Tendiffpure: Tensorizing Diffusion Models for Purification

Derun Zhou [*1,2] **Mingyuan Bai** [*1] **Qibin Zhao** [1]

Abstract

Diffusion models are effective purification methods to purify the noised or adversarially perturbed examples before feeding them into classifiers. One major limitation of existing diffusion models for purification is low efficiency. Current solutions are knowledge distillation which in fact jeopardizes the generation quality, i.e., the purification performance, because of the small number of generation steps. We propose Tendiffpure as a compressed diffusion model for purification via tensorization. Unlike knowledge distillation methods, we keep the number of generation steps unchanged and directly compress u-nets, the 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 $R \leq 4$. Experimental results show that Tendiffpure can generate high quality purified results more efficiently and outperform the baseline purification methods on CIFAR-10, FashionMNIST and MNIST datasets for two noises and one adversarial attack.

1. Introduction

Diffusion models are ubiquitous generative models in the recent three years in text, image and video generation. They appeal to both academics and practitioners due to their mode coverage, stationary training objective, easy scalability and sample quality (Ho et al., 2020; Song et al., 2021; Dhariwal & Nichol, 2021; Vahdat et al., 2021; Ho & Salimans, 2021). Diffusion models also demonstrate strong capabilities as purification methods.

[*]Equal contribution [1]Tensor Learning Team, RIKEN AIP, Tokyo, Japan [2]School of Environment and Society , Tokyo Institute of Technology, Tokyo, Japan. Correspondence to: Qibin Zhao <qibin.zhao@riken.jp>.

Workshop on New Frontiers in Learning, Control, and Dynamical Systems at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023.

Purification employs generative models to purify images perturbed by noises or adversarial attacks for data preprocessing, followed by classification without retraining the classifier (Shi et al., 2021; Yoon et al., 2021). As aforementioned, the powerful generative capability of diffusion models makes them prevalent methods in purification with state-of-the-art results (Nie et al., 2022), compared with past methods relying on generative adversarial networks (GAN) (Samangouei et al., 2018), autoregressive generative models (Song et al., 2018) and energy-based models (EBMs) (Du & Mordatch, 2019; Grathwohl et al., 2020; Hill et al., 2021). However, diffusion models as purification methods suffer from the slow sampling speed which is caused by the iterative generation process. Also, the images naturally possess the multi-dimensional spatial structures which can be easily neglected by the convolution kernels of u-nets (Ronneberger et al., 2015) which are the common backbones of diffusion models. Furthermore, nearly all u-nets in pretrained diffusion models are of the same large number of parameters with the space complexity $O(N^2)$, except a small number of them, such as u-nets in denoising diffusion implicit models (DDIMs) (Song et al., 2021). This large number of parameters in u-nets prevents diffusion models from achieving efficient generation and purification.

To the best of our knowledge, there is no existing work addressing the efficiency of purification using diffusion models. Nevertheless, with the purpose of obtaining efficient and high-quality generation results of diffusion models, the majority of existing solutions fall in knowledge distillation (Meng et al., 2023; Song et al., 2023). In these methods, the goal is to reduce the number of iterative steps to accelerate the generation process, where the student models are also diffusion models. In practice, limited steps in the student models can hardly achieve the same performance as the teacher models (Song et al., 2023). Furthermore, they did not consider the number of parameters and the multi-dimensional structural information in images. Hence the qualitative performance of compressed models can be reduced significantly.

Given the aforementioned problems in the scalability of diffusion models for purification, we propose the tensor denoising diffusion purifier (Tendiffpure) to compress diffusion models and generate higher-quality purified images. Specificially, we tensorize the convolution kernels in u-nets

using tensor-train (TT) decomposition (Oseledets, 2011), enhancing the purification quality and reducing the space complexity from $O(N^2)$ to $O(NR^2)$ where TT rank $R \leq 4$, especially for noisy or perturbed images (Li et al., 2019), which distinguishes Tendiffpure from knowledge distillation methods for diffusion models. We conduct 3 experiments on CIFAR-10, FashionMNIST and MNIST datasets, respectively on 2 noises and 1 adversarial attack: Gaussian noises, salt and pepper noises and AutoAttack (Croce & Hein, 2020).

