Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

TendiffPure: a convolutional tensor-train denoising diffusion model for purification

Mingyuan BAI[1], Derun ZHOU[1,2], Qibin ZHAO[‡1]

[1]*RIKEN AIP, Tokyo 1030027, Japan*

[2]*School of Environment and Society, Tokyo Institute of Technology, Tokyo 1528550, Japan*

E-mail: mingyuan.bai@riken.jp; zhouderun2000@gmail.com; qibin.zhao@riken.jp

Received May 31, 2023; Revision accepted Jan. 3, 2024; Crosschecked Jan. 15, 2024

Abstract: Diffusion models are effective purification methods, where the noises or adversarial attacks are removed using generative approaches before pre-existing classifiers conducting classification tasks. However, the efficiency of diffusion models is still a concern, and existing solutions are based on knowledge distillation which can jeopardize the generation quality because of the small number of generation steps. Hence, we propose TendiffPure as a tensorized and compressed diffusion model for purification. Unlike the knowledge distillation methods, we directly compress U-Nets as backbones of diffusion models using tensor-train decomposition, which reduces the number of parameters and captures more spatial information in multi-dimensional data such as images. The space complexity is reduced from $O(N^2)$ to $O(NR^2)$ with $R \leq 4$ as the tensor-train rank and N as the number of channels. Experimental results show that TendiffPure can more efficiently obtain high-quality purification results and outperforms the baseline purification methods on CIFAR-10, Fashion-MNIST, and MNIST datasets for two noises and one adversarial attack.

Key words: Diffusion models; Tensor decomposition; Image denoising
https://doi.org/10.1631/FITEE.2300392 **CLC number:** TP391.4

1 Introduction

Diffusion models (Dhariwal and Nichol, 2021; Gao et al., 2023) are ubiquitous in the recent three years in text, image, and video generation. They appeal to both academics and practitioners for their mode coverage, stationary training objective, and easy scalability. Among the generative models, compared with generative adversarial networks (GANs), as diffusion models do not require adversarial training, they are able to process a significantly larger range of distributions of features and hence avoid mode collapse. For the same reason, their training process is more stable than that of GANs. In terms of sample quality, diffusion models outperform variational autoencoders (VAEs) and normalizing flows (Ho and Salimans, 2021). They demonstrate strong capabilities as purification methods of removing both noises and adversarial attacks for data preprocessing, followed by the classifiers.

Benefiting from denoising score matching (Vincent, 2011) or sliced score matching (Song Y et al., 2020), diffusion models using score-based generative modeling methods are scalable to high-dimensional data in the deep learning settings. However, they still suffer from low sampling speed, which is caused by the iterative generation process. In specific, for each sampling or generation step, data are iteratively updated following the direction determined by the score until the mode is reached, where the score can be described by a score function. This score function is commonly approximated by a U-Net which is the backbone of a diffusion model. A large variety of data naturally possess multi-dimensional spatial

‡ Corresponding author
ORCID: Mingyuan BAI, https://orcid.org/0000-0002-2454-4219; Derun ZHOU, https://orcid.org/0009-0008-0931-4520; Qibin ZHAO, https://orcid.org/0000-0002-4442-3182

structures, which can be easily neglected by convolution kernels of U-Nets (Ronneberger et al., 2015). U-Nets are the common backbone of diffusion models and in substance enable them to generate high-quality images compared with other generative models, where nearly all U-Nets in pre-trained diffusion models have the same number of parameters, except a small number of them, such as U-Nets in denoising diffusion implicit models (DDIMs) (Song JM et al., 2021). Nevertheless, the large number of parameters in U-Nets still prevents diffusion models from achieving efficient generation and purification.

With the purpose of obtaining efficient and high-quality purification and generation with diffusion models, a majority of existing solutions are in knowledge distillation (Meng et al., 2023; Song Y et al., 2023). For these methods, the goal is to reduce the number of iterative steps to accelerate the generation process, where the student models are diffusion models. In practice, a limited number of steps in the student models can hardly achieve the same performance as the teacher models (Song Y et al., 2023). These knowledge distillation methods did not consider the number of parameters or the multi-dimensional structural information in data. Hence, the qualitative performance of compressed models can easily be unrealistic. Besides, LoRA as a fine-tuning method for pre-trained diffusion models was recently proposed and it relies on matrix factorization, where the number of parameters was reduced and the two-dimensional structural information was tackled (Hu et al., 2022). However, when the pre-trained diffusion models are unavailable or when there is complicated multi-dimensional structural information, LoRA will not be so effective and other methods are demanded.

Given the aforementioned problems in the scalability of diffusion models, we propose to compress diffusion models for purification and evaluate their performance on purification tasks. In specific, we design the tensor denoising diffusion purifier (TendiffPure), where we tensorize the convolution kernels in U-Nets using tensor-train (TT) decomposition (Oseledets, 2011) as shown in Fig. 1, enhancing or at least not jeopardizing the purification quality and reducing the space complexity from $O(N^2)$ to $O(NR^2)$ with usually $R \leq 4$ as the TT rank and N as the number of channels, especially for noisy or perturbed images (Li et al., 2019). This tensorization for compression distinguishes TendiffPure from knowledge distillation methods for diffusion models. We conduct three experiments on CIFAR-10 (Krizhevsky and Hinton, 2009), Fashion-MNIST (Xiao et al., 2017), and MNIST (LeCun et al., 1998) datasets separately, on two noises and one adversarial attack: Gaussian noises, salt and pepper noises, and AutoAttack (Croce and Hein, 2020).

