig. 1.

B.2 Tucker Tensor Decomposition

Formally, given a tensor \mathcal{X} of size $I_1 \times I_2 \times \dots \times I_N$, we project it into a lower-dimensional space using Tucker factor matrices: $\mathcal{G} = \mathcal{X} \times_2 \mathbf{U}_2^T \times_3 \mathbf{U}_3^T \times_4 \mathbf{U}_4^T \times_5 \mathbf{U}_5^T$, where \mathcal{G} is the core tensor of size $r_1 \times r_2 \times \dots \times r_5$ for $r_n \leq I_n$ ($n \in \{1, 2, \dots, 5\}$), \mathbf{U}_n is the factor matrix for mode n with size $I_n \times r_n$, and \times_n denotes the mode- n product. The reconstruction signal is $\hat{\mathcal{X}} = \mathcal{G} \times_5 \mathbf{U}_5 \times_4 \mathbf{U}_4 \times_3 \mathbf{U}_3 \times_2 \mathbf{U}_2$. For convenient, we express the above process of projecting-recovering as $\mathcal{X} \approx \hat{\mathcal{X}} = d^{-1} \circ d(\mathcal{X})$, where d and d^{-1} denote the projection and the reconstruction, respectively.

B.3 Attacking methods

We use AutoAttack at perturbation levels of $(\frac{8}{255}, \frac{16}{255}, \frac{32}{255})$ and when applicable, compare our results with the reported accuracies in the RobustBench benchmark. For evaluation, we employ the RobustBench codebase and model zoo³ to obtain hyperparameters whenever the standard model is available. For AutoAttack under the L_∞ norm, and due to the complexity of ImageNet, we use a perturbation level of $\frac{4}{255}$ for the implementation of the EOT-PGD attack (Kim 2020). For black-box attacks, we use the standard version of AutoAttack, which includes APGD-CE, APGD-T, FAB-T, and Square, as well as PGD+EOT on CelebA-HQ. For white-box attacks, we use the AutoAttack RAND version, which comprises APGD-CE and APGD-DLR. For high iterations, we utilize the repository from (Kang, Song, and Li 2024) to explicitly compute the attack gradients rather than relying on the computational graph.

B.4 Implicit trick for surrogate gradients

To obtain the exact gradients resulting from the DDPM with t effective time-steps, the attacker needs to store and backward a computational graph whose size is proportional to $N \times t \times \|\Phi\|$, where N is the batch size and $\|\Phi\|$ is the size of the DDPM’s denoiser. As Φ is typically of millions parameters, exact gradients’ computation creates an extreme burden on computational resource. To alleviate this challenge, Lee and Kim (2023) proposes to compute the gradient in a *skip-ping* manner. In particular, instead of iteratively compute the reconstructed signal, they use a proxy process of skipping k time-step per iteration and computes:

$$\begin{aligned} \mathbf{x}_{t-k} &\approx \frac{\sqrt{\bar{\alpha}_{t-k}}}{\sqrt{\bar{\alpha}_t}} \mathbf{x}_t \\ &+ \sqrt{\bar{\alpha}_{t-k}} \times \left(\sqrt{\frac{1 - \bar{\alpha}_{t-k}}{\bar{\alpha}_{t-k}}} - \sqrt{\frac{1 - \bar{\alpha}_t}{\bar{\alpha}_t}} \right) \epsilon_\theta(\mathbf{x}_t, t) \end{aligned} \quad (37)$$

Then, the surrogate gradient, which is the gradients computing based on the reconstructed signal resulted from the skipping computation above, is used instead of the exact gradients. As this skipping trick reduces the computational graph by a factor of k , it allows attacks with reasonable computational complexity.

It is noteworthy to point out that, this surrogate method results in attackers that is heuristically significantly stronger than the adjoint (Nie et al. 2022), as reported in (Lee and Kim 2023).

B.5 Algorithm

This appendix provides the pseudocode for LoRID, which is shown in Algo. 1.

B.6 Selection of LoRiD parameters

In the evaluation of our model, we meticulously selected the parameters t and iterations (L) based on a detailed analysis of clean accuracy as a function of the LoRID parameters. The contour map presented in Figure 4 illustrates the

³<https://github.com/RobustBench/robustbench>

Algorithm 1: Low-Rank Iterative Diffusion

Input : Input image \mathbf{x} , looping parameter L , and total purified time-step t
Given : DDPM’s forwarding and reversing functions $\{f_t, r_t\}_{t=1}^T$, and Tucker decomposition TF
Output: Purified image $\hat{\mathbf{x}}_0$
Step 1: Tucker decomposition
 $\mathbf{x} \leftarrow \text{TF}(\mathbf{x})$
Step 2: Iterative Diffusion
 $t' = \lfloor t/L \rfloor$
 For $l = 1$ to L do:
 $\mathbf{x} \leftarrow f_{t'}(r_{t'}(\mathbf{x}))$
 Return $\hat{\mathbf{x}}_0 \leftarrow \mathbf{x}$

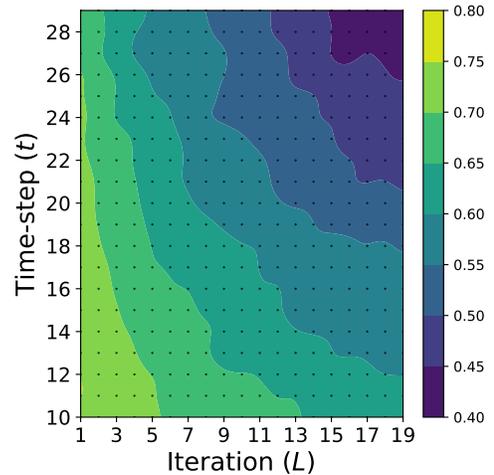


Figure 4: Impact of the time-step t and iterative factor L on the standard accuracy of WideResnet-28-10 in CIFAR-100 dataset.

nuanced relationship between these parameters and clean accuracy, where variations in t and L can lead to significant shifts in performance. Specifically, the map reveals that while selecting parameter regions that correspond to higher clean accuracy might intuitively seem advantageous, it paradoxically leads to a degradation in robust accuracy. This is attributed to the reduction in model complexity when parameters are set to optimize for clean accuracy, thereby compromising the model’s robustness under adversarial conditions. Conversely, increasing t and iter enhances the complexity of the diffusion process, which bolsters robustness but at the expense of clean accuracy. Given this trade-off, our parameter selection strategy focused on identifying a balance that optimizes both clean and robust accuracies, ensuring that the model remains resilient without sacrificing performance on clean data.

B.7 Hardware Setup

For this paper, our experiments utilized a HPC cluster where each node integrates four NVIDIA Hopper (H100) GPUs,

each paired with a corresponding NVIDIA Grace CPU via NVLink-C2C, facilitating rapid data transfer crucial for intensive computational tasks. The GPUs are equipped with 96GB of HBM2 memory, optimal for handling large models and datasets. This setup is supported by an HPE/Cray Slingshot 11 interconnect with a bandwidth of 200GB/s, ensuring efficient inter-node communication essential for scalable machine learning operations.

B.8 Code availability

The code used in this study is currently under review for release by the organization. We are awaiting approval, and once granted, the code will be made publicly available.