2. Background

2.1. Diffusion Models

Benefiting from high sample quality, great sample diversity and large mode coverage, diffusion models become appealing tools for purification, for example, DiffPure (Nie et al., 2022) where the perturbed data $\mathbf{x}_a \in \mathbb{R}^d, \mathbf{x}_a \sim q(\mathbf{x})$ by noises and even adversarial attacks can be purified by diffusion models. The purified image should be as close to the clean data $\mathbf{x} \in \mathbb{R}^d, \mathbf{x} \sim p(\mathbf{x})$ as possible. A typical diffusion model consists of two procedures: the forward process and the reverse process. The forward process progressively injects Gaussian noises to the data where the perturbed data $\mathbf{x}_a$ is diffused towards a noise distribution. For a discrete diffusion model, its forward process is formulated as

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

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

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

Instead of predicting $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ which is a linear combination of $\epsilon_\theta(\mathbf{x}_t, t)$ and $\mathbf{x}_t$, practically it is common to predict the noise component as part of $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ using the noise predictor u-net $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ (Ho et al., 2020). The covariance predictor $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ can be learnable parameters for enhanced model quality (Nichol & Dhariwal, 2021).

2.2. Tensor Decomposition

Tensor decomposition and tensor networks are prevalent workhorses for multi-dimensional data analysis to capture their spatial structural information, reduce the number of model parameters and avoid the curse of dimensionality issue, including images (Luo et al., 2022). Here we refer a multi-dimensional array as a tensor where the number of "aspects" of a tensor is its order and the aspects are the modes. For example, a $1024 \times 768 \times 3$ image is a 3rd-order tensor with the sizes of mode-1, mode-2 and mode-3 are 1024, 768 and 3. The key of tensor decomposition and tensor networks is to dissect a tensor into the product or the sum of products of vectors such as CANDECOMP/PARAFAC (CP) decomposition (Carroll & Chang, 1970), or the product of matrices and tensors such as Tucker decomposition (Hitchcock, 1927; Tucker, 1966), and small-sized tensors such as tensor-train (TT) decomposition (Oseledets, 2011), tensor ring decomposition (Zhao et al., 2016) and tensor networks, for example, 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 model 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 Dth-order tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ as the product of D 3rd-order tensors $\mathcal{X}_d \in \mathbb{R}^{R_{d-1} \times I_d \times R_d}, d = 1, \cdots, D$ with R_d largely smaller than I_d: $\mathcal{Y} = \mathcal{X}_1 \times_3^1 \cdots \times_3^1 \mathcal{X}_D$. Here $\mathcal{X}_d \times_3^1 \mathcal{X}_{d+1}$ is the contraction of mode-3 of $\mathcal{X}_d$ and mode-1 of $\mathcal{X}_{d+1}$. More details of TT decomposition and tensor decomposition/network methods are released in Appendices A and B.

3. Tensorizing Diffusion Models for Purification

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

In each step of a generic diffusion model as in Equations (1) and (2), the key backbone is the u-net $\epsilon_\theta(\mathbf{x}_t, t)$ 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 share the same parameters. For the u-net $\epsilon_\theta(\mathbf{x}_t, t)$, we compress it as

$$\epsilon_\theta(\mathbf{x}_t, t) = \mathrm{ConvTTUNet}(\mathbf{x}_t, t). \tag{3}$$

For $\mathrm{ConvTTUNet}(\mathbf{x}_t, t)$, each convolution kernel is parameterized using TT decomposition. In existing diffusion models, u-nets often employ 2D convolution kernels, where each convolutional kernel is $\mathcal{W}_i \in \mathbb{R}^{O_i \times C_i \times K_i \times D_i}$ 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 the second kernel size. In Tendiffpure, we decompose these 4th-order tensors into the following tensor-trains.