2 Background

2.1 Diffusion models for purification

Purification is to eliminate noises and adversarial perturbations in data using generative models before classification. Unlike other defense methods, purification does not assume the forms of noises,

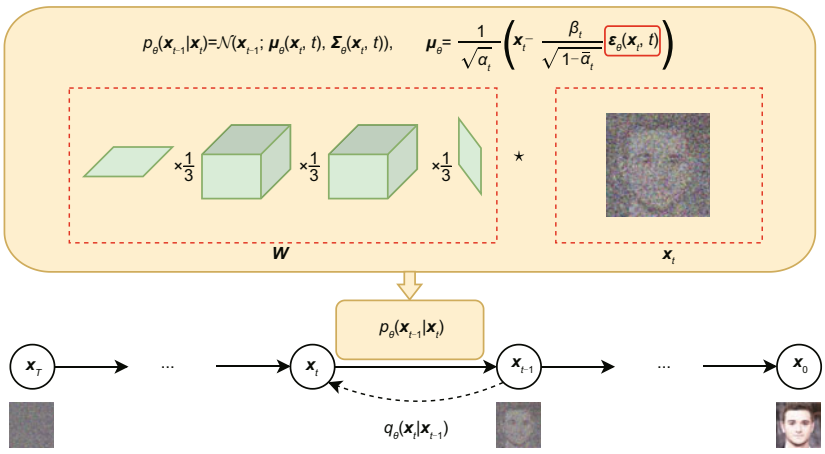

Fig. 1 A brief summary of TendiffPure

adversarial attacks, and classification models. Hence, generative purification models do not require retraining of classifiers and are not trained with threat models. Diffusion models as emerging generative models have been recently scrutinized for purification (Nie et al., 2022) due to their extraordinary generative power. They purify noised or adversarially perturbed data in two phases. First, in the forward process of diffusion models, Gaussian noises are iteratively added to the noised or adversarially perturbed data until they become Gaussian noises as well. Afterwards, in the reverse process, they are denoised iteratively to generate the purified data. Hence, the noises or adversarial perturbations are eliminated. Note that for datasets on which there are diffusion models pre-trained, we can directly use the pre-trained diffusion models for purification, and hence no training process is required. Besides, for those without pre-trained diffusion models, we need to first train diffusion models on clean, i.e., unperturbed, data, and then use these trained diffusion models for purification.

Diffusion models have been the prevalent generative model in recent years. They impress the machine learning and deep learning community with their powerfulness on sample quality, sample diversity, and mode coverage (Ho et al., 2020; Dhariwal and Nichol, 2021; Song JM et al., 2021; Vahdat et al., 2021). Benefiting from these advantages, they become appealing tools for purification, for example, DiffPure (Nie et al., 2022), where noises and even adversarial attacks in the perturbed data $\boldsymbol{x}_{\mathrm{a}} \in \mathbb{R}^d$, $\boldsymbol{x}_{\mathrm{a}} \sim q(\boldsymbol{x})$ can be removed by diffusion models. The denoised or purified data should be as close to the clean data $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{x} \sim p(\boldsymbol{x})$ as possible. A typical diffusion model consists of two procedures: forward process and reverse process. The forward process progressively injects Gaussian noises to the data where the perturbed data $\boldsymbol{x}_{\mathrm{a}}$ are diffused towards a noise distribution. For a discrete diffusion model, its forward process is formulated as

$$\begin{cases} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}), \\ q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \end{cases} \tag{1}$$

where $t = 1, 2, \cdots, T$ is the step to add the small amount of Gaussian noises and $\boldsymbol{x}_0 = \boldsymbol{x}_{\mathrm{a}}$. The step size is controlled by the fixed variance schedule $\beta_t \in (0, 1)$ $(t = 1, 2, \cdots, T)$, where $\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}_t$, $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Usually the reparameterization trick is applied to sample $\boldsymbol{x}_t$ at any arbitrary time point t, where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. Hence, $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\bar{\boldsymbol{\epsilon}}_t$, $\bar{\boldsymbol{\epsilon}}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. At the final step T, where T is large enough, $\boldsymbol{x}_T$ follows a standard Gaussian distribution, i.e., $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. For the reverse process, the Gaussian noises are gradually removed from $\boldsymbol{x}_T$, and hence the denoised or purified image $\hat{\boldsymbol{x}}_0 \in \mathbb{R}^d$ is generated at the end of the reverse process, where $\hat{\boldsymbol{x}}_0 \sim p(\boldsymbol{x})$. Ideally, the distribution of the denoised images $\{\hat{\boldsymbol{x}}_t\}_{t=1}^{T}$ is the same as that in the forward process $\{\boldsymbol{x}_t\}_{t=1}^{T}$. $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ is a model to approximate $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$, so we can avoid to use the entire dataset. In specific,

$$\begin{cases} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}\Big(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)\Big), \\ p_\theta(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t). \end{cases} \tag{2}$$

Instead of predicting $\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)$, which is a linear combination of $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ and $\boldsymbol{x}_t$, practically it is common to predict the noise component as part of $\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)$ using the noise predictor U-Net $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ (Ho et al., 2020). θ is the parameter for describing the mean and variance. The covariance predictor $\boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)$ can be learnable parameters for enhanced model quality (Nichol and Dhariwal, 2021). To thoroughly remove the noise or adversarial attack and keep the semantic information, Nie et al. (2022) proposed to add Gaussian noise in $t^* \in (0, T]$ steps.

