On the Collapse Errors Induced by the Deterministic Sampler for Diffusion Models

Yi Zhang
The HKU Musketeers Foundation Institute of Data Science
The University of Hong Kong
yizhang101@connect.hku.hk

Difan Zou
Department of Computer Science
The University of Hong Kong
dzou@cs.hku.hk

Abstract

In this paper, we identify and investigate a critical issue in ODE-based sampling for diffusion models, referred to as the "collapse error," where samples tend to collapse locally. This phenomenon is observed across various datasets, including the MNIST, Mixture of Gaussians (MoG), and other synthetic datasets. Our analysis shows that this error occurs even in the early sampling process, with the error progressively accumulating and amplifying throughout the entire ODE sampling. To better understand the collapse error, we explore several factors that influence the collapse errors, including data distribution, model size, and training settings. Furthermore, We apply a set of techniques to mitigate the collapse error: (1) using an SDE-based sampler, (2) training different models for different segments of the diffusion process, and (3) employing new parameterizations for models. We hope this paper will draw attention to the "collapse error" phenomenon and encourage further research to better understand and address this issue in diffusion models.

1 Introduction

Diffusion models Ho et al. (2020); Song et al. (2020) are a type of generative model designed to produce data by reversing a predefined diffusion process. In the diffusion process, data is gradually transformed into noise by following a stochastic differential equation (SDE). To generate new data, the model reverses this process, using a reverse SDE or an equivalent ordinary differential equation (ODE). These models have achieved remarkable success in tasks such as super-resolution Li et al. (2022); Yue et al. (2024), text-to-image generation Rombach et al. (2022); Saharia et al. (2022); Nichol et al. (2021); Ramesh et al. (2022), and video generation Ho et al. (2022); Wu et al. (2023).

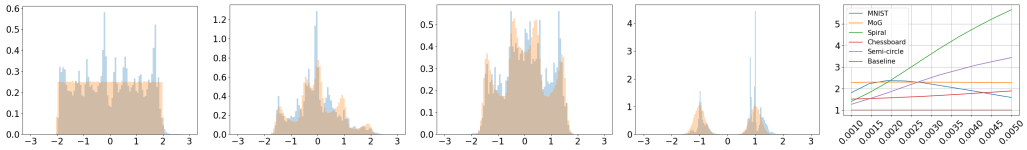
Despite substantial efforts to improve diffusion models in training Karras et al. (2022); Esser et al. (2024), sampling Karras et al. (2022); Watson et al. (2021), and noise schedule Lipman et al. (2022); Liu et al. (2022); Nichol and Dhariwal (2021); Song et al. (2020), in this paper, we identify a new issue with ODE-based sampling methods in these models. Specifically, we observe that the generated samples tend to cluster in low-probability regions. We find that this collapse error occurs even in the early ODE sampling, and progressively increases as sampling proceeds.

In the following section, we examine the nature of this collapse error, presenting our findings and analyzing the conditions under which it arises. By doing so, we aim to provide new insights into improving the robustness and performance of diffusion models.

2 Collapse Errors in ODE-based Sampling for Diffusion Models

In this section, we will introduce the “collapse errors” in ODE-based deterministic sampling for diffusion models, where the generated data distribution collapse locally to certain areas in the entire domain.

Workshop on Scientific Methods for Understanding Deep Learning, NeurIPS 2024.



(a) Chessboard (b) Spiral (c) Semi-circle (d) MoG (e) ANSD ratios

Figure 1: 1D projection of ODE-generated samples (blue) against the ground truth distribution (orange) across four synthetic datasets: (a) Chessboard, (b) Spiral, (c) Semi-circle, and (d) Mixture of Gaussians (MoG). The histograms in (a-d) show sharp peaks in the ODE-generated samples that do not match the ground truth, indicating a local collapse. The fifth panel (e) presents ANSD ratio across different datasets, showing a consistent higher ANSD than truth data distribution in all cases.

Background. In particular, we consider the standard variance-preserving (VP) diffusion model, where the forward process can be defined by the following stochastic differential equation (SDE):

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{w}, \tag{1}$$

where the time variable t progresses from 0 to 1, $\mathbf{x}_t$ represents the data vector at time t, and $\mathrm{d}\mathbf{w}$ denotes the standard Wiener process. The time-dependent noise variance function $\beta(t)$ controls the amount of noise added to the data over time, and it is defined as $\beta(t) = \bar{\beta}_{\text{min}} + t\left(\bar{\beta}_{\text{max}} - \bar{\beta}_{\text{min}}\right)$, where $\bar{\beta}_{\text{min}} = 0.1$ and $\bar{\beta}_{\text{max}} = 20$ Ho et al. (2020); Song et al. (2020).

Then, the generation process of diffusion model can be regarded as solving the inverse process of (1). A widely applied method in practice is the ODE-based deterministic sampler Song et al. (2020), which is designed to solve the following reverse ODE:

$$\mathrm{d}\mathbf{x}_t = \left[\frac{1}{2}\beta(1-t)\mathbf{x}_t + \frac{1}{2}\beta(1-t)\nabla_{\mathbf{x}_t}\log p_{1-t}(\mathbf{x}_t)\right]\mathrm{d}t. \tag{2}$$

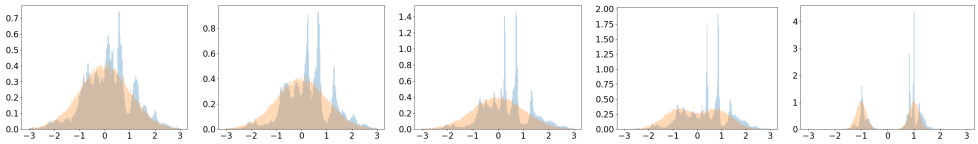
ODE Collapse Errors across Various Datasets. Figures 1a, 1b, 1c, and 1d visualize the discrepancies between the 1D-projection of ODE-generated data and the ground truth in four synthetic datasets, including 2D chessboard-shaped, spiral-shaped, semi-circle-shaped distributions, and a 1D Mixture of Gaussian (MoG). In all these datasets, we observe that ODE-generated samples exhibit sharp peaks, indicating that the model is concentrating the samples into fewer regions than expected. These experiments suggest a common issue across different datasets where ODE sampling tends to create localized concentrations or "collapses" of samples.

To quantify the collapse error, the intuition is to discover and quantify the set of data that are concentrated with each other abnormally. To achieve this, we will examine the neighborhood region of each data point, and count the number of other data points inside the neighborhood region. Mathematically, we propose a quantitative criteria to characterize the severity of the collapse of the data distribution, which is defined as the Average Number of Samples within a certain Distance (ANSD) from each other, i.e.,

$$\text{ANSD}(\mathbf{X}, \epsilon) = \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{j=1, j\neq i}^{N}\mathbb{I}(\|\mathbf{x}_i - \mathbf{x}_j\| \leq \epsilon)\right),$$

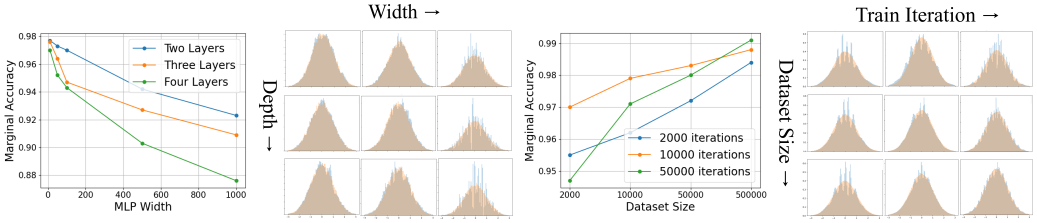
where $\mathbf{X}$ represents the sample set, $\|\mathbf{x}_i - \mathbf{x}_j\|$ is the Euclidean distance between samples $\mathbf{x}_i$ and $\mathbf{x}_j$, ϵ is the size of the neighborhood set, and $\mathbb{I}(\cdot)$ is an indicator function that equals 1 if the condition inside is true (i.e., the distance between two samples is less than or equal to ϵ), and 0 otherwise. A higher ANSD value for the generated data compared to the dataset indicates a more severe ODE collapse error. As shown in Figure 1(e), the ANSD ratio between the generated samples and the groundtruth examples consistently exceeds 1 across all datasets. This outcome demonstrates that the model tends to concentrate samples more tightly around certain regions of the data distribution.

