Test-Time Alignment of Discrete Diffusion Models with Sequential Monte Carlo

Chinmay Pani [*1] **Zijing Ou** [*1] **Yingzhen Li** [1]

Abstract

Discrete diffusion models have become highly effective across various domains. However, real-world applications often require the generative process to adhere to certain constraints but without task-specific fine-tuning. To this end, we propose a training-free method based on Sequential Monte Carlo (SMC) to sample from the reward-aligned target distribution at the test time. Our approach leverages twisted SMC with an approximate locally optimal proposal, obtained via a first-order Taylor expansion of the reward function. To address the challenge of ill-defined gradients in discrete spaces, we incorporate a Gumbel-Softmax relaxation, enabling efficient gradient-based approximation within the discrete generative framework. Empirical results on both synthetic datasets and image modelling validate the effectiveness of our approach.

1. Introduction

We consider the task of the test-time alignment (Uehara et al., 2025) of pre-trained discrete diffusion models (DDMs) (Austin et al., 2021; Sahoo et al., 2024; Shi et al., 2024). This task has a broad applications such as molecular generation (Hoogeboom et al., 2022), materials discovery (Yang et al., 2023), and math reasoning (Zhao et al., 2025).

However, effective alignment of DDMs at test time remains a significant challenge. Existing alignment strategies mainly fall into two categories: i) fine-tuning and ii) guidance methods. Fine-tuning methods, including techniques such as steering (Rector-Brooks et al., 2024), reinforcement learning (Zekri & Boullé, 2025), and direct backpropagation (Wang et al., 2024), have demonstrated promising results. Nevertheless, they often suffer from reward over-optimisation, which can compromise sample quality and diversity. On the other hand, guidance methods (Li et al., 2024; Gruver et al., 2023; Nisonoff et al., 2024; Guo et al., 2024) provide training-free alternatives that are easier to deploy, but they often suffer from reward under-optimisation. This limits their ability to enforce correct alignment, resulting in outputs that may not fully meet complex objectives.

Inspired by the success of Sequential Monte Carlo (SMC) in continuous diffusion models (Wu et al., 2023; Kim et al., 2025), we explore its extension to discrete diffusion models. However, applying SMC in this context presents unique challenges due to the combinatorial explosion of possible states and the non-differentiable nature of discrete variables. To address these issues, we propose a twisted SMC framework that leverages an approximate locally optimal proposal distribution, obtained via a first-order Taylor expansion of the reward function. This design encourages more efficient exploration of high-reward regions in the sample space. To enable gradient-based updates in the discrete setting, we incorporate Gumbel-Softmax relaxation, which provides a continuous and differentiable approximation of the discrete variables. Our method remains training-free and asymptotically unbiased, and empirical results on both synthetic and image benchmarks show that it achieves effective reward alignment while preserving sample quality and diversity.

2. Background

Notation. Let $\mathcal{V}$ denote the set of one-hot vectors of length V, the number of categorical values. Discrete variables are represented as $\mathbf{z}_t, \mathbf{x} \in \mathcal{V}$, with distributions $\mathrm{Cat}(\mathbf{x}; \mathbf{p})$ where $\mathbf{p} \in \Delta^V$, the V-simplex. For L-dimensional data, we write $\mathbf{z}_t^{1:L}, \mathbf{x}^{1:L} \in \mathcal{V}^L$, and use $\mathbf{z}_t^l, \mathbf{x}^l$ for the l^{th} token.

2.1. Discrete Diffusion Models

Discrete diffusion models (Austin et al., 2021) define a forward process that interpolates data with prior $\boldsymbol{\pi} \in \Delta^V$

$$q(\mathbf{z}_t|\mathbf{x}) = \mathrm{Cat}(\mathbf{z}_t; \alpha_t \mathbf{x} + (1 - \alpha_t)\boldsymbol{\pi}), \tag{1}$$

where α_t is a monotonically decreasing noise schedule from 1 to 0, implying $\mathbf{z}_0 = \mathbf{x}$ and $\mathbf{z}_1 \sim \mathrm{Cat}(\boldsymbol{\pi})$. Masked diffusion models (Sahoo et al., 2024; Shi et al., 2024) are a special case that use an additional mask token $\boldsymbol{m}$ as the prior, with the induced posterior shown to be

$$q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x}) = \begin{cases} \mathrm{Cat}(\mathbf{z}_s; \mathbf{z}_t) & \mathbf{z}_t \neq \boldsymbol{m}, \\ \mathrm{Cat}(\mathbf{z}_s; \frac{(1-\alpha_s)\boldsymbol{m} + (\alpha_s - \alpha_t)\mathbf{x}}{1-\alpha_t}) & \mathbf{z}_t = \boldsymbol{m} \end{cases} \tag{2}$$