2.2 Tensor decomposition

Tensor decomposition and tensor networks are prevalent workhorses for multi-dimensional data analysis to capture their spatial structural information, to reduce the number of model parameters, and to avoid the curse of dimensionality issue, including images (Luo et al., 2022). Here we refer to a multi-dimensional array as a tensor, where the number of "aspects" of a tensor is its order and the aspects are the modes of this tensor; for example, a $1024 \times 768 \times 3$ image is a 3^{rd}-order tensor with the sizes of mode 1, mode 2, and mode 3 being 1024, 768, and 3, respectively. The key of tensor decomposition and tensor networks is to dissect a tensor into the sum of products of vectors as CANDECOMP/PARAFAC (CP) decomposition (Carroll and Chang, 1970), matrices and tensors as Tucker decomposition (Hitchcock, 1927; Tucker, 1966), small-sized tensors such

as TT decomposition (Oseledets, 2011) and tensor ring decomposition (Zhao et al., 2016), and tensor networks such as multi-scale entanglement renormalization ansatz (MERA) (Giovannetti et al., 2008). Among them, TT decomposition demonstrates its prevalence in a number of deep learning models for compression because of its low space complexity and capabilities of improving the performance of deep learning models (Su et al., 2020). In specific, TT decomposition considers a D^{th}-order tensor $\boldsymbol{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_D}$ as the product of D 3^{rd}-order tensors $\boldsymbol{X}_d \in \mathbb{R}^{R_{d-1} \times I_d \times R_d}$, $d = 1, 2, \cdots, D$, with the rank R_d much smaller than the mode size I_d: $\boldsymbol{Y} = \boldsymbol{X}_1 \times_3^1 \boldsymbol{X}_2 \times \cdots \times_3^1 \boldsymbol{X}_D$. Here, $\boldsymbol{X}_d \times_3^1 \boldsymbol{X}_{d+1}$ is the contraction of mode 3 of $\boldsymbol{X}_d$ and mode 1 of $\boldsymbol{X}_{d+1}$. Note that for $\boldsymbol{X}_1$ and $\boldsymbol{X}_D$, $R_0 = R_D = 1$.

3 Tensorizing diffusion models for purification

As aforementioned, we aim to compress the diffusion models from the perspective of reducing the parameter size, at least to attain similar performance of the uncompressed diffusion model on image denoising and purification tasks, i.e., using generative models to remove perturbations in data including adversarial attacks. Therefore, we propose TendiffPure which is a convolutional TT denoising diffusion model.

In each step of a generic diffusion model as in Eqs. (1) and (2), the key backbone is the U-Net $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ in the reverse process. Hence, it provides the potential to compress the diffusion models by reducing the number of parameters of the U-Net. Note that the U-Net at each step of the reverse process shares the same parameters. For the U-Net $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$, we compress it as

$$\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) = \text{ConvTTUNet}(\boldsymbol{x}_t, t). \tag{3}$$

For $\text{ConvTTUNet}(\boldsymbol{x}_t, t)$, each convolution kernel is parameterized using TT decomposition. In existing diffusion models, U-Nets often employ $2D$ convolution kernels, where each convolutional kernel is $\boldsymbol{W}_i \in \mathbb{R}^{O_i \times C_i \times K_i \times D_i}$, where O_i is the number of output channels, C_i is the number of input channels, K_i is the first kernel size, and D_i is the second kernel size. In TendiffPure, we decompose these 4^{th}-order tensors into the following TT cores:

$$\boldsymbol{W}_i = \boldsymbol{U}_1 \times_3^1 \boldsymbol{U}_2 \times_3^1 \boldsymbol{U}_3 \times_3^1 \boldsymbol{U}_4, \tag{4}$$

where $\boldsymbol{U}_1 \in \mathbb{R}^{1 \times O_i \times R_{1,i}}$, $\boldsymbol{U}_2 \in \mathbb{R}^{R_{1,i} \times C_i \times R_{2,i}}$, $\boldsymbol{U}_3 \in \mathbb{R}^{R_{2,i} \times K_i \times R_{3,i}}$, and $\boldsymbol{U}_4 \in \mathbb{R}^{R_{3,i} \times D_i \times 1}$ as demonstrated in Fig. 2. This parameterization follows the standard TT decomposition in Section 2.2, where $R_{0,i} = R_{4,i} = 1$. Hence, the space complexity is reduced from $O(N^2)$ to $O(NR^2)$.

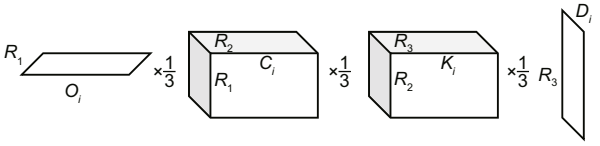

Fig. 2 Convolution tensor-train kernels of TendiffPure

Practically, $R_{0,i}$ can equal the number of input channels. Hence, we have parameterization of U-Nets as

$$\boldsymbol{W}_i = \boldsymbol{U}_1 \times_3^1 \boldsymbol{U}_2 \times_3^1 \boldsymbol{U}_3, \tag{5}$$

where $\boldsymbol{U}_1 \in \mathbb{R}^{O_i \times C_i \times R_{1,i}}$, $\boldsymbol{U}_2 \in \mathbb{R}^{R_{1,i} \times K_i \times R_{2,i}}$, and $\boldsymbol{U}_3 \in \mathbb{R}^{R_{2,i} \times D_i \times 1}$. We allow for a more generic parameterization, where the convolution kernels are decomposed into two TT cores, i.e.,

$$\boldsymbol{W}_i = \boldsymbol{U}_1 \times_4^1 \boldsymbol{U}_2, \tag{6}$$

with $\boldsymbol{U}_1 \in \mathbb{R}^{1 \times O_i \times C_i \times K_i \times R_{1,i}}$ and $\boldsymbol{U}_2 \in \mathbb{R}^{R_{1,i} \times D_i \times 1}$. Note that for all three decomposition schemes, the convolution kernels $\boldsymbol{W}_i$ are squeezed to remove the modes with size 1 for programming. We design these three decomposition schemes to enable a wider range of choices of ranks of TT cores, as the performance of the decomposed model can be sensitive to the ranks of parameters, and we aim to attain the optimal ranks. At the end, each convolution operation in the convolutional TT U-Nets is defined as

$$\boldsymbol{h}_1 = \text{ReLU}(\boldsymbol{W}_1 \star \boldsymbol{x}_t),\ \boldsymbol{h}_i = \text{ReLU}(\boldsymbol{W}_i \star \boldsymbol{h}_{i-1}), \tag{7}$$

where "$\star$" represents the convolution.