Collapse Errors Accumulates during ODE Sampling. We find that the collapse error is accumulated throughout the ODE sampling process, even from the early stages, and becomes more severe as sampling progresses. Figure 2 provides a series of density plots showing the evolution of this error at different stages of the reverse diffusion process. From the plots, we observe that the generated



(a) $t = 0.2$ (b) $t = 0.4$ (c) $t = 0.6$ (d) $t = 0.8$ (e) $t = 1.0$

Figure 2: These plots correspond to the intermediate distributions generated by ODE-based sampling at various time points, $t = 0.2, 0.4, 0.6, 0.8, 1.0$. The blue lines represent the generated distributions, while the orange areas depict the ground truth distributions.



(a) (b) (c) (d)

Figure 3: (a) Marginal accuracy vs. model width for MLPs with varying depths. (b) Distribution of generated samples at $t = 0.4$ for varying model width and model depth. (c) Marginal accuracy vs. dataset size at different numbers of training iterations. (d) Distribution of generated samples at $t = 0.4$ for varying training iterations and dataset sizes.

samples begin to deviate from the ground truth distribution as early as $t = 0.2$, indicating that errors are present at the start of the ODE sampling process. As t increases, these deviations become more severe, with the generated samples forming sharper and more concentrated peaks compared to the smoother ground truth. This trend continues, and by $t = 1.0$, the generated distribution shows significant collapse, clustering locally rather than evenly covering the target distribution. These observations demonstrate that the collapse error is not only a consequence of the final stage of sampling but is instead a compounding issue that intensifies throughout the entire ODE sampling trajectory.

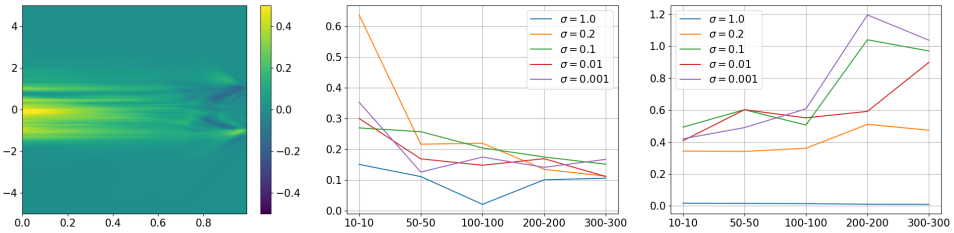
Factors Influencing Collapse Errors. To better understand the collapse errors, we investigate their relationship with various factors, such as model size (both width and depth) and training settings (including the number of iterations and dataset size). Given that collapse errors are observed in the early stages of the sampling process, we evaluate the marginal accuracy of intermediate distributions generated by ODE-based sampling at $t = 0.4$. The experimental results are summarized in Figure 3.

As illustrated in Figures 3a and 3b, an increase in model width or depth generally leads to a reduction in marginal accuracy, with more pronounced effects for wider and deeper networks. This suggests that larger models may be more prone to collapse errors due to their capacity to overfit or represent overly complex distributions. Figures 3c and 3d show that using larger datasets and conducting more training iterations typically enhance marginal accuracy, thus mitigating collapse errors. However, when the dataset size is small, increasing the number of training iterations does not necessarily reduce collapse errors.

3 Analysis of Collapse Errors

In this section, we will a perform deeper investigation regarding the collapse errors observed in practice and highlight several methods to migate the errors.

Velocity Error Propagates along t. To analyze the collapse error observed in ODE-based sampling, a possible direction is to analyze the velocity field error, i.e., the approximation error between the learned vector filed and the groundtruth one (see (2)), during the reverse diffusion process. Specifically, considering the MoG data, Figure 4a visualizes how the velocity error, significant even at early sampling stages, propagates and accumulates along t throughout the reverse diffusion process. It can be observed that the velocity errors for different t and $\mathbf{x}$ are not random, it exhibits certain



(a) Velocity error (b) MAE of score at $t = 0.8$. (c) MAE of score at $t = 0$.

Figure 4: (a) Visualization of the velocity field error during reverse ODE sampling. (b-c) Mean Absolute Error (MAE) of the score function at $t = 0.8$ and $t = 0$, respectively, across different MLP configurations for different values of the Gaussian standard deviation (σ) in the MoG dataset.

well-organized shapes in both data (i.e., for different $\mathbf{x}_t$) and temporal domains (i.e., for different t). Consequently, the velocity errors at different $\mathbf{x}_t$ and t during the reverse ODE will not be cancelled but will accumulated problematically. A minor deviations at the beginning of the sampling process gradually compound into more significant errors over time, ultimately driving the generated samples to cluster into fewer modes at later stages.

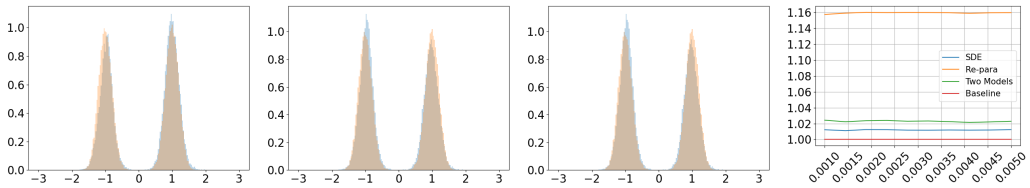
In response to this issue, we apply SDE-based sampling as a practical intervention. By introducing stochasticity into the sampling process, the inherent randomness helps mitigate the cumulative propagation of velocity error. The SDE formulation used in our approach is further detailed in Appendix 5. As evidenced by Figure 5a, SDE-based sampling significantly reduces the sharp peaks that characterize the ODE results, providing a smoother distribution of generated samples. Moreover, Figure 5d showcases how the ANSD ratio remains much closer to 1, indicating that the generated samples align more closely with the ground truth distribution, and the SDE-based sampling is effective in reducing the collapse error.

Challenges in Fitting Score Functions of Different Sampling Stages. Note that in practice (also in our paper), only one model is used to learn the velocity field at different t's, while the time t appears as an embedding that is concatenated with the data feature. Then, we find out that the collapse errors are also highly affected by the model's difficulty in simultaneously learning score functions of for varying t, i.e., small $t \sim 0.1$ and large $t \sim 0.9$. These two different time stamps lead to the target vector field functions with fundamentally different complexities. As shown in Figures 4b, larger MLPs fit the score function at $t = 0.8$ more effectively. This is because the score function at $t = 0.8$ becomes sharper with smaller σ, requiring more model capacity to capture its complexity. However, as the model width increases and improves its ability to fit the more complex score at $t = 0.8$, its performance at $t = 0$ deteriorates. This trade-off highlights a key trend: larger models may fit complex score functions better at the cost of overfitting simpler ones. Additionally, as σ decreases, the model's ability to fit the score function at $t = 0$ degrades more significantly with increasing model width.