[*]Equal contribution Code will be available at https://github.com/J-zin/smc_ddm [1]Imperial College London. Correspondence to: Zijing Ou <z.o22@imperial.ac.uk.>.

Accepted at ICML 2025 PUT Workshop.

where $s < t$. Since, $\mathbf{x}$ is not available during inference, the reverse unmasking process is parametrised as,

$$p_\theta(\mathbf{z}_s|\mathbf{z}_t) = q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x}_\theta(\mathbf{z}_t, t)), \tag{3}$$

where $\mathbf{x}_\theta(\mathbf{z}_t, t)$ is the denoising model to predict the clean data $\mathbf{x}$. In practice, $\mathbf{x}_\theta(\mathbf{z}_t, t)$ is parametrised with the constraints: i) *Zero Masking Probabilities*: $\langle \mathbf{x}_\theta(\mathbf{z}_t, t), \boldsymbol{m} \rangle = 0$, ensuring no mass on the mask token; and ii) *Carry-Over Unmasking*: if $\mathbf{z}_t \neq \boldsymbol{m}$, then $\mathbf{x}_\theta(\mathbf{z}_t, t) = \mathbf{z}_t$. To enable efficient sampling in the multi-dimensional case, the reverse unmasking process is further assumed to factorise independently across dimensions given $\mathbf{z}_t^{1:L}$, i.e.,

$$p_\theta(\mathbf{z}_s^{1:L}|\mathbf{z}_t^{1:L}) = \prod_{l=1}^{L} p_\theta(\mathbf{z}_s^l|\mathbf{z}_t^{1:L}) \triangleq \prod_{l=1}^{L} q(\mathbf{z}_s^l|\mathbf{z}_t^l, \mathbf{x}_\theta(\mathbf{z}_t^{1:L}, t)^l).$$

2.2. Sequential Monte Carlo

Sequential Monte Carlo (SMC) is a general framework for approximately sampling from a sequence of intermediate distributions π_t. It does so by employing a forward proposal $\mathcal{F}_s(\mathbf{x}_s|\mathbf{x}_t)$, from which particles are drawn and subsequently resampled based on importance weights

$$w_s(\mathbf{x}_s) = w_t(\mathbf{x}_t) \frac{\pi_s(\mathbf{x}_s)}{\pi_t(\mathbf{x}_t)} \frac{\mathcal{B}_s(\mathbf{x}_t|\mathbf{x}_s)}{\mathcal{F}_s(\mathbf{x}_s|\mathbf{x}_t)}, \tag{4}$$

where $\mathcal{B}_s$ denotes the backward distribution. While SMC is asymptotically consistent, its practical efficacy typically suffers from weight degeneracy, an issue that arises when only a few particles carry significant weight. This problem is especially severe when the proposal distribution poorly matches the target or when the intermediate distributions differ significantly, necessitating careful design of both the proposal and resampling strategies to ensure robust performance (Del Moral et al., 2006).

3. Test-Time Alignment of DDMs with SMC

We consider the task of sampling from the target distribution

$$p_{\text{tar}}(\mathbf{x}^{1:L}) \propto p_{\text{pre}}(\mathbf{x}^{1:L}) \exp\left(\frac{r(\mathbf{x}^{1:L})}{\alpha}\right), \tag{5}$$

where p_{pre} is a pre-trained discrete diffusion model and r is a reward model. p_{tar} is also known to be the optimum of the following fine-tuning objective (Uehara et al., 2024)

$$p_{\text{tar}} = \arg\max_p \mathbb{E}_{\mathbf{x}^{1:L} \sim p}[r(\mathbf{x}^{1:L})] - \alpha D_{\text{KL}}(p \| p_{\text{pre}}), \tag{6}$$

where the coefficient α controls the trade-off between reward maximisation and staying close to the pre-trained model. Instead of optimising this objective directly, which often leads to reward overoptimisation (Clark et al., 2023), we propose to sample from p_{tar} directly using SMC.