Building on these convolutional TT U-Nets as backbones, the proposed TendiffPure is in substance a convolutional TT denoising diffusion model. We follow the general architecture of the denoising diffusion probabilistic model (DDPM) to remove the perturbations, including the adversarial attacks. Instead of completing the forward process, we add only Gaussian noises until step t^*, where $t^* < T$, inspired by Nie et al. (2022). Hence, we can control the amount of Gaussian noises added to ensure that

the perturbations can be properly removed and that the semantic information is not destroyed in the denoised or purified images. In our case, we use the search methods to find the optimal t^*. These search methods include the commonly applied grid search and random search for hyperparameter tuning. For the experiments, we use grid search to seek t^* for its simplicity.

Furthermore, we recognize that low rankness might be related to the robustness of diffusion models (Nie et al., 2022). In particular, according to Theorem 3.2 in DiffPure (Nie et al., 2022), the l_2 distance between the clean data $\boldsymbol{x}$ and the purified data $\hat{\boldsymbol{x}}_0$ is

$$\|\hat{\boldsymbol{x}}_0-\boldsymbol{x}\|_2\le\|\boldsymbol{\epsilon}_{\mathrm{a}}\|_2+\gamma(t^*)C_{\mathrm{s}}+\sqrt{\exp(2\gamma(t^*))-1}\sqrt{2d+4\sqrt{d\log\frac{1}{\delta}}+4\log\frac{1}{\delta}} \quad (8)$$

for continuous-time diffusion models, where the forward and reverse processes are stochastic differential equations (SDEs), $\gamma(t^*)$ is a constant with $\gamma(t^*)=\int_0^{t^*}\frac{1}{2}\beta_s\mathrm{d}s$, $\boldsymbol{\epsilon}_{\mathrm{a}}$ is the adversarial attack or noises, and the score function $\boldsymbol{s}_\theta(\boldsymbol{x},t)=-\frac{1}{\sqrt{1-\bar{\alpha}}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t)$ of diffusion models is bounded: $\|\boldsymbol{s}_\theta(\boldsymbol{x},t)\|_2\le\frac{1}{2}C_{\mathrm{s}}$ with C_{s} as a constant. Note that the discrete-time diffusion model is approximately equivalent to continuous-time diffusion models when generating the purified data $\hat{\boldsymbol{x}}_0$. The only difference between them is whether t can be a continuous value. Here, d refers to the dimensionality of $\boldsymbol{x}$. Reducing the rank of parameters of $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t)$ can be considered as projecting $\boldsymbol{x}_t$ into a lower-dimensional space and hence reducing d. Therefore, it is possible that applying TT parameterization of $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t)$ can lower the ranks of parameters and in consequence might tighten the bound to improve the robustness. However, different TT ranks may have different impacts on the purification results, and we aim to investigate them in the experiments.

4 Experiments

4.1 Experimental settings

4.1.1 Datasets and network architectures

With the purpose of investigating the numerical performance of the proposed TendiffPure, we implement experiments on three datasets: CIFAR-10 (Krizhevsky and Hinton, 2009), Fashion-MNIST (Xiao et al., 2017), and MNIST (LeCun et al., 1998). After conducting the purification or denoising tasks, we intend to investigate if the purified images by the models are close to the clean images enough. Hence, we harness the pre-trained classifiers ResNet56 and LeNet, where ResNet56 is for the CIFAR-10 dataset and LeNet is for the Fashion-MNIST and MNIST datasets. Then we use them to classify the purified or denoised images. If the purified or denoised images can be classified into their original classes by the classifier, it is quantitatively close enough to the clean image. Note that for all the diffusion models in our experiments, we employ the classifier guidance.

4.1.2 Noises and adversarial attacks

We add two different noises, Gaussian noise and salt and pepper noise (S&P noise), and one adversarial attack, AutoAttack, to each of CIFAR-10, Fashion-MNIST, and MNIST datasets. The Gaussian noise level is 51, whereas the proportion of S&P noise added in images is 15%. In terms of the adversarial attack, AutoAttack ℓ_2 threat models are commonly used (Croce and Hein, 2020). Here we use the STANDARD version of AutoAttack. It consists of $\mathrm{APGD_{CE}}$ (which does not have random starts), the targeted version of APGD ($\mathrm{APGD^T}$) as the difference of logit ratio loss handling a model with a minimum of four classes, the targeted version of the FAB attack ($\mathrm{FAB^T}$), and the Square Attack as a score-based blackbox attack for norm-bounded perturbations. In practice, the STANDARD version AutoAttack actually makes stronger attacks (Nie et al., 2022). For AutoAttack, we evaluate TendiffPure against the ℓ_2 threat model with $\epsilon=0.5$.