To address this issue, we adopt two practical strategies. The first approach employs two separate models to handle the score functions at different stages of the sampling process, as depicted in Figure 5b. Empirically, we set one model to learn the score function for $t < 0.6$, while the other model is responsible for learning the score function for the remaining stages. This division allows each model to specialize in a narrower range of time steps, mitigating the complexity trade-offs seen when a single model is tasked with learning the entire score function. The second approach reparameterizes the model to provide a more accurate initial approximation of the score function at the beginning of the sampling process, as illustrated in Figure 5c. Further details on both techniques are provided in Appendix D. These strategies effectively mitigate the collapse error, as evidenced by the smoother sample distributions and ANSD values approaching 1, suggesting improved sample diversity and closer alignment with the true data distribution.

4 Conclusion

In this paper, we shield light on a "collapse error" in ODE-based sampling methods for diffusion models, a phenomenon that samples collapse locally. Our experiments on the real-world MNIST dataset,



(a) SDE (b) Two models (c) Reparameterization (d) ANSD ratios

Figure 5: (a-c) Sample distribution using techniques SDE-based sampling, seprate models for training, and reparameterization, respectively. (d) ANSD ratio using the three techniques.

Mixture of Gaussians (MoG), and other synthetic datasets reveal that the collapse error appears even in the early sampling stages and progressively accumulates. We explored the impact of model size and training settings on the collapse errors. To mitigate the collapse errors, we experimented with three simple yet effective approaches: (1) SDE-based sampling, (2) using separate models for different stage of sampling, and (3) reparametrization. We hope this paper will draw attention to the "collapse error" phenomenon and its underlying causes, encouraging further exploration and discussion in the community. Our work aims to provide a foundation for future research in refining sampling methods and improving the robustness of diffusion models.

References

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. (2024). Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. (2022). Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646.

Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577.

Li, H., Yang, Y., Chang, M., Chen, S., Feng, H., Xu, Z., Li, Q., and Chen, Y. (2022). Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. (2022). Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

Liu, X., Gong, C., and Liu, Q. (2022). Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. (2021). Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

Nichol, A. Q. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. (2022). Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2020). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Watson, D., Ho, J., Norouzi, M., and Chan, W. (2021). Learning to efficiently sample from diffusion probabilistic models. *arXiv preprint arXiv:2106.03802*.

Wu, J. Z., Ge, Y., Wang, X., Lei, S. W., Gu, Y., Shi, Y., Hsu, W., Shan, Y., Qie, X., and Shou, M. Z. (2023). Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633.

Yue, Z., Wang, J., and Loy, C. C. (2024). Resshift: Efficient diffusion model for image super-resolution by residual shifting. *Advances in Neural Information Processing Systems*, 36.

A The Background of Diffusion Models

In this paper, we follow the typical score-based diffusion model with the variance-preserving (VP) schedule.

The forward diffusion process adds Gaussian noise to the data gradually over time. In this paper, we follow the most prominent variance-preserving schedule, the forward diffusion process is defined by the following stochastic differential equation (SDE):

$$\mathrm{d}\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}w, \tag{3}$$

where t progresses from 0 to 1, $\mathbf{x}$ represents the data vector at time t, and $\mathrm{d}w$ denotes the standard Wiener process. The time-dependent noise variance function $\beta(t)$ controls the amount of noise added to the data over time, and it is defined as:

$$\beta(t) = \bar{\beta}_{\text{min}} + t\left(\bar{\beta}_{\text{max}} - \bar{\beta}_{\text{min}}\right), \tag{4}$$

where $\bar{\beta}_{\text{min}} = 0.1$ and $\bar{\beta}_{\text{max}} = 20$.

The backward diffusion process reverses the forward process to gradually transform the noise back into data. For the variance-preserving schedule, the reverse-time SDE is given by:

$$\mathrm{d}\mathbf{x} = \left[\frac{1}{2}\beta(1-t)\mathbf{x} + \beta(1-t)\nabla_{\mathbf{x}}\log p_{1-t}(\mathbf{x})\right]\mathrm{d}t + \sqrt{\beta(1-t)}\,\mathrm{d}w, \tag{5}$$

where t progresses from 0 to 1, $\nabla_{\mathbf{x}}\log p_t(\mathbf{x})$ is the score function, representing the gradient of the log probability density at time t.

The equivalent ODE removes the noise term, allowing for deterministic sampling:

$$\mathrm{d}\mathbf{x} = \left[\frac{1}{2}\beta(1-t)\mathbf{x} + \frac{1}{2}\beta(1-t)\nabla_{\mathbf{x}}\log p_{1-t}(\mathbf{x})\right]\mathrm{d}t. \tag{6}$$

This ODE represents a deterministic path that can be followed to sample data points from the learned distribution.

To estimate $\nabla_{\mathbf{x}}\log p_t(\mathbf{x})$, we follow Song et al. (2020) to train a time-dependent score model $\mathbf{s}_\theta(\mathbf{x}, t)$:

$$\theta^* = \arg\min_\theta \mathbb{E}_t\left\{\lambda(t)\mathbb{E}_{\mathbf{x}(0)}\mathbb{E}_{\mathbf{x}(t)|\mathbf{x}(0)}\left[\left\|\mathbf{s}_\theta(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)}\log p(\mathbf{x}(t)|\mathbf{x}(0))\right\|_2^2\right]\right\}, \tag{7}$$

where $\lambda : [0, T] \to \mathbb{R}_{>0}$ is a positive weighting function. To balance the terms in the objective, we typically choose $\lambda = 1/\mathbb{E}\left[\left\|\nabla_{\mathbf{x}(t)}\log p(\mathbf{x}(t)|\mathbf{x}(0))\right\|_2^2\right]$. The time variable t is uniformly sampled over the interval $[0, T]$, $\mathbf{x}(0) \sim p_0(\mathbf{x})$ represents the data distribution, and $\mathbf{x}(t)$ denotes the marginal distribution of $\mathbf{x}$ at time t, resulting from running the forward diffusion process up to time t.

To evaluate how well the diffusion model recovers the target distribution, we introduce the criteria of marginal accuracy, which can be expressed as:

$$\text{Marginal Accuracy}(\hat{p}, p) = 1 - 0.5 \times \frac{1}{d}\sum_{i=1}^{d} TV(\hat{p}_i, p_i),$$

where $\hat{p}_i(x)$ is the marginal distribution of the i-th dimension of the sampled data, $p_i(x)$ is the true marginal distribution of the i-th dimension, d is the total number of dimensions, and TV is the total variation distance.

B Score Functions at Different Sampling Stages

The score function of $p_1(x)$, which corresponds to a standard Gaussian distribution, is depicted in Figure 6(a). For a Gaussian distribution, the score function, defined as the gradient of the log probability density with respect to x, is a simple linear function:

$$\nabla_{\mathbf{x}} \log p_1(\mathbf{x}) = -\mathbf{x}.$$

Regardless of how the data distribution $p_0(x)$ changes, the score function for $p_1(x)$ remains a consistent, simple negative mapping, reflecting its purely Gaussian nature.