3.1. Twisted SMC with Locally Optimal Proposal

A key ingredient of SMC is the intermediate targets π_t with the terminal potential $\pi_0 = p_{\text{tar}}$ as in Eq. (5). Inspired by Wu et al. (2023); Kim et al. (2025), we define these target as

$$\pi_t(\mathbf{z}_t^{1:L}) \propto p_t(\mathbf{z}_t^{1:L}) \exp\left(\frac{\lambda_t}{\alpha} \hat{r}(\mathbf{z}_t^{1:L})\right), \tag{7}$$

where $p_t(\mathbf{z}_t^{1:L})$ denotes the marginal distribution induced by the pre-trained diffusion model p_θ, $\hat{r}(\mathbf{z}_t^{1:L}) = \mathbb{E}_{\mathbf{x}^{1:L} \sim \mathbf{x}_\theta(\mathbf{z}_t^{1:L}, t)}[r(\mathbf{x}^{1:L})]$ is the estimated reward, and λ_t is a temperature parameter that increases monotonically from 0 to 1 as t goes from 1 to 0. To ensure the tractability of importance weights, the backward kernel is defined as

$$\mathcal{B}_t(\mathbf{z}_t^{1:L}|\mathbf{z}_s^{1:L}) = \frac{p_\theta(\mathbf{z}_s^{1:L}|\mathbf{z}_t^{1:L}) p_t(\mathbf{z}_t^{1:L})}{p_s(\mathbf{z}_s^{1:L})}, \tag{8}$$

which yields the following weight update:

$$w_s(\mathbf{z}_s^{1:L}) = \frac{p_\theta(\mathbf{z}_s^{1:L}|\mathbf{z}_t^{1:L})}{\mathcal{F}_s(\mathbf{z}_s^{1:L}|\mathbf{z}_t^{1:L})} \cdot \frac{\exp\left(\frac{\lambda_s}{\alpha} \hat{r}(\mathbf{z}_s^{1:L})\right)}{\exp\left(\frac{\lambda_t}{\alpha} \hat{r}(\mathbf{z}_t^{1:L})\right)} \cdot w_t(\mathbf{z}_t^{1:L}) \tag{9}$$

As previously discussed, the choice of proposal distribution $\mathcal{F}_s$ significantly impacts the performance of SMC. A straightforward option is to use the reverse process of the pre-trained diffusion model, leading to the proposal:

$$\mathcal{F}_s(\mathbf{z}_s^{1:L}|\mathbf{z}_t^{1:L}) = p_\theta(\mathbf{z}_s^{1:L}|\mathbf{z}_t^{1:L}). \tag{10}$$

Since p_θ is factorised, it is easy to sample from and allows efficient evaluation of the probability mass. However, in practice, this naive choice often leads to high variance in the particle weights. To mitigate this issue, one can instead use the locally optimal proposal, which minimises the variance of the importance weights:

$$\mathcal{F}_s^*(\mathbf{z}_s^{1:L}|\mathbf{z}_t^{1:L}) \propto p_\theta(\mathbf{z}_s^{1:L}|\mathbf{z}_t^{1:L}) \exp\left(\frac{\lambda_s}{\alpha} \hat{r}(\mathbf{z}_s^{1:L})\right) \tag{11}$$

The factors $\exp\left(\frac{\lambda_s}{\alpha} \hat{r}(\mathbf{z}_s^{1:L})\right)$ serve as the optimal twisting functions, enabling SMC to produce exact samples from the target distribution π_t even with a single particle. However, computing them is generally intractable in practice, as the reward function is typically non-factorizable. This necessitates evaluating the reward $\mathcal{O}(V^L)$ times, which is only computationally feasible in low-dimensional scenarios with small vocabularies. In high-dimensional tasks, approximated alternatives are required to maintain tractability.

