Diffusion Models Critical Analysis report

Kiana Hadysadegh

***Abstract*— Discussion and Critical Analysis of Diffusion Models.**

I. Introduction

Diffusion models have been developed to generate high-quality data closely resembling real-world data. These models have a unique training process that enables them to capture complex underlying structures of datasets and produce highly realistic outputs. In the next section, the diffusion model will be introduced. We review the latent diffusion models in section 3 and discuss further improvements in section 4. In the last section limitations and future research areas will be discussed.

II. Diffusion Model

The general idea of the diffusion model is to add a great amount of noise to an image (forward process) and then learn a neural network to remove that noise (reverse process). Using this trained network, we can give a completely random noise to the network and let the network remove the noise until we have a new image that could occur in our training data. The architecture of the model is like U-Net. It has a bottleneck in the middle. It takes an input image then Downsample-blocks and Resnet-blocks project it to a small resolution. After the bottleneck, it projects it back to its original size using upsample blocks instead. Also at certain resolutions, they put attention blocks and employed skip connections between layers of the same spatial resolutions. The model is always designed for each timestep.

A. Method

Forward process: we add noise to an input image iteratively and destroy information until it becomes pure noise (known normal distribution). We sample noise from the normal distribution. We don't add the same amount of noise at each time step of the process, this is regulated by schedule (beta) which scales the mean and the variance and ensures the variance does not explode as we add more noise. All schedules are ranging between 0 and 1. We apply a linear schedule but the last couple of images become complete noise and redundant since the information is destroyed too fast. So future papers change it with a cosine schedule. The cosine schedule destroys information more slowly and solves the problems of the linear schedule.

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

Using reparametrization and defining the alpha variable (1-beta) for each timestep we can ease the forward process and get each x_t image from the original image x_0:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) \quad (2)$$

Reverse process: we train a neural network that predicts the noise of an input image so it can be subtracted from the input and have a less-noise image. The network learns to remove noise from an image step by step. So we can give a model an image consisting of noise sampled from the normal distribution and let the model gradually remove the noise until we have a clear image. We just predict the mean of the normal distribution for noise, not the variance, since they assume that it is fixed.

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \quad (3)$$

B. Loss function

Using the variational lower bound technique (to remove dependency) and mathematical simplification we derive the loss function. Our loss function is the mean squared error between the actual noise and the predicted noise by the network at each timestep for the given image. We optimize it using the gradient descent method:

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right] \quad (4)$$

C. Problems

While improving sampling speed and enhancing sample quality are always consistent trends across various developments in diffusion models, one of the problems with these models is the slow reverse process. As the quality of the image increases, the dimensionality of the image matrix also increases, resulting in the need for more computations. To address this challenge, Latent diffusion models were introduced which we will introduce in section three. Other problems and improvements will be discussed in section four.

III. Latent diffusion models

The reverse process is slow especially if our is big. As a result, the image matrix will be large and computationally expensive. The idea of the latent diffusion models is to compress the image using variational autoencoders. The latent diffusion model doesn't learn the distribution p(x) of our dataset of images, but rather, the distribution of a latent representation of our data using a variational autoencoder. We compress our data and use it to learn the denoising process and then decompress it to have the original data. Autoencoder is a network that transforms the input image to a vector (code) via an encoder whose dimension is much smaller than the original image. If we pass this code to its decoder it transforms it to the original image by reconstructing it. The problem with autoencoders is the code learned for different images does not make any sense from the semantic view. To overcome this we use variational autoencoders which learn to compress the data while this data is distributed according to a multivariate distribution (e.g. Gaussian). The model learns the mean and the sigma of this distribution.

A. Conditioning mechanism

We can introduce a conditioning signal during the denoising chain in which we influence the model into how to remove the noise so that the output will move toward our condition. We preprocess a condition from different modalities via a domain-specific encoder to create an intermediate representation of it. This is mapped to the intermediate layers of the model using a cross-attention layer.