In contrast, the score function of $p_0(x)$, which corresponds to the data distribution, is shown in Figure 6(b) for a 1D mixture of Gaussians (MoG) setting with means at -1 and 1. Unlike the Gaussian score function, the score function of $p_0(x)$ is highly dependent on the specific characteristics of the data distribution and varies significantly depending on the standard deviation (σ) of the components in the MoG. As σ decreases, the score function becomes increasingly sharp and complex, reflecting the multimodal nature of the data distribution. The score functions for different values of σ (1, 0.2, 0.1, 0.01, 0.001) exhibit intricate structures that capture the density peaks and valleys, contrasting sharply with the linear and simple form of the Gaussian score.

Thus, while the score function for the Gaussian distribution $p_1(x)$ is a smooth, linear function that remains unaffected by the data distribution, the score function for the data distribution $p_0(x)$ is highly nonlinear and varies dramatically with different standard deviations, illustrating the challenges involved in learning such complex distributions during the reverse diffusion process.

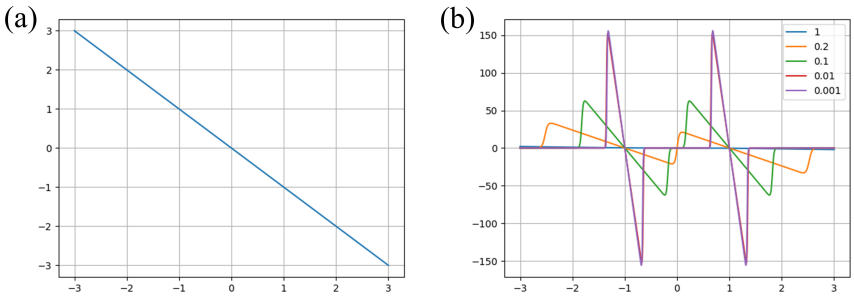


Figure 6: (a) The score function of the standard Gaussian distribution, showing a simple negative linear mapping. (b) The score function of the data distribution in a 1D mixture of Gaussians (MoG) setting with means at -1 and 1 and different standard deviations ($\sigma = 1, 0.2, 0.1, 0.01, 0.001$).

The score function during the early sampling stage should consistently remain a negation function, unaffected by the complexity of the score function in the late sampling stage. While it might be expected that this negation function would be straightforward for the model to learn, we empirically find that the complexity of the score function in the late sampling stage can significantly impact the model's ability to learn the simpler score function in the early sampling stage. Figure 7 demonstrates this effect by showing the diffusion model's intermediate ODE sampling results at $t = 0.4$ under different settings of standard deviation σ in the 1-dimensional mixture of Gaussians (MoG) with fixed means at -1 and 1. The plots indicate that the samples start to deviate from the standard Gaussian distribution even at this early stage. Moreover, the smaller the σ, the greater the deviation from the Gaussian in the early sampling stages.

C The Gaussian Preserving Property in the Variance Preserving Schedule

Figure 8 demonstrates the Gaussian-preserving property of a typical variance-preserving (VP) schedule during the reverse diffusion process in a 1D MoG settings with 2 Guassians centered at -1 and 1 and have a standard deviation of 0.2.

In Figure 8(a), the heatmap shows the probability density evolution over time, highlighting how the distribution remains nearly Gaussian throughout the initial stages of the reverse diffusion process.

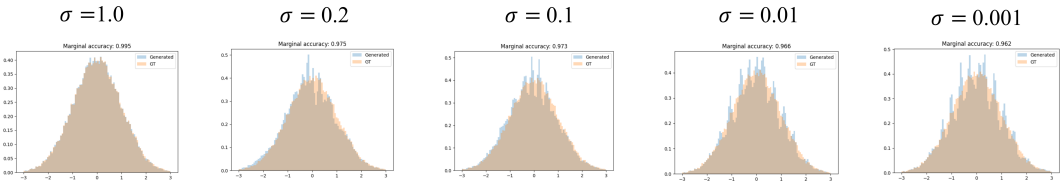


Figure 7: Histograms of the intermediate samples generated by ODE-based sampling at t=0.4 for a 1D mixture of Gaussians (MoG) with means at -1 and 1, and varying standard deviations (σ=1.0,0.2,0.1,0.01,0.001).

Even as time progresses from t=0 to t=0.6, the probability density remains concentrated and Gaussian-like, indicating minimal deviation from the initial Gaussian state.

In Figure 8(b), the sampling trajectories starting from 50 evenly spaced points between -2 and 2 further illustrate this property. From t=0 (Gaussian) to around t=0.6, the trajectories remain tightly aligned with their origins. This suggests that the VP schedule effectively maintains the Gaussian characteristics in the early sampling stages.

In Figure 8(c), the density plots at different time steps ($t = 0.20, 0.40, 0.60, 0.80, 1.00$) reinforce this observation. Up to t=0.6, the distributions closely resemble a Gaussian form, confirming that the VP schedule preserves the Gaussian structure until later in the process. It is only after t=0.6 that the distribution begins to diverge towards the more complex target data distribution.

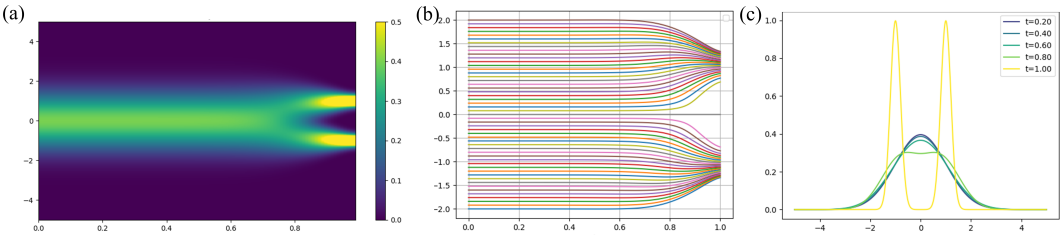


Figure 8: The ideal reverse diffusion process of a 1D mixture of Gaussians (MoG) with means at -1 and 1, and standard deviations (σ=0.2) in a typical variance-preserving (VP) schedule. (a) Heatmap of the probability density evolution over reverse diffusion process. (b) Sampling trajectories starting from 50 evenly spaced points between -2 and 2. (c) Density plots at different stages of reverse diffusion process (t=0.20,0.40,0.60,0.80,1.00).

D Experiment Settings

D.1 Mixture of Gaussians (MoG) Experiment

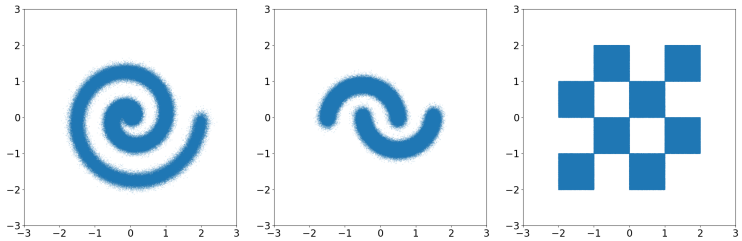
For the Mixture of Gaussians (MoG) experiment, we used a two-layer Multilayer Perceptron (MLP) with 100 neurons for each layers and Tanh activation functions. The time variable t was concatenated to the input x. The model was optimized using Adam with a learning rate of 5×10^{-3}. A total of 50,000 synthetic data points were generated, and the model was trained for 10,000 iterations using gradient descent. We follows a typical variance-preserving noise schedule for training and sampling as we describe in A.