3.2. First order Approx. Locally Optimal Proposal

To make Eq. (11) tractable, we consider using a first-order Taylor approximation to approximate the twisting factor:

$$\begin{aligned} \hat{r}(\mathbf{z}_s^{1:L}) &\approx \hat{r}(\mathbf{z}_t^{1:L}) + \langle \nabla_{\mathbf{z}_t^{1:L}} \hat{r}(\mathbf{z}_t^{1:L}), \mathbf{z}_s^{1:L} - \mathbf{z}_t^{1:L} \rangle \\ &= \hat{r}(\mathbf{z}_t^{1:L}) + \sum_{l=1}^{L} \langle \nabla_{\mathbf{z}_t^l} \hat{r}(\mathbf{z}_t^{1:L}), \mathbf{z}_s^l - \mathbf{z}_t^l \rangle. \end{aligned} \tag{12}$$

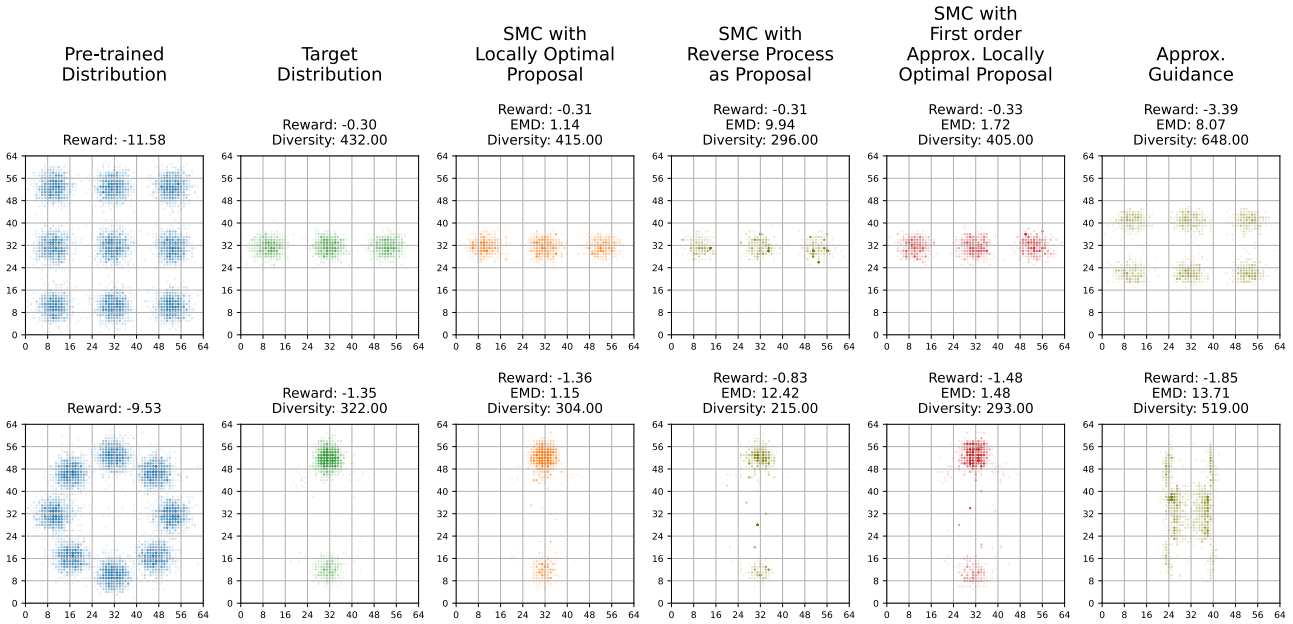

Figure 1. From left to right: samples from the pre-trained distribution p_{pre}; samples from the target distribution p_{tar}, samples generated using tempered SMC with the locally optimal proposal (Eq. (11)), the reverse process as proposal (Eq. (10)), and the first order Taylor approximation of the locally optimal proposal (Eq. (17)); samples generated using approximate guidance (Eq. (39)). KL weight $\alpha = 1$ for both distributions. Top row: reward $r(X, Y) = -\hat{X}^2/100 - \hat{Y}^2$, bottom row: reward $r(X, Y) = -\hat{X}^2 - (\hat{Y} - 1)^2/10$, where $\hat{X} = 12(X/63 - 1/2)$ and $\hat{Y} = 12(Y/63 - 1/2)$. Note that the rewards are actually computed as a differentiable function of one hot inputs corresponding to X and Y, more details in Appendix F.2.1.