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V \quad (5)$$

So during the training process, the image x is encoded using an encoder to a latent code z. In a diffusion process, it turns into a complete noise latent code z_T (which is a known normal distribution). Meanwhile, the condition is also encoded to an intermediate representation. This representation and latent code z_T will given to the U-Net as inputs and the output is the clear image latent code which is turned into the image via the decoder.

$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2\right] \quad (6)$$

B. Problems

Although the latent diffusion models increase the speed of the reverse process significantly, There are still problems like sampling speed, training efficiency, likelihood optimization, and bridging more complex distributions which will be discussed in the next section.

IV. Other Improvements

This section highlights four recent developments for enhancing diffusion models:

A. Sampling Acceleration techniques

These techniques focus on enhancing sampling speed and training efficiency. The knowledge Distillation method is an example of this category. It aims to transfer knowledge from larger, more complex models to smaller, simpler models and align and minimize the discrepancy between original and generated samples. This reduces the number of diffusion steps or network sizes required for sampling. Using this method slow sampling speed due to the need for multiple diffusion steps or large network problems is addressed. Denoising Student [7], DSNO [8], Stochastic interpolant [9], and DDIB [10] are algorithms based on this technique. The Denoising Student method trains a smaller, simpler model (the "student" model) to mimic the behavior of a larger, more complex model (the "teacher" model). This involves learning to denoise noisy samples generated by the teacher model. Knowledge distillation involves training a smaller model that can capture essential characteristics of the larger model. The limitation of these techniques is loss of information which leads to quality reduction of the generated samples. Computational overhead is also another problem during the distillation process.

B. New Forward Processes

These techniques address challenges such as suboptimal generative modeling in pixel space, computational complexity, and limitations in handling data from non-Euclidean spaces. Latent diffusion models are categorized in this type of improvement. Diffusion Models on Non-Euclidean Space are another example in this group. They address the problem that the standard diffusion models may not be suitable for data defined within non-Euclidean spaces (such as discrete spaces, manifolds, or graphs). These approaches extend diffusion sampling to non-Euclidean spaces, including discrete spaces (D3PM [11]), manifold data (RDM [12], RGSM [13]), and graph data (EDP-GNN[14]). They modified diffusion theories according to the characteristics of data structures within these spaces. This allows effective generative modeling. Computational complexity is the limitation of the methods.

C. Likelihood Optimization techniques

These techniques aim to address the challenge of likelihood optimization in diffusion models. Diffusion models typically optimize the Evidence Lower Bound rather than the direct optimization of likelihood which is challenging for continuous-time models. MLE training can address this problem by establishing a connection between maximum likelihood training and the weighted denoising score-matching (DSM) objective. ScoreFlow [15], and VDM [16] demonstrate that DSM objectives can provide an upper bound on the negative log-likelihood under specific weighting schemes. This enables a neural network parameter-independent approximation of score-based MLE. The limited degree of independence could affect the scalability and flexibility of these methods which is still a problem.

D. Bridging Distributions

While diffusion models excel at transforming simple Gaussian distributions, they have problems with bridging more complex distributions in tasks such as image-to-image translation and cell distribution transportation. Bridging distribution techniques can tackle this issue. The α-blending [20] method addresses this problem using iterative blending and deblending distributions to create a deterministic bridge. This process involves gradually (via α as a control parameter) transitioning from one distribution to another while ensuring smoothness and fidelity in the transformation. This method may prioritize simplicity and efficiency while still producing high-quality results. Other approaches like Rectified Flow [21] aim to learn an invertible mapping between a simple initial distribution (e.g., Gaussian) and a more complex target distribution. It ensures a smooth and efficient transformation from the initial distribution to the target distribution by incorporating additional steps and techniques. The limitation of this method is computational complexity for high-dimensional data.

V. Limitations and Future Trends

The most significant disadvantage of diffusion models is the need to perform multiple steps at inference time to generate only one sample. Despite the important amount of research, GANs are still faster at producing images. Other issues of diffusion models are the commonly used strategy to employ CLIP embeddings for text-to-image generation. The model struggles to generate readable text in an image since CLIP embeddings do not contain information about spelling which is an interesting problem for me to work on since it is difficult. Future work can study diffusion models applied in other computer vision tasks, such as image dehazing, video anomaly detection, or visual question answering. There are also some works studying anomaly detection in medical images [17], [18], [19], which can be used in other domains like video surveillance or industrial inspection.