4.1.3 Baselines

We compare our proposed TendiffPure with two other diffusion models, DDPM (Ho et al., 2020) and DDIM (Song JM et al., 2021), which are the core of nearly all existing diffusion models. Note that we employ two settings of the diffusion timesteps for DDPM and DDIM as $t^*\in\mathbb{N}^+$ ($t^*\le T$) and T, and we present the better results between the two settings. The reason is that we aim to follow the vital diffusion model for purification, DiffPure, where the amount of Gaussian noise is carefully chosen to ensure that the noise or adversarial attacks

in the images can be eliminated and that the label semantics of the purified images is not destroyed. As the discrete version of DiffPure is DDPM with the diffusion timestep t^*, we emphasize DDPM as a discrete DiffPure in the experimental results if its performance under setting t^* is better than that under T.

4.1.4 Evaluation criteria

1. Quantitative criterion

The quantitative evaluation metrics are the standard accuracy, which measures the generative power, and the robust accuracy, which shows both the generative power and the robustness of purification models. To obtain the robust accuracy, the perturbed and adversarial examples are the input of purification models. Once the purification models produce the purified data, these purified data are classified by the classification models whose output is the classification accuracy, i.e., the robust accuracy. To obtain the standard accuracy, it follows the same procedure as what is for the robust accuracy, except that the input data of purification models are clean data without adding noises or adversarial attacks. For the quantitative results, we run the experiments multiple times. Then we report the average standard accuracy and robust accuracy with their error bars. We use the aforementioned classifiers to test if TendiffPure is able to sufficiently remove the noises and adversarial attacks and meanwhile preserve the label semantics of images with the reduced number of parameters compared with the baselines.

2. Qualitative criterion

The performance of TendiffPure is also evaluated in the qualitative perspective. Whether as a human we agree the purified or denoised images by TendiffPure to be more realistic than those purified by the baselines is a vital criterion to evaluate the performance of TendiffPure. Hence, we present the purified images by TendiffPure. Those generated by DDPM with t^* as discrete DiffPure or T as DDPM of purification are presented, where the images with the higher quality are selected. Note that we decide not to present the results generated by DDIM, because of its incapability of removing the noises and adversarial attacks, even compared with DDPM. This is indicated in the quantitative results.

4.2 Experimental result analysis

4.2.1 Quantitative result analysis

1. Comparison with baselines

To begin with, we scrutinize the quantitative performance of TendiffPure compared with those of the baseline models. As aforementioned, a higher robust accuracy produced by the pre-trained classifier ResNet56 or LeNet indicates that the denoised or purified images are closer to the clean images. Table 1 demonstrates that for the CIFAR-10 dataset, the proposed TendiffPure outperforms the baseline diffusion models on the Gaussian noise, S&P noise, and AutoAttack. It also shows that the TT parameterization in TendiffPure successfully captures the multi-dimensional spatial structural information in images and enhances the performance of diffusion models in denoising and purification tasks, along with the reduction of the number of parameters. We can draw the same conclusions from the results on the Fashion-MNIST dataset (Table 2). However, for the results on the MNIST dataset demonstrated in Table 3, TendiffPure produces the purified images with the highest quality in terms of the classification accuracy, except the robust accuracy under Gaussian noises where DDPM (DiffPure) ranks the first. The possible reason is that tensor decomposition methods prefer spatially complicated data, whereas the MNIST dataset contains only handwritten digits with simple spatial information compared with Fashion-MNIST and CIFAR-10 datasets.

2. Ablation studies

We are interested in the effect of TT ranks, i.e., $R_{d,i}$'s, on the purification or denoising results, because they can reveal how much compression can be beneficial to the purification or denoising performance of TendiffPure. Tables 4–6 indicate that TendiffPure prefers TT parameterization as in Eq. (5) with a smaller number of parameters reduced, in specific, with the compression rates being 44.29%, 22.78%, and 22.78% for CIFAR-10, Fashion-MNIST, and MNIST respectively, where the compression rates are computed as the number of parameters of TendiffPure divided by the number of parameters of DDPM (DiffPure). It unveils that in practice, it may not be beneficial to dissect the convolution kernel in terms of the product of the numbers of input channels and output channels, for purification or denoising tasks. We observe that the ranges of the

Table 1 Purification performance of TendiffPure on CIFAR-10 evaluated by the pre-trained ResNet56 classifier

Model	Standard accuracy (%)	Robust accuracy (%)		
		Gaussian noise	Salt and pepper noise	AutoAttack
DDPM	93.34 ± 0.13	93.39 ± 0.34	92.89 ± 0.66	91.31 ± 0.44
DDIM	54.41 ± 0.39	54.85 ± 0.40	37.83 ± 0.25	44.86 ± 0.31
TendiffPure (ours)	$\mathbf{95.62 \pm 0.30}$	$\mathbf{95.02 \pm 0.47}$	$\mathbf{95.65 \pm 0.17}$	$\mathbf{92.45 \pm 0.24}$

Best results are in bold

Table 2 Purification performance of TendiffPure on Fashion-MNIST evaluated by the pre-trained LeNet classifier

Model	Standard accuracy (%)	Robust accuracy (%)		
		Gaussian noise	Salt and pepper noise	AutoAttack
DDPM	93.69 ± 0.15	92.79 ± 0.34	92.38 ± 0.30	90.72 ± 0.39
DDIM	69.60 ± 0.24	47.75 ± 0.42	66.65 ± 0.37	68.86 ± 0.36
TendiffPure (ours)	$\mathbf{95.64 \pm 0.32}$	$\mathbf{93.62 \pm 0.06}$	$\mathbf{94.82 \pm 0.36}$	$\mathbf{92.43 \pm 0.22}$

Best results are in bold

Table 3 Purification performance of TendiffPure on MNIST evaluated by the pre-trained LeNet classifier