.

However, since $\mathbf{z}_t^{1:L}$ is discrete, the gradient $\nabla_{\mathbf{z}_t^{1:L}} \hat{r}(\mathbf{z}_t^{1:L})$ is not well-defined. To solve this problem, we break down $\hat{r}(\mathbf{z}_t^{1:L})$ into three steps:

$$p_{\mathbf{x}} = \mathbf{x}_\theta(\mathbf{z}_t^{1:L}, t) \tag{13}$$

$$\mathbf{x}^l \sim \text{Cat}(\mathbf{x}^l; p_{\mathbf{x}}^l) \quad \forall l \in \{1, \dots, L\} \tag{14}$$

$$\hat{r}(\mathbf{z}_t^{1:L}) = r(\mathbf{x}^{1:L}) \tag{15}$$

We propose the following method to make sure that each of these steps are differentiable. For Eq. (14), we use the reparametrization trick using Gumbel-Softmax (Jang et al., 2016) to make $\mathbf{x}^l$ differentiable with respect to $p_{\mathbf{x}}^l$. For both Equations (13) and (15), similar to Grathwohl et al. (2021); Zhang et al. (2022), we can treat both $r(\cdot)$ and $\mathbf{x}_\theta(\cdot, t)$ as functions which can take continuous real-valued inputs, in order to be able to take their gradients. Additionally, for $\mathbf{x}_\theta$, to avoid the discontinuity induced by the *Carry-Over Unmasking* constraint, we propose the following equivalent continuous formulation:

$$\mathbf{x}_\theta(\mathbf{z}_t^{1:L}, t)^l = \gamma \tilde{\mathbf{x}}_\theta(\mathbf{z}_t^{1:L}, t)^l + (\mathbf{1} - \boldsymbol{m}) \odot \mathbf{z}_t^l \tag{16}$$

where $\tilde{\mathbf{x}}_\theta(\mathbf{z}_t^{1:L}, t)$ is the output of the denoising model after applying the *Zero Masking Probabilities* constraint, and γ is a scalar which can be defined as $\gamma = 1 - \langle \mathbf{1} - \boldsymbol{m}, \mathbf{z}_t^l \rangle$. Notice that $\gamma = 1$ when $\mathbf{z}_t^l = \boldsymbol{m}$ and 0 otherwise. Now that all three steps are differentiable with well-defined gradients, we can compute $\nabla_{\mathbf{z}_t^{1:L}} \hat{r}(\mathbf{z}_t^{1:L})$ using automatic differentiation. Substituting $\hat{r}(\mathbf{z}_s^{1:L})$ from Eq. (12) in Eq. (11) leads to the first order approximated locally optimal proposal:

$$\mathcal{F}_s(\mathbf{z}_s^{1:L} | \mathbf{z}_t^{1:L}) \propto p_\theta(\mathbf{z}_s^{1:L} | \mathbf{z}_t^{1:L}) \times \exp\Big(\frac{\lambda_s}{\alpha} \sum_{l=1}^{L} \big\langle \nabla_{\mathbf{z}_t^l} \hat{r}(\mathbf{z}_t^{1:L}),\, \mathbf{z}_s^l \big\rangle\Big). \tag{17}$$

This proposal can be further factorised as:

$$\mathcal{F}_s(\mathbf{z}_s^{1:L} | \mathbf{z}_t^{1:L}) = \prod_{l=1}^{L} \mathcal{F}_s(\mathbf{z}_s^l | \mathbf{z}_t^{1:L}),$$

$$\mathcal{F}_s(\mathbf{z}_s^l | \mathbf{z}_t^{1:L}) = p_\theta(\mathbf{z}_s^l | \mathbf{z}_t^{1:L}) \exp\Big(\frac{\lambda_s}{\alpha} \langle \nabla_{\mathbf{z}_t^l} \hat{r}(\mathbf{z}_t^{1:L}), \mathbf{z}_s^l \rangle\Big), \tag{18}$$

which facilitates efficient sampling, as it requires evaluating and differentiating the function $\hat{r}(\cdot)$ only once at $\mathbf{z}_t^{1:L}$. The final algorithm is summarised in Alg. 1.