References

[1] Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in neural information processing systems* 33 (2020): 6840-6851.

[2] Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684-10695. 2022.

[3] Cao, Hanqun, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, and Stan Z. Li. "A survey on generative diffusion models." *IEEE Transactions on Knowledge and Data Engineering* (2024).

[4] Chang, Ziyi, George A. Koulieris, and Hubert PH Shum. "On the design fundamentals of diffusion models: A survey." *arXiv preprint arXiv:2306.04542* (2023).

[5] Croitoru, Florinel-Alin, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. "Diffusion models in vision: A survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[6] Yang, Ling, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. "Diffusion models: A comprehensive survey of methods and applications." *ACM Computing Surveys* 56, no. 4 (2023): 1-39.

[7] E. Luhman and T. Luhman, "Knowledge distillation in iterative generative models for improved sampling speed," arXiv, 2021. 3.1.1, 1, 6, 7.

[8] H. Zheng, W. Nie, A. Vahdat, K. Azizzadenesheli, and A. Anandkumar, "Fast sampling of diffusion models via operator learning," arXiv:2211.13449, 2022. 3.1.1, 1.

[9] M. S. Albergo and E. Vanden-Eijnden, "Building normalizing flows with stochastic interpolants," ArXiv, vol. abs/2209.15571, 2022. (document), 1, 3.4.

[10] X. Su, J. Song, C. Meng, and S. Ermon, "Dual diffusion implicit bridges for image-to-image translation," arXiv:2203.08382, 2022. 1, 3.4.

[11] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. Van Den Berg, "Structured denoising diffusion models in discrete state-spaces," NeurIPS, vol. 34, pp. 17 98117 993, 2021. (document), 1, 3.2.3, 2, 4.5.1, 5, 7, 8.

[12] C.-W. Huang, M. Aghajohari, A. J. Bose, P. Panangaden, and A. Courville, "Riemannian diffusion models," arXiv:2208.07949, 2022. 1, 3.2.3, 7.

[13] V. De Bortoli, E. Mathieu, M. Hutchinson, J. Thornton, Y. W. Teh, and A. Doucet, "Riemannian score-based generative modeling," arXiv:2202.02763, 2022. 1, 3.2.3, 7.

[14] C. Niu, Y. Song, J. Song, S. Zhao, A. Grover, and S. Ermon, "Permutation invariant graph generation via score-based generative modeling," in AISTATS. PMLR, 2020, pp. 4474–4484. (document), 1, 3.2.3, 2, 4.9, 7.

[15] Y. Song, C. Durkan, I. Murray, and S. Ermon, "Maximum likelihood training of score-based diffusion models," NeurIPS, vol. 34, pp. 1415–1428, 2021. 1, 3.3.1, 7.

[16] D. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," NeurIPS, vol. 34, pp. 21 69621 707, 2021. 1, 3.1.2, 3.3.1, 5, 7.

[17] W. H. Pinaya et al., "Fast unsupervised brain anomaly detection and segmentation with diffusion models," 2022, arXiv:2206.03461.

[18] J. Wolleb, F. Bieder, R. Sandkühler, and P. C. Cattin, "Diffusion models for medical anomaly detection," 2022, arXiv:2203.04306.

[19] J. Wyatt, A. Leach, S. M. Schmon, and C. G. Willcocks, "AnoDDPM: Anomaly detection with denoising diffusion probabilistic models using simplex noise," in Proc. Conf. Comput. Vis. Pattern Recognit. Workshop, 2022, pp. 650–656.

[20] E. Heitz, L. Belcour, and T. Chambon, "Iterative α -(de)blending: a minimalist deterministic diffusion model," ArXiv, vol. abs/2305.03486, 2023. 1, 3.4.

[21] X. Liu, C. Gong, and Q. Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," ArXiv, vol. abs/2209.03003, 2022. (document), 1, 3.4.