Model	Standard accuracy (%)	Robust accuracy (%)		
		Gaussian noise	Salt and pepper noise	AutoAttack
DDPM	98.97 ± 0.52	$\mathbf{98.93 \pm 0.05}$	98.03 ± 0.34	99.27 ± 0.22
DDIM	81.25 ± 0.30	62.18 ± 0.42	22.30 ± 0.25	83.17 ± 0.41
TendiffPure (ours)	$\mathbf{99.27 \pm 0.22}$	98.68 ± 0.29	$\mathbf{98.70 \pm 0.12}$	$\mathbf{99.48 \pm 0.08}$

Best results are in bold

Table 4 Ablation studies of TendiffPure on CIFAR-10 evaluated by the pre-trained ResNet56 classifier

Rank	Standard accuracy (%)	Robust accuracy (%)		
		Gaussian noise	Salt and pepper noise	AutoAttack
$(3, 3, 3)$	50.42 ± 0.36	50.97 ± 0.17	49.56 ± 0.18	46.01 ± 0.38
$(4, 4, 4)$	67.45 ± 0.08	67.76 ± 0.16	64.60 ± 0.62	57.82 ± 0.47
$(4, 3, 4)$	63.74 ± 0.29	61.51 ± 0.61	65.20 ± 0.44	58.33 ± 0.30
$(4, 4)$	93.11 ± 0.57	94.04 ± 0.32	94.89 ± 0.24	$\mathbf{92.45 \pm 0.24}$
$(3, 3)$	92.56 ± 0.66	91.36 ± 0.37	91.80 ± 0.29	91.31 ± 0.24
$(3, 4)$	$\mathbf{95.62 \pm 0.30}$	$\mathbf{95.02 \pm 0.47}$	$\mathbf{95.65 \pm 0.17}$	91.03 ± 0.37
$(2, 3)$	91.71 ± 0.22	91.54 ± 0.25	91.36 ± 0.42	90.02 ± 0.57
(2)	92.66 ± 0.17	92.99 ± 0.38	91.65 ± 0.51	90.87 ± 0.40

Best results are in bold

Table 5 Ablation studies of TendiffPure on Fashion-MNIST evaluated by the pre-trained LeNet classifier

Rank	Standard accuracy (%)	Robust accuracy (%)		
		Gaussian noise	Salt and pepper noise	AutoAttack
$(3, 3, 3)$	57.99 ± 0.08	41.34 ± 0.34	37.78 ± 0.42	55.71 ± 0.35
$(4, 4, 4)$	82.32 ± 0.61	66.83 ± 0.44	60.35 ± 0.26	74.20 ± 0.13
$(3, 4, 3)$	81.80 ± 0.28	79.10 ± 0.35	72.41 ± 0.36	83.71 ± 0.68
$(4, 4)$	92.50 ± 0.29	92.81 ± 0.16	92.19 ± 0.39	89.85 ± 0.51
$(4, 3)$	93.08 ± 0.40	92.72 ± 0.69	91.45 ± 0.67	91.28 ± 0.32
$(3, 3)$	95.39 ± 0.38	$\mathbf{93.62 \pm 0.06}$	$\mathbf{94.82 \pm 0.36}$	$\mathbf{92.43 \pm 0.22}$
(3)	93.65 ± 0.34	93.28 ± 0.22	93.02 ± 0.17	92.37 ± 0.26
(2)	$\mathbf{95.64 \pm 0.32}$	93.29 ± 0.47	93.03 ± 0.45	92.04 ± 0.21

Best results are in bold

Table 6 Ablation studies of TendiffPure on MNIST evaluated by the pre-trained LeNet classifier

Rank	Standard accuracy (%)	Robust accuracy (%)		
		Gaussian noise	Salt and pepper noise	AutoAttack
(3, 3, 3)	90.77 ± 0.22	68.51 ± 0.08	63.51 ± 0.38	91.67 ± 0.14
(4, 4, 4)	92.83 ± 0.30	74.75 ± 0.22	71.16 ± 0.27	91.93 ± 0.35
(3, 4, 3)	94.41 ± 0.14	79.16 ± 0.48	79.21 ± 0.20	94.17 ± 0.44
(4, 4)	97.36 ± 0.43	91.57 ± 0.30	90.61 ± 0.43	97.28 ± 0.29
(3, 3)	97.74 ± 0.28	95.05 ± 0.12	93.75 ± 0.39	97.12 ± 0.26
(4, 3)	$\mathbf{99.27 \pm 0.22}$	$\mathbf{98.68 \pm 0.29}$	$\mathbf{98.70 \pm 0.12}$	$\mathbf{99.48 \pm 0.08}$
(2, 3)	98.94 ± 0.24	$\mathbf{98.68 \pm 0.15}$	98.08 ± 0.30	98.78 ± 0.41
(2)	98.24 ± 0.27	98.34 ± 0.20	97.38 ± 0.30	98.86 ± 0.41

Best results are in bold

standard accuracy and the robust accuracy among different TT ranks are larger than the difference between those of DiffPure and TendiffPure with the optimal TT ranks. For all three datasets, TendiffPure has the lowest standard accuracy and robust accuracy at rank (3, 3, 3) according to ablation studies in Tables 4–6, where TendiffPure performs worse than DiffPure using either DDPM or DDIM. It is possible that the effect of TT parameterization is sensitive to the TT ranks, which can significantly affect the robustness of diffusion models. In specific, the lowest standard accuracy and robust accuracy often occur at the higher TT ranks, which can be an interesting finding about the relationship between low rankness of parameters and robustness of diffusion models. In conclusion, these findings can pave the way for our future study on theoretically analysis of how to decompose convolution kernels using tensor decomposition or tensor networks to compress U-Nets in diffusion models properly.