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

where $\mathcal{U}_1 \in \mathbb{R}^{1 \times O_i \times R_{1,i}}$, $\mathcal{U}_2 \in \mathbb{R}^{R_{1,i} \times C_i \times R_{2,i}}$, $\mathcal{U}_3 \in \mathbb{R}^{R_{2,i} \times K_i \times R_{3,i}}$ and $\mathcal{U}_4 \in \mathbb{R}^{R_{3,i} \times D_i \times 1}$. This parameterization follows the standard TT decomposition in Section 2.2 where $R_{0,i} = R_{4,i} = 1$. Hence the space complexity reduces from $O(N^2)$ to $O(NR^2)$ where N is for the number of channels O_i or C_i, and R is the rank of tensor-train cores.

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

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

where $\mathcal{U}_1 \in \mathbb{R}^{O_i \times C_i \times R_{1,i}}$, $\mathcal{U}_2 \in \mathbb{R}^{R_{1,i} \times K_i \times R_{2,i}}$ and $\mathcal{U}_3 \in \mathbb{R}^{R_{2,i} \times D_i \times 1}$. We allow for a more generic parameterization where the convolution kernels are decomposed into two core tensors, i.e., $\mathcal{W}_i = \mathcal{U}_1 \times_4^1 \mathcal{U}_2$ with $\mathcal{U}_1 \in \mathbb{R}^{1 \times O_i \times C_i \times K_i \times R_{1,i}}$ and $\mathcal{U}_2 \in \mathbb{R}^{R_{1,i} \times D_i \times 1}$. Note that for all three decomposition schemes, the convolution kernels $\mathcal{W}_i$ are squeezed to remove the modes with the size 1 for programming. At the end, each convolution operation in the convolutional TT u-nets is defined as

$$\mathbf{h}_1 = \mathrm{ReLU}(\mathcal{W}_1 \star \mathbf{x}_t), \quad \mathbf{h}_i = \mathrm{ReLU}(\mathcal{W}_i \star \mathbf{h}_{i-1}). \tag{6}$$

Building on these convolutional TT u-nets as backbones, the proposed Tendiffpure is in substance a convolutional tensor-train 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 whole forward process, we only add Gaussian noises until the 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 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^*.

4. Experiments

With the purpose of investigating the numerical performance of the proposed Tendiffpure, we design three experiments on three different perturbations: Gaussian noises, salt and pepper noises (S&P noises) and one adversarial attack: AutoAttack on CIFAR-10, FashionMNIST and MNIST datasets, respectively. The Gaussian noise level, i.e., standard deviation, is 51, whereas the proportion of S&P noises added in images is 15%. In terms of the adversarial attack, AutoAttack ℓ_2 threat models are commonly used (Croce & Hein, 2020) and here we use its STANDARD version. In practice, the STANDARD version AutoAttacks actually makes stronger attacks (Nie et al., 2022). For AutoAttack, we evaluate Tendiffpure against the ℓ_2 threat model with $\epsilon = 0.5$. The evaluation metric is robust accuracy which measures the performance of models on the adversarial examples or perturbed examples. We compare our proposed Tendiffpure with two other discrete diffusion models: DiffPure (Nie et al., 2022) and denoising diffusion implicit models (DDIM) (Song et al., 2021) which are the core of nearly all the existing diffusion models. Note that we attempted two settings of the steps in DDIM as t^* and T and we present the higher accuracy. We also incorporate the classifier-free guidance into all diffusion models and use pretrained ResNet56 classifier to evaluate if the purified images are clean enough to be classified in their belonging class. We also aim to scrutinize how ranks of tensor-train cores, i.e., $R_{d,i}$'s affect the performance.

Table 1 indicates that for the CIFAR-10 dataset, the proposed Tendiffpure outperforms the baseline diffusion models on both kinds of noises and AutoAttack. It also demonstrates that the tensor-train 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 on the results on FashionMNIST dataset. However, for the results on the MNIST dataset, DDPM produces the purified images with the highest qualities in terms of the classification accuracy and Tendiffpure has ranks second with relatively the same robust accuracy. 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 FashionMNIST and CIFAR-10 datasets.