D.2 MNIST Experiment

For the MNIST experiment, the dataset was downsampled to a resolution of 6×6 to reduce computational cost. A total of 60,000 samples were used, trained with stochastic gradient descent (SGD) with a batch size of 60 over 1,200 epochs. The model was a two-layer U-Net Ronneberger et al. (2015), consisting of an encoder and decoder. The encoder included two convolutional layers (kernel size 3, 256 channels, Tanh activation) and a MaxPooling layer (2×2) for downsampling. The decoder mirrored this structure with transpose convolutions for upsampling and used skip connections to combine features from the encoder. A final output convolution layers (kernel size 3, 1 channel) layer

to reconstruct the image. The time variable t was expanded and concatenated to the input x as an additional channel. The optimizer was Adam with a learning rate of 3×10^{-4}. We follows a typical variance-preserving noise schedule for training and sampling as we describe in A.

D.3 Other Synthetic Datasets Experiment



(a) (b) (c)

Figure 9: Synthetic datasets containing (a) spiral-shaped, (b) semicircle-shaped, and (c) chessboard-shaped distributions.

For the other synthetic datasets, we generated 2D clusters in various shapes, including spiral-shaped, semi-circle-shaped, and chessboard-shaped distributions.

The spiral-shaped dataset consists of points along a single spiral curve. The spiral is generated by varying the radius and the angle of each point. The radius increases linearly from 0 to 2 units as the angle progresses from 0 to 4π (representing two full turns). A total of 500,000 points are sampled with Gaussian noise of standard deviation 0.1 added to each coordinate. The semi-circle-shaped dataset comprises two semi-circles with radius of 1, positioned at slightly different vertical offsets. The first semi-circle is centered at $(0.5, 0.1)$, and the second one is centered at $(-0.5, -0.1)$. Points are evenly distributed along these arcs, with Gaussian noise of standard deviation 0.1 added to each coordinate. The chessboard-shaped dataset is generated to form a 4×4 grid pattern, mimicking a chessboard, where data points are concentrated in alternate cells. Each cell is 1×1 unit in size. The points within each cell are uniformly distributed.

We used a three-layer Multilayer Perceptron (MLP) with 100 neurons in each layer and Tanh activation functions. The time variable t was concatenated to the input x. The model was trained in gradient decent using the Adam optimizer with a learning rate of 5×10^{-3}, and the training was conducted over 10,000 iterations.

We follows a typical variance-preserving noise schedule for training and sampling as we describe in A.

D.4 Sampling Procedure

ODE sampling results were generated using the Euler's method with 100 steps, while SDE-based sampling was performed with 1,000 steps.

D.5 Training Separate Models for Different Sampling Stages

We propose training separate models for different segments of the diffusion process. Specifically, two models are identical in structure and independently trained: one for the earlier stages (up to $t = 0.6$) and another for the later stages.

D.6 Reparametrization to for models to mitigate collapse errors

We recall that the training objective for score models is 7. Following the annotions from A, we can further derive a more explicit training objective.

The solution of the SDE in 3 is:

$$\mathbf{x}(t) \sim \mathcal{N}\left(\exp\left(-\frac{1}{4}t^2(\beta_1-\beta_0)-\frac{1}{2}t\beta_0\right)\mathbf{x}(0),\ \sqrt{1-\exp\left(2\left(-\frac{1}{4}t^2(\beta_1-\beta_0)-\frac{1}{2}t\beta_0\right)\right)}\right), \tag{8}$$

where we denote $\sigma_t = \sqrt{1-\exp\left(2\left(-\frac{1}{4}t^2(\beta_1-\beta_0)-\frac{1}{2}t\beta_0\right)\right)}$. Then training objective for score model is:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_t\left\{\lambda(t)\mathbb{E}_{\mathbf{x}(0)}\mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\left\|\mathbf{s}_\theta(\mathbf{x}(t),t)+\frac{\epsilon}{\sigma_t}\right\|_2^2\right]\right\}. \tag{9}$$

Song et al. (2020) proposed an alternative parameterization approach to directly predict the noise and subsequently transform it back to the score. The training objective for ϵ parameterization is:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_t\left\{\lambda(t)\mathbb{E}_{\mathbf{x}(0)}\mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\|\mathbf{s}'_\theta(\mathbf{x}(t),t)-\epsilon\|_2^2\right]\right\}, \tag{10}$$

and after training, the truth score function $s^*_\theta(x,t)$ is parameterized by $-\frac{\mathbf{s}'_{\theta*}(x,t)}{\sigma_t}$. In this parameterization, $\mathbf{s}'_{\theta*}(x,1) \sim x$ for any x. To set it as an edge condition of the score model, we set $s_\theta(x,t) = \alpha_t N_\theta(x,t) + (1-\alpha_t)x$ where $\alpha_1 \sim 0$, $\alpha_0 \sim 1$, and N is a neural network (an MLP in our experiments). Empirically, we find that setting α_t as σ_t gives satisfying performance.

E ODE Collapse Errors in Early Sampling Stages

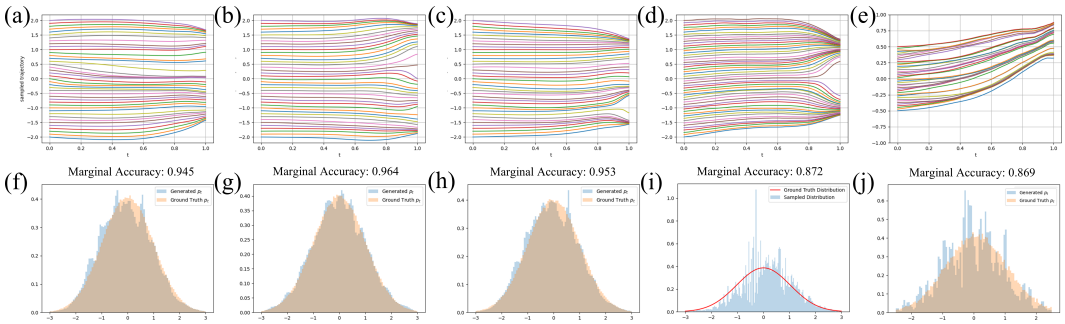


Figure 10: (a-d) Visualization of ODE sampled trajectories for the spiral-shaped, semi-circle-shaped, chessboard-shaped, MoG distribution and MNIST respectively. (e-j) Visualization of intermediate ODE sampling distribution when t progresses from 0 to 0.4 in ODE for the spiral-shaped, semi-circle-shaped, and chessboard-shaped, MoG distribution, and MNIST respectively.

To verify that the ODE collapse occurs in early sampling stages, we also visualize the corresponding ODE sampling trajectory and intermediate sampling distribution when t progresses from 0 to 0.4 accross various datasets. In Figure 10 (a-d), we observe that the trajectories display a concentration of paths. In Figure 10 (e-j), we observed even in early sampling stage, the ODE-sampled distribution is deviated from the expected Gaussian-like distribution. To quantify the difference between two distribution, we introduce Marginal Accuracy, which is shown on the top of Figure 10 (f-j). The detailed definition of marginal accuracy is shown in Appendix A.

F Mitigating Collapse Error in More Datasets

In this section, we apply the three techniques we mentioned in 3 to mitigate the collapse error observed in ODE sampling in various datasets.

In the first row of Figure 11, we show that using SDE-based sampling with more sampling steps can effectively address the collapse errors observed in ODE sampling. By incorporating stochasticity into the diffusion process, SDE sampling maintains a closer approximation to the target distribution, thereby reducing the risk of collapse and producing a smoother sample distribution across various datasets.