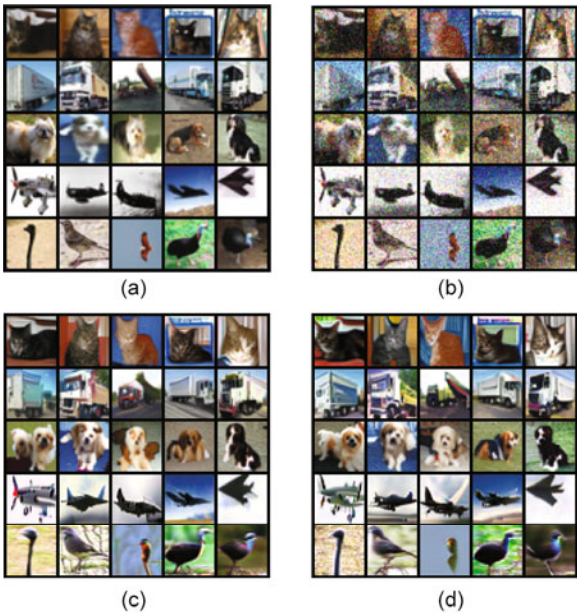

Fig. 3 Selected purified or denoised images by TendiffPure on the Gaussian noised CIFAR-10 dataset compared with DDPM (DiffPure): (a) original images; (b) Gaussian noised; (c) DDPM (DiffPure); (d) TendiffPure

4.2.2 Qualitative result analysis

As for the qualitative performance of TendiffPure, we present the purified or denoised images as a subset of the purified or denoised CIFAR-10 dataset on Gaussian noise, S&P noise, and AutoAttack. In Fig. 3, TendiffPure generates evidently more realistic images which are closer to the original images, i.e., clean images. In specific, DDPM as a discrete DiffPure even produces an image of a dog with two heads in the third row and second column of Fig. 3c. For the case with S&P noise as shown in Fig. 4, although DDPM (DiffPure) and TendiffPure both demonstrate their limitations on removal of noises added on structurally complicated images such as toads and frogs, TendiffPure still preserves more structural information with a largely reduced number of parameters and possesses more robustness. It is consistent with the qualitative performance of TendiffPure for AutoAttack perturbed on the CIFAR-10 dataset. TendiffPure also eliminates this adversarial attack and generates images with more realism than DDPM (DiffPure) as in Fig. 5.

5 Conclusions

To enhance the efficacy of diffusion models in purification, we propose TendiffPure as a diffusion model with convolutional TT U-Net backbones. Compared with existing methods,

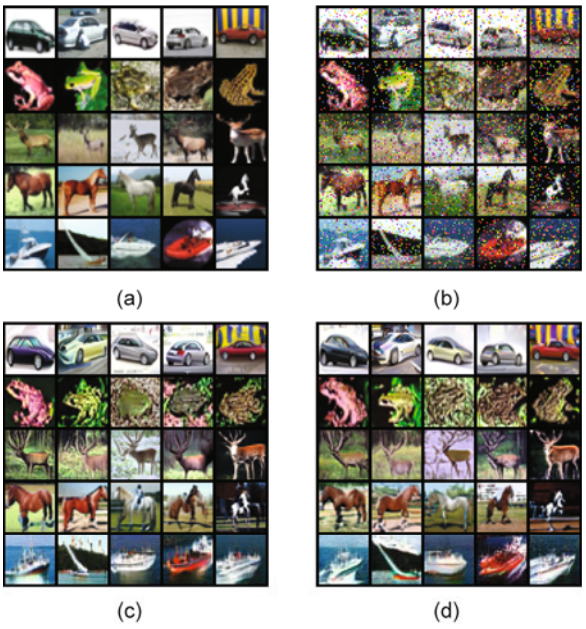

Fig. 4 Selected purified or denoised images by TendiffPure on the CIFAR-10 dataset with salt and pepper noise compared with DDPM (DiffPure): (a) original images; (b) S&P noised; (c) DDPM (DiffPure); (d) TendiffPure

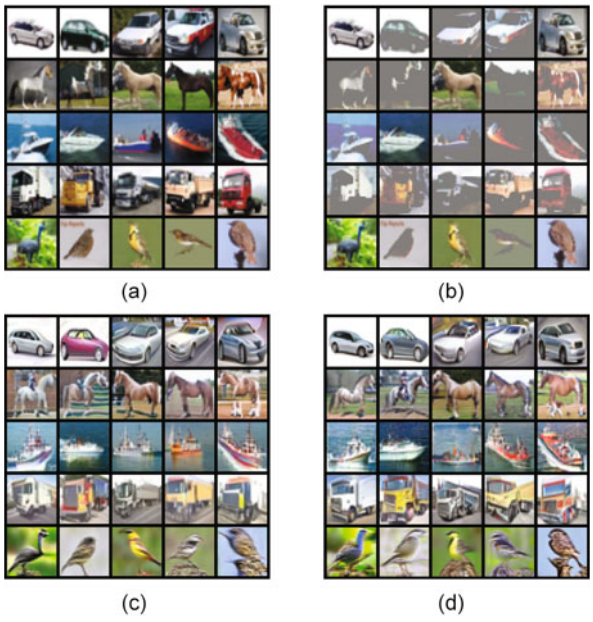

Fig. 5 Selected purified or denoised images by TendiffPure on the CIFAR-10 dataset with AutoAttack compared with DDPM (DiffPure): (a) original images; (b) AutoAttacked; (c) DDPM (DiffPure); (d) TendiffPure

TendiffPure largely reduces the space complexity, and is able to analyze spatially complicated information in multi-dimensional data such as images. Our experimental results on CIFAR-10, Fashion-MNIST, and MNIST for Gaussian and S&P noises and AutoAttack show that TendiffPure outperforms existing diffusion models for purification or denoising tasks, quantitatively and qualitatively.