5. Conclusions

In order to enhance the efficacy of diffusion models in purification, we propose Tendiffpure as a diffusion model with convolutional tensor-train u-net backbones. Compared with existing methods, Tendiffpure possesses largely reduce the space complexity and is able to analyze spatially more complicate information in multi-dimensional data such as images. Our experimental results on CIFAR-10, FashionM-

Model	Robust Accuracy		
	Gaussian noise	Salt and pepper noise	AutoAttack
DiffPure	93.46%	92.87%	91.31%
DDIM	54.49%	38.09%	44.73%
TendiffDenoiser (Ours)			
Rank (3, 3, 3)	51.10%	49.51%	45.90%
Rank (4, 4, 4)	67.58%	64.65%	58.11%
Rank (4, 3, 4)	61.43%	65.43%	58.59%
Rank (4, 4)	93.75%	94.73%	**92.29%**
Rank (3, 3)	91.41%	91.99%	91.31%
Rank (3, 4)	**94.73%**	**95.70%**	91.41%
Rank (2, 3)	91.50%	91.41%	89.94%
Rank (2)	92.68%	91.41%	90.53%

Table 1. CIFAR10: Classifier: ResNet56

Model	Robust Accuracy		
	Gaussian noise	Salt and pepper noise	AutoAttack
DiffPure	92.72%	92.38%	91.02%
DDIM	48.14%	66.60%	69.14%
TendiffDenoiser (Ours)			
Rank (3, 3, 3)	41.41%	37.70%	55.62%
Rank (4, 4, 4)	66.80%	60.45%	74.32%
Rank (3, 4, 3)	79.20%	72.51%	83.64%
Rank (4, 4)	92.92%	92.04%	90.09%
Rank (4, 3)	92.82%	91.60%	91.60%
Rank (3, 3)	**93.65%**	**94.92%**	**92.68%**
Rank (3)	93.51%	93.12%	92.53%
Rank (2)	93.31%	93.41%	91.80%

Table 2. FashionMNIST: Classifier: LeNet

Model	Robust Accuracy		
	Gaussian noise	Salt and pepper noise	AutoAttack
DiffPure	98.93%	98.34%	**99.46%**
DDIM	62.60%	22.51%	83.64%
TendiffDenoiser (Ours)			
Rank(3, 3, 3)	68.46%	63.62%	91.75%
Rank (4, 4, 4)	74.95%	71.34%	92.33%
Rank (3, 4, 3)	79.44%	79.39%	94.58%
Rank (4, 4)	91.89%	90.72%	97.61%
Rank (3, 3)	95.12%	94.14%	97.41%
Rank (4, 3)	**99.02%**	**98.63%**	99.41%
Rank (2, 3)	98.68%	98.29%	99.12%
Rank (2)	98.34%	97.66%	99.07%

Table 3. MNIST: Classifier: LeNet

NIST and MNIST for Gaussian and S&P noises and AutoAttack show that Tendiffpure outperform the existing diffusion models for purification. In the future work, we aim to theoretically study the effect of tensor decomposition methods on diffusion models on purification.

Acknowledgements

This work was partially supported by JSPS KAKENHI Grant Number 20H04249.

References

Carroll, J. D. and Chang, J.-J. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, September 1970. ISSN 1860-0980. doi: 10.1007/BF02310791.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th International Conference on Machine Learning*, number 206 in ICML'20, pp. 11. JMLR.org, 2020.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021.

Du, Y. and Mordatch, I. Implicit generation and modeling with energy based models. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Giovannetti, V., Montangero, S., and Fazio, R. Quantum multiscale entanglement renormalization ansatz channels. *Phys. Rev. Lett.*, 101:180503, Oct 2008. doi: 10.1103/PhysRevLett.101.180503.

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020.

Hill, M., Mitchell, J. C., and Zhu, S.-C. Stochastic security: Adversarial defense using long-run dynamics of energy-based models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=gwFTuzxJW0.

Hitchcock, F. L. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927. doi: https://doi.org/10.1002/sapm192761164.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020.

Li, C., Sun, Z., Yu, J., Hou, M., and Zhao, Q. Low-rank embedding of kernels in convolutional neural networks under random shuffling. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3022–3026, 2019. doi: 10.1109/ICASSP.2019.8682265.