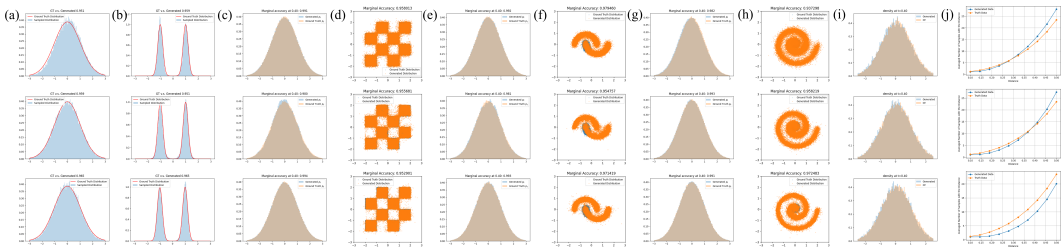


Figure 11: Visualizations of three methods to mitigate the collapse error in ODE-based diffusion models across different datasets. Columns (a, c, e, g, i) show the distributions generated at $t = 0.4$ by the ODE, while columns (b, d, f, h) provide corresponding visualizations of the generated data. The last column (j) presents the plot of Average Number of Samples within a certain Distance threshold on the MNIST dataset. From top to bottom, the rows represent three methods: (1) using Stochastic Differential Equations (SDE)-based reverse diffusion process, (2) employing two separate models with a transition at $t = 0.6$ to learn score functions, and (3) applying re-parameterization.

In Section 2, we demonstrated that the learning of the score function for early sampling is heavily influenced by the complexity of the score function required for late sampling stages. To mitigate this issue, we propose training separate models for different segments of the diffusion process. The second row in Figure 11 illustrates the results of this approach, where two models are independently trained: one for the earlier stages (up to $t = 0.6$) and another for the later stages. This strategy significantly improves the sample quality and distribution consistency, as evidenced by the generated samples more accurately matching the ground truth across different datasets.

In the third row of Figure 11, we train the model using re-parameterization. Given that the score function at $t = 0$ is a strict negation mapping, irrespective of the data distribution, we impose this negation mapping as a boundary condition for the neural network. More detailed desciption on the re-parametrization is shown in Appendix D.6. This approach allows the model to get rid of the trade-off between learning a simple negation mapping for early sampling and a complex score function for late sampling, enabling it to focus its capacity more effectively on the challenging task of learning the complex score function required for the later stages. The results indicate that this method improves the alignment between the generated and true distributions, particularly at intermediate sampling points.

G Case Study on MNIST with More Details

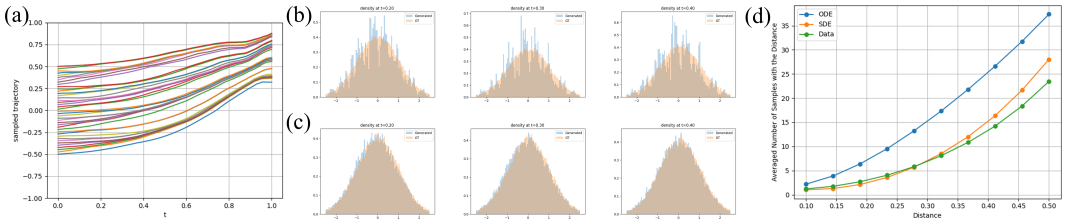


Figure 12: (a) Sampling trajectories of a pixel in the reverse diffusion process using ODE on the MNIST dataset, starting from 40 evenly spaced points between -0.5 and 0.5. (b) Histograms of a pixel in the generated samples using ODE-based sampling at different stages during the reverse diffusion process, corresponding to $t = 0.2, 0.4, 0.6$. (c) Histograms of a pixel in the generated samples using SDE-based sampling at the same stages ($t = 0.2, 0.4, 0.6$). (d) The average number of samples within a certain distance threshold from each other among 20,000 samples.

In our experiment using the MNIST dataset, we observe in the early sampling stage, the distribution of sampled data deviates from the intended Gaussian distribution. As shown in Figure 12(a), the sampling trajectories become increasingly concentrated, particularly in the early sampling stage. This concentration of paths indicates that the model tends to focus on fewer modes, contributing to a collapse in diversity among the generated samples. Figure 12(b) presents the density evolution plots for ODE-based sampling, showing that even early in the sampling process, there are noticeable

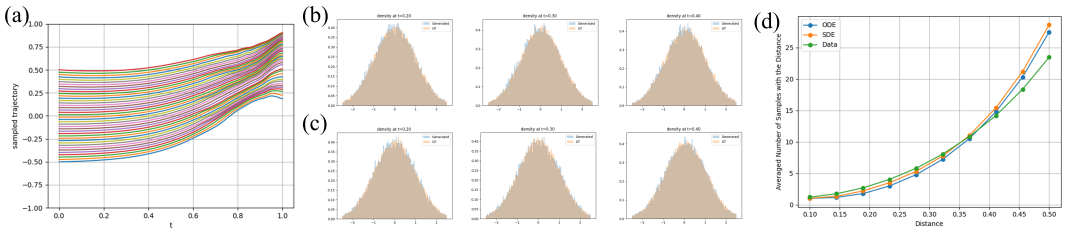


Figure 13: Results of the proposed method applied to the MNIST dataset. (a) Sampling trajectories of a pixel in the reverse diffusion process using ODE, starting from 40 evenly spaced points between -0.5 and 0.5. (b) Histograms of a pixel in the generated samples using ODE-based sampling at different stages of the reverse diffusion process, corresponding to $t = 0.2, 0.4, 0.6$. (c) Histograms of a pixel in the generated samples using SDE-based sampling at the same stages ($t = 0.2, 0.4, 0.6$). (d) The average number of samples within a certain distance threshold from each other among 20,000 data points.

deviations from the standard Gaussian distribution. These deviations accumulate over time, causing the distribution to diverge further from the target as the sampling progresses.

Besides, as shown in Figure 12(d), for ODE-based sampling, the ANSD is consistently higher than the reference data, reinforcing the evidence of a severe collapse phenomenon. This outcome demonstrates reduced diversity among the generated samples, as the model tends to concentrate samples in fewer modes. In contrast, Figure 12(c) shows the results for SDE-based sampling, where the deviations from the reference Gaussian are much smaller, suggesting that, while both methods may start with slight deviations from the Gaussian distribution, ODE-based sampling accumulates errors more rapidly, leading to a greater reduction in sample diversity compared to SDE-based sampling.

Figure 13 shows the results of applying the seperate-model-training strategy to the MNIST dataset, compared to the collapse error shown in Figure 12. By traing separate models for different sampling stages, the sampling trajectories (Figure 13(a)) are more spread out compared to Figure 12(a), indicating improved diversity. The density plots for ODE-based sampling (Figure 13(b)) show a much closer alignment with the standard Gaussian distribution in the early stages of the sampling process, compared to Figure 12(b). Similarly, the results for SDE-based sampling (Figure 13(c)) maintain their consistency with the reference distribution, as seen in Figure 12(c).

Moreover, Figure 13(d) demonstrates that ANSD for ODE is now more closely aligned with the dataset, significantly reducing the collapse effect observed in Figure 12. This comparison highlights that the new method effectively mitigates the collapse phenomenon in ODE-based sampling while maintaining the advantages of SDE-based sampling.

H Experiment on Mixture of Gaussian with More Details

H.1 Collapse error in Mixture of Gaussian (MoG)

In this experiment, we demonstrate the occurrence of collapse errors in ODE-based sampling using a simple mixture of Gaussians (MoG) with means of -1 and 1 and a standard deviation of 0.2. As shown in Figure 14(a), the sampled trajectories under ODE-based sampling tend to converge along specific paths, resulting in a concentration of samples in certain areas. This behavior indicates a reduction in diversity, where the generated samples fail to represent the full data distribution.