4. Experiments

We evaluate the proposed method on both a synthetic dataset and an image modelling task. Detailed experimental settings and additional results are presented in Appendix F.

Figure 2. Binarised MNIST samples generated for $y_{\text{target}} = 4$, $\alpha = 1$ using a masked diffusion model with tempered SMC (left and middle), and approximate guidance (right).

4.1. Mixture of Gaussians

Settings. We created discretised versions of the mixture of Gaussian dataset whose data points take integer values between 0 and 63. For each dataset, we train a masked discrete diffusion model using an MLP with 2 hidden layers as the denoising model. For the twisted SMC, we use a linear temperature schedule, $\lambda_t = 1 - t$ and 2000 particles. We calculate the estimated reward $\hat{r}(\mathbf{z}_t^{1:L})$ by taking 100 samples from $\mathbf{x}_\theta(\mathbf{z}_t^{1:L}, t)$ and calculating the mean reward.

Results. In Fig. 1, we provide a comparison of the resulting samples when using twisted SMC with different proposals, and when using approximate guidance only. We report the mean reward achieved, EMD (earth mover's distance) w.r.t. the target distribution, and sample diversity (the number of unique samples). SMC with the locally optimal proposal gives the best results, achieving the required reward value while having a low EMD and high sample diversity. SMC with the first order approximation of the locally optimal proposal also achieves comparable results. When using the reverse process as the proposal, there is a significant dip in sample diversity, which is caused due the higher variance in weights, which in turn leads to more resampling steps. Lastly, the approximate guidance performs the worst as the gradient term overpowers the diffusion reverse process, leading to samples which disregard the pre-trained distribution. A further analysis on the effect of using different proposals is presented in Appendix F.2.2.

4.2. Binarised MNIST

Settings. We construct a binarised version of the MNIST dataset, where each pixel is assigned a value of either 0 or 1. For the denoising model, we adopt a U-Net architecture following Ho et al. (2020). To define the reward function, we first train a classifier $p_\phi(y|\mathbf{x}^{1:L})$ on the clean data.The reward is then given by $r(\mathbf{x}^{1:L}) = \log p_\phi(y = y_{\text{target}}|\mathbf{x}^{1:L})$, where y_{target} can be any target digit. For the SMC, we use 20 particles and set the tempering schedule as $\lambda_t = \min(1.05^{T(1-t)} - 1, 1)$, where $T = 100$ is the number of discrete time steps for inference. We calculate the estimated reward $\hat{r}(\mathbf{z}_t^{1:L})$ by taking a single sample from $\mathbf{x}_\theta(\mathbf{z}_t^{1:L}, t)$ and passing it through the classifier. Additionally, we use the partial resampling scheme (Martino et al., 2016), in which only half of the particles are resampled when the ESS threshold is breached. This approach mitigates mode collapse, especially in high dimensions (Lee et al., 2025).

Results. In Fig. 2, we provide the resulting particles using different sampling methods for $y_{\text{target}} = 4$ and $\alpha = 1$. SMC with first order Taylor approximation of the locally optimal proposal achieves the maximum possible reward of 0, while generating diverse and high fidelity samples. Using the reverse as the proposal, relies solely on the SMC resampling to weed out incorrect digits from the particle set. Some incorrect digits still show up in the final particle set as resampling only takes place when the ESS drops below a certain threshold. Lastly, when using approximate guidance, most particles become corrupted. This is due to inexactness of guidance in the early steps, paired with the fact that in masked diffusion, pixels once unmasked at any time step can no longer be masked or modified.

The above stated property of masked diffusion models also makes it somewhat difficult to sample using the twisted SMC method. We needed to carefully choose the tempering schedule to get good results. Since, unmasked pixels can not be modified further, we need to ensure that a sufficient amount of guidance is injected into the proposal early on before the particles evolve into an incorrect digit. At the same time, we cannot increase the temperature too fast since the guidance can be inaccurate in earlier time steps. This means we need to find a balance. Our choice of tempering schedule for this experiment is visualised in Fig. 6. This problem is resolved if we use either ReMDM (Wang et al., 2025) or UDLM (Schiff et al., 2024) in place of masked diffusion, as they allow pixels to be modified and guided throughout the entire sampling process. We provide the details of our experiments with ReMDM and UDLM on the binarised MNIST dataset in Appendix F.3.