However, there are still potential limitations of TendiffPure. At this stage, how the TT ranks affect the purification or denoising performance is not theoretically studied. Hence, other than grid search, there is no better method to provide an optimal scheme to decide how to decompose the convolution kernel in U-Nets as backbones of diffusion models using TT decomposition or even tensor decomposition or tensor networks. In the future work, we aim to theoretically analyze the effect of tensor decomposition methods on diffusion models for purification.

Contributors

Mingyuan BAI designed the research. Derun ZHOU processed the data. Mingyuan BAI drafted the paper. Qibin ZHAO helped organize the paper. Mingyuan BAI and Derun ZHOU revised and finalized the paper.

Compliance with ethics guidelines

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Carroll JD, Chang JJ, 1970. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283-319. https://doi.org/10.1007/BF02310791

Croce F, Hein M, 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. Proc 37$^{\text{th}}$ Int Conf on Machine Learning, Article 206.

Dhariwal P, Nichol A, 2021. Diffusion models beat GANs on image synthesis. Proc 35$^{\text{th}}$ Conf on Neural Information Processing Systems, p.8780-8794.

Gao Q, Li ZL, Zhang JP, et al., 2023. CoreDiff: contextual error-modulated generalized diffusion model for low-dose CT denoising and generalization. *IEEE Trans Med Imag*, early access. https://doi.org/10.1109/TMI.2023.3320812

Giovannetti V, Montangero S, Fazio R, 2008. Quantum multiscale entanglement renormalization ansatz channels. *Phys Rev Lett*, 101(18):180503. https://doi.org/10.1103/PhysRevLett.101.180503

Hitchcock FL, 1927. The expression of a tensor or a polyadic as a sum of products. *J Math Phys*, 6(1-4):164-189. https://doi.org/10.1002/sapm192761164

Ho J, Salimans T, 2021. Classifier-free diffusion guidance. Proc Workshop on Deep Generative Models and Downstream Applications.

Ho J, Jain A, Abbeel P, 2020. Denoising diffusion probabilistic models. Proc 34th Int Conf on Neural Information Processing Systems, Article 574.

Hu EJ, Shen YL, Wallis P, et al., 2022. LoRA: low-rank adaptation of large language models. Proc 10th Int Conf on Learning Representations.

Krizhevsky A, Hinton G, 2009. Learning Multiple Layers of Features from Tiny Images. Technical Report. University of Toronto, Toronto, Canada.

LeCun Y, Bottou L, Bengio Y, et al., 1998. Gradient-based learning applied to document recognition. *Proc IEEE*, 86(11):2278-2324. https://doi.org/10.1109/5.726791

Li C, Sun Z, Yu JS, et al., 2019. Low-rank embedding of kernels in convolutional neural networks under random shuffling. Proc IEEE Int Conf on Acoustics, p.3022-3026. https://doi.org/10.1109/ICASSP.2019.8682265

Luo YS, Zhao XL, Meng DY, et al., 2022. HLRTF: hierarchical low-rank tensor factorization for inverse problems in multi-dimensional imaging. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.19281-19290. https://doi.org/10.1109/CVPR52688.2022.01870

Meng CL, Rombach R, Gao RQ, et al., 2023. On distillation of guided diffusion models. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.14297-14306. https://doi.org/10.1109/CVPR52729.2023.01374

Nichol AQ, Dhariwal P, 2021. Improved denoising diffusion probabilistic models. Proc 38th Int Conf on Machine Learning, p.8162-8171.

Nie WL, Guo B, Huang YJ, et al., 2022. Diffusion models for adversarial purification. Proc 39th Int Conf on Machine Learning, p.16805-16827.

Oseledets IV, 2011. Tensor-train decomposition. *SIAM J Sci Comput*, 33(5):2295-2317. https://doi.org/10.1137/090752286

Ronneberger O, Fischer P, Brox T, 2015. U-Net: convolutional networks for biomedical image segmentation. Proc 18th Int Conf on Medical Image Computing and Computer-Assisted Intervention, p.234-241. https://doi.org/10.1007/978-3-319-24574-4_28

Song JM, Meng CL, Ermon S, 2021. Denoising diffusion implicit models. Proc 9th Int Conf on Learning Representations.

Song Y, Garg S, Shi JX, et al., 2020. Sliced score matching: a scalable approach to density and score estimation. Proc 35th Uncertainty in Artificial Intelligence Conf, p.574-584.

Song Y, Dhariwal P, Chen M, et al., 2023. Consistency models. Proc 40th Int Conf on Machine Learning, Article 1335.

Su JH, Byeon W, Kossaifi J, et al., 2020. Convolutional tensor-train LSTM for spatio-temporal learning. Proc 34th Int Conf on Neural Information Processing Systems, Article 1150.

Tucker LR, 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279-311. https://doi.org/10.1007/BF02289464

Vahdat A, Kreis K, Kautz J, 2021. Score-based generative modeling in latent space. Proc 35th Conf on Neural Information Processing Systems.

Vincent P, 2011. A connection between score matching and denoising autoencoders. *Neur Comput*, 23(7):1661-1674. https://doi.org/10.1162/NECO_a_00142

Xiao H, Rasul K, Vollgraf R, 2017. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. https://arxiv.org/abs/1708.07747

Zhao QB, Zhou GX, Xie SL, et al., 2016. Tensor ring decomposition. https://arxiv.org/abs/1606.05535