Figure 14(b) further illustrates the evolution of the probability distribution during the sampling process. While the variance-preserving (VP) schedule aims to keep the distribution close to a Gaussian in the early stages, the ODE-based sampling method accumulates small errors that cause deviations from the Gaussian distribution almost from the beginning. These errors accumulate over time, leading to a noticeable divergence from the intended distribution.

In contrast, SDE-based sampling effectively maintains the distribution close to the Gaussian throughout the process, avoiding the error accumulation seen with ODE-based methods, as shown in Figure 14(c). This comparison suggests that, the ODE-based sampling approach can introduce biases that

reduce diversity, whereas SDE-based sampling remains more robust in preserving the desired data distribution.

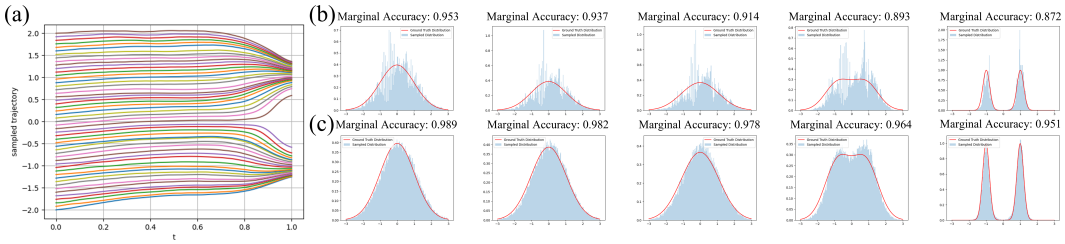


Figure 14: Results for a 1-dimensional mixture of Gaussians (MoG) with means at -1 and 1, and standard deviations σ=0.2 for both components. (a) ODE sampling trajectories of the reverse diffusion process, starting from 50 evenly spaced points between -2 and 2. (b) Density plots of the generated samples using ODE-based sampling at different stages of the reverse diffusion process, corresponding to t=0.2,0.4,0.6,0.8,1.0. Marginal accuracy of the density is shown on the top of the figures. (c) Density plots of the generated samples using SDE-based sampling at the same stages (t=0.2,0.4,0.6,0.8,1.0). Marginal accuracy of the density is shown on the top of the figures.

H.2 Addressing Collapse Errors in Mixture of Gaussian (MoG)

We compare the results from Figure 15, where separate models were used for learning scores functions at different sampling statges, with those from Figure 14, which shows the results without this approach. As discussed earlier, Figure 14 demonstrates a severe concentration in the sampling trajectories, indicating that the model fails to maintain a diverse distribution throughout the reverse diffusion process.

In contrast, Figure 15, using the separate models approach, shows that the sampling trajectories do not exhibit such concentration, suggesting that the model better preserves diversity across the diffusion path. Additionally, the density plots in Figure 15 (b) and (f) show a closer match to both the standard Gaussian and target data distributions, further highlighting the method's effectiveness in reducing collapse error and improving sampling accuracy.

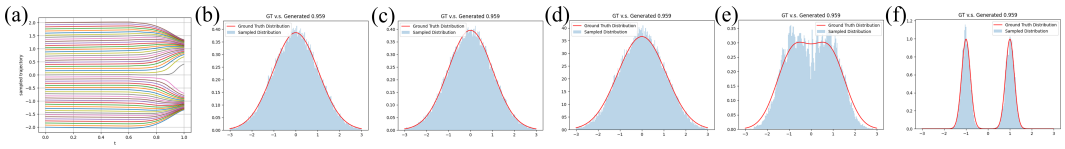


Figure 15: Results of the proposed method on a 1-dimensional mixture of Gaussians (MoG) with means at -1 and 1, and standard deviations σ=0.2 for both components. (a) Sampling trajectories of the reverse diffusion process starting from 51 evenly spaced points between -2 and 2. (b-f) Density plots of the generated samples using the proposed method at different stages of the reverse diffusion process, corresponding to t=0.2,0.4,0.6,0.8,1.0. Marginal accuracy of the density is shown on the top of the figures.

I More Experiments on Synthetic Datasets

I.1 Collapse Errors in Synthetic Datasets with More Details

In these experiments, we show collapse errors in more synthetic datasets, including spiral-shaped, semi-circle-shaped, and chessboard-shaped distributions.

In the ODE sampling trajectories shown in Figures 16(a), 17(a), and 18(a), the sampling paths become overly concentrated as the reverse diffusion progresses. This concentration of trajectories suggests that the model's samples are not spreading adequately across the data space, leading to a reduction in diversity. As the ODE-based sampling progresses, this concentration effect becomes more severe, with samples clustering together in a limited region.

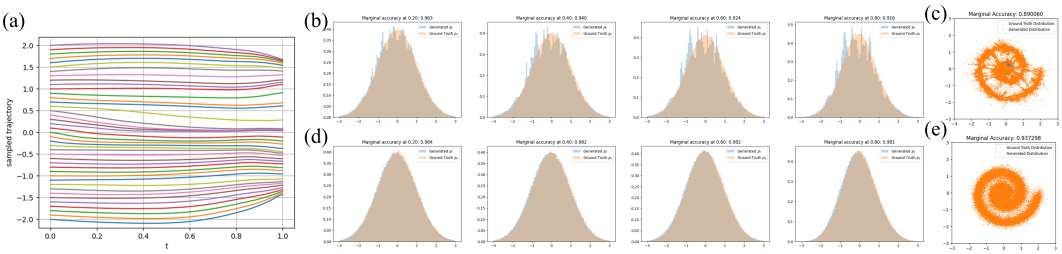


Figure 16: Results for the spiral-shaped dataset: (a) ODE sampling trajectories of the reverse diffusion process, starting from 40 evenly spaced points between -2 and 2. (b) Density plots of the generated samples using ODE-based sampling at different stages of the reverse diffusion process, corresponding to t=0.2,0.4,0.6,0.8. Marginal accuracy of the density is shown on the top of the figures. (c) Visualization of the ODE-generated data and ground truth data. (d) Density plots of the generated samples using SDE-based sampling at the same stages (t=0.2,0.4,0.6,0.8,1.0). Marginal accuracy of the density is shown on the top of the figures. (f) Visualization of the SDE-generated data and ground truth.

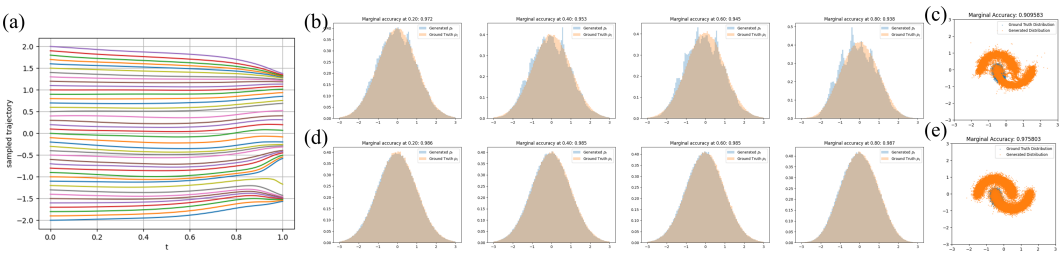


Figure 17: Results for the semi-circle-shaped dataset. The figure descriptions are the same as in Figure 16.