Luo, Y., Zhao, X., Meng, D., and Jiang, T. Hlrtf: Hierarchical low-rank tensor factorization for inverse problems in multi-dimensional imaging. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19281–19290, 2022. doi: 10.1109/CVPR52688.2022.01870.

Meng, C., Gao, R., Kingma, D. P., Ermon, S., Ho, J., and Salimans, T. On distillation of guided diffusion models. In *to appear in CVPR 2023*. PMLR, 2023. URL https://openreview.net/forum?id=6QHpSQt6VR-.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171. PMLR, 18–24 Jul 2021.

Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., and Anandkumar, A. Diffusion models for adversarial purification. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16805–16827. PMLR, 17–23 Jul 2022.

Oseledets, I. V. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011. doi: 10.1137/090752286.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F. (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

Samangouei, P., Kabkab, M., and Chellappa, R. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018.

Shi, C., Holtz, C., and Mishne, G. Online adversarial purification based on self-supervised learning. In *International Conference on Learning Representations*, 2021.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning*

Representations, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Song, Y., Kim, T., Nowozin, S., Ermon, S., and Kushman, N. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. *arXiv*, 2303.01469, 2023.

Su, J., Byeon, W., Kossaifi, J., Huang, F., Kautz, J., and Anandkumar, A. Convolutional tensor-train lstm for spatio-temporal learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 13714–13726. Curran Associates, Inc., 2020.

Tucker, L. R. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, September 1966. ISSN 1860-0980. doi: 10.1007/BF02289464.

Vahdat, A., Kreis, K., and Kautz, J. Score-based generative modeling in latent space. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, pp. nan, 2021. URL https://openreview.net/forum?id=P9TYG0j-wtG.

Yoon, J., Hwang, S. J., and Lee, J. Adversarial purification with score-based generative models. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12062–12072. PMLR, 18–24 Jul 2021.

Zhao, Q., Zhou, G., Xie, S., Zhang, L., and Cichocki, A. Tensor ring decomposition. *arXiv*, 1606.05535, 2016.

A. Formulation of Tensor-Train Decomposition

As aforementioned in Section 2.2, for a Dth-order tensor, $\mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$, it can be dissected as the product of D 3rd-order tensors with smaller sizes. In specific, one example is that a 3rd-order tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ can be decomposed as the product of $\mathcal{X}_1 \in \mathbb{R}^{R_0 \times I_1 \times R_1}$, $\mathcal{X}_2 \in \mathbb{R}^{R_1 \times I_2 \times R_2}$ and $\mathcal{X}_3 \in \mathbb{R}^{R_2 \times I_3 \times R_3}$ which are often referred to as tensor-train-cores (TT-cores), under TT decomposition paradigm:

$$\mathcal{Y} = \mathcal{X}_1 \times_3^1 \mathcal{X}_2 \times_3^1 \mathcal{X}_3 \tag{7}$$

where $R_0 = R_3 = 1$. Here $\mathcal{X}_{d-1} \times_3^1 \mathcal{X}_d$ means contraction which is a product between tensors $\mathcal{X}_{d-1} \in \mathbb{R}^{R_{d-2} \times I_{d-1} \times R_{d-1}}$ and $\mathcal{X}_d \in \mathbb{R}^{R_{d-1} \times I_d \times R_d}$ with mode-3 of $\mathcal{X}_{d-1}$ contracting mode-1 of $\mathcal{X}_d$. Hence the result has the size $R_{d-2} \times I_{d-1} \times I_d \times R_d$.

B. Space Complexity of Tensor Decomposition and Tensor Networks

Tensor decomposition and tensor network are common methods for parameterization in model compression. Here we compare the space complexity of major tensor decomposition and tensor network methods in the following table, where R is for the tensor core rank. For the original tensor, we consider it in the full or raw tensor format and let it be with the size I^D where I is for the modes and D is for the number of modes, i.e., order.

CP decomposition	$O(DIR)$
Tucker decomposition	$O(DIR + R^D)$
Tensor-train decomposition	$O(DIR^2)$
Hierarchical Tucker	$O(DIR + DR^3)$