The density plots of the generated samples using ODE-based sampling, depicted in Figures 16(b), 17(b), and 18(b), further highlight this issue. These plots show that, even in the earlier stages of sampling, the generated data begins to deviate from a standard Gaussian distribution, indicating a failure to preserve the Gaussian-like properties expected at those stages. In contrast, Figures 16(d), 17(d), and 18(d) show that the SDE-based samples maintain a distribution closer to the ground truth, with less deviation from the Gaussian form.

Figures 16(c), 17(c), and 18(c) visualize the discrepancies between the ODE-generated data and the ground truth, revealing that ODE-generated samples often incorrectly concentrate in low-probability regions of the target distribution. Besides, ODE-generated samples fail to adequately cover areas of higher density in the ground truth data. The SDE-based samples, as shown in Figures 16(e), 17(e), and 18(e), demonstrate a more accurate coverage of both low- and high-density regions.

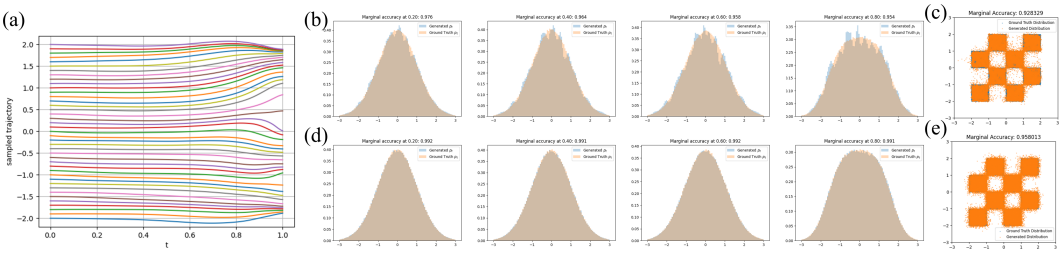


Figure 18: Results for the chessboard-shaped dataset. The figure descriptions are the same as in Figure 16.

I.2 Addressing the Collapse Errors in More Synthetic Datasets

To address the collapse error observed in ODE-based sampling methods, we applied a simple yet effective approach that separates the training of the score function into two distinct phases: one for learning the simpler score function associated with the Gaussian distribution and another for learning the more complex score function associated with the data distribution. This method aims to prevent interference between these phases and improve the overall robustness of the sampling process. The results, shown in Figures 19, 20, and 21, demonstrate the effectiveness of this approach across various synthetic datasets.

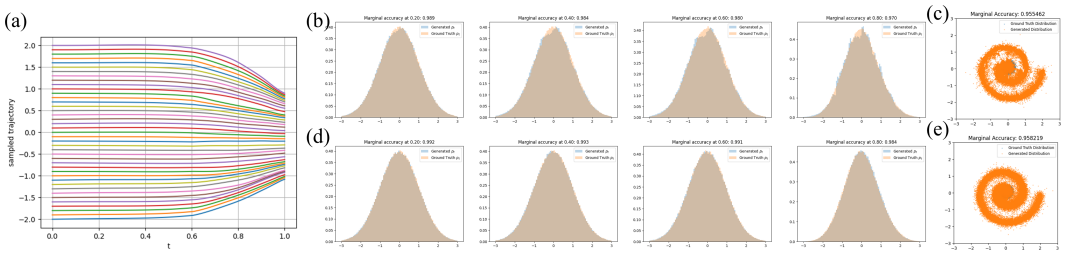


Figure 19: Results of the proposed method applied on the spiral-shaped dataset: (a) ODE sampling trajectories of the reverse diffusion process, starting from 40 evenly spaced points between -2 and 2. (b) Density plots of the generated samples using ODE-based sampling at different stages of the reverse diffusion process, corresponding to t=0.2,0.4,0.6,0.8. Marginal accuracy of the density is shown on the top of the figures. (c) Visualization of the ODE-generated data and ground truth data. (d) Density plots of the generated samples using SDE-based sampling at the same stages (t=0.2,0.4,0.6,0.8,1.0). Marginal accuracy of the density is shown on the top of the figures. (f) Visualization of the SDE-generated data and ground truth.

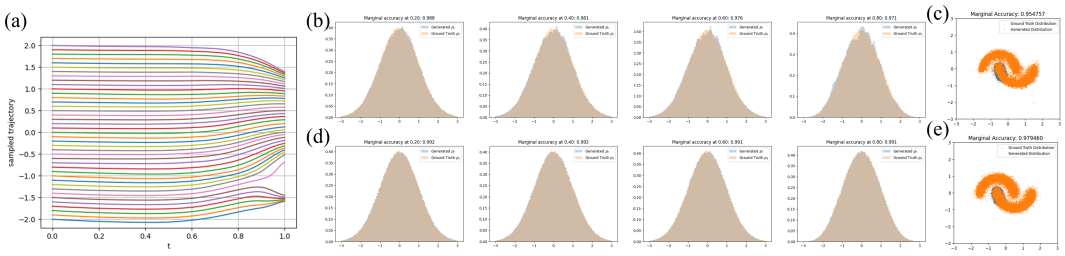


Figure 20: Results of the proposed method applied on the semi-circle-shaped dataset. The figure descriptions are the same as in Figure 19.

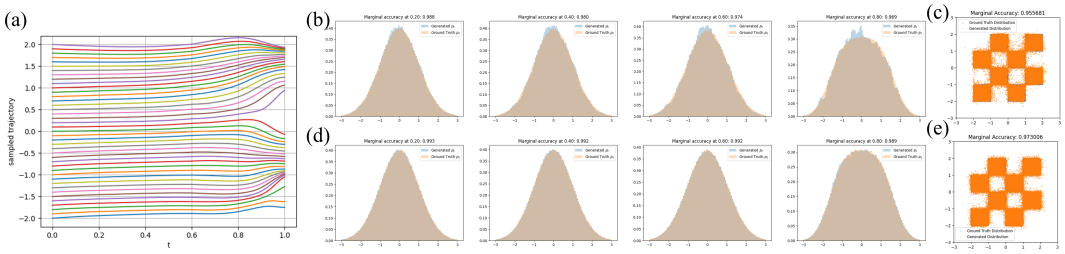


Figure 21: Results of the proposed method applied on the chessboard-shaped dataset. The figure descriptions are the same as in Figure 19.

In Figures 19(a), 20(a), and 21(a), the ODE sampling trajectories for the reverse diffusion process trained by the proposed method show a significant reduction in path concentration compared to those in Figures 16(a), 17(a), and 18(a). The trajectories are now more evenly spread across the data space, suggesting that the proposed solution successfully mitigates the issue of trajectory clustering, which previously led to a lack of diversity in the generated samples.

In Figures 19(b), 20(b), and 21(b), the density plots of the generated samples using the proposed method show less deviation from the Gaussian distribution, especially in the early stages, compared to the results in Figures 16(b), 17(b), and 18(b). This suggests that the proposed solution effectively reduces the deviation from the expected Gaussian-like properties, aligning more closely with the SDE sampling results that remain consistent with the ground truth.

Figures 19(c), 20(c), and 21(c) reveal that the seperate-model-training method also improves the distribution of the generated samples. In contrast to Figures 16(c), 17(bc), and 18(c), where the ODE-generated samples often incorrectly clustered in low-probability regions while failing to cover high-density areas adequately, the modified method produces a distribution that more accurately represents the ground truth, covering both high- and low-density regions more effectively.