5. Conclusions and Limitations

In this work, we present a method for test-time reward alignment of discrete diffusion models using twisted SMC. We introduce a first-order Taylor approximation of the locally optimal proposal, and provide methods to counter the discontinuities when computing the gradients of the estimated reward for masked diffusion models. The effectiveness of our method is validated through empirical results on both synthetic and image datasets. We also discuss limitations and potential future work in Appendix G.

References

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Black, K., Janner, M., Du, Y., Kostrikov, I., and Levine, S. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Borso, U., Paglieri, D., Wells, J., and Rocktäschel, T. Preference-based alignment of discrete diffusion models. *arXiv preprint arXiv:2503.08295*, 2025.

Cao, H., Shi, H., Wang, C., Pan, S. J., and Heng, P.-A. Glid^{2}e: A gradient-free lightweight fine-tune approach for discrete sequence design. In *ICLR 2025 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2025.

Cardoso, G., Idrissi, Y. J. E., Corff, S. L., and Moulines, E. Monte carlo guided diffusion for bayesian linear inverse problems. *arXiv preprint arXiv:2308.07983*, 2023.

Clark, K., Vicol, P., Swersky, K., and Fleet, D. J. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023.

Del Moral, P., Doucet, A., and Jasra, A. Sequential monte carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436, 2006.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Dou, Z. and Song, Y. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *The Twelfth International Conference on Learning Representations*, 2024.

Grathwohl, W., Swersky, K., Hashemi, M., Duvenaud, D., and Maddison, C. Oops i took a gradient: Scalable sampling for discrete distributions. In *International Conference on Machine Learning*, pp. 3831–3841. PMLR, 2021.

Gruver, N., Stanton, S., Frey, N., Rudner, T. G., Hotzel, I., Lafrance-Vanasse, J., Rajpal, A., Cho, K., and Wilson, A. G. Protein design with guided discrete diffusion. *Advances in neural information processing systems*, 36: 12489–12517, 2023.

Guo, W., Zhu, Y., Tao, M., and Chen, Y. Plug-and-play controllable generation for discrete masked models. *arXiv preprint arXiv:2410.02143*, 2024.

He, J., Hernández-Lobato, J. M., Du, Y., and Vargas, F. Rne: a plug-and-play framework for diffusion density estimation and inference-time control. *arXiv preprint arXiv:2506.05668*, 2025.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Holderrieth, P., Albergo, M. S., and Jaakkola, T. Leaps: A discrete neural sampler via locally equivariant networks. *arXiv preprint arXiv:2502.10843*, 2025.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Kim, S., Kim, M., and Park, D. Alignment without overoptimization: Training-free solution for diffusion models. *arXiv preprint arXiv:2501.05803*, 2025.

Lee, C. K., Jeha, P., Frellsen, J., Lio, P., Albergo, M. S., and Vargas, F. Debiasing guidance for discrete diffusion with sequential monte carlo. *arXiv preprint arXiv:2502.06079*, 2025.

Li, X., Thickstun, J., Gulrajani, I., Liang, P. S., and Hashimoto, T. B. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343, 2022.

Li, X., Zhao, Y., Wang, C., Scalia, G., Eraslan, G., Nair, S., Biancalani, T., Ji, S., Regev, A., Levine, S., et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024.

Lovelace, J., Kishore, V., Wan, C., Shekhtman, E., and Weinberger, K. Q. Latent diffusion for language generation. *Advances in Neural Information Processing Systems*, 36: 56998–57025, 2023.

Martino, L., Elvira, V., and Louzada, F. Weighting a resampled particle in sequential monte carlo. In *2016 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 1–5. IEEE, 2016.

Ninniri, M., Podda, M., and Bacciu, D. Classifier-free graph diffusion for molecular property targeting. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 318–335. Springer, 2024.

Nisonoff, H., Xiong, J., Allenspach, S., and Listgarten, J. Unlocking guidance for discrete state-space diffusion and flow models. *arXiv preprint arXiv:2406.01572*, 2024.

Norris, J. R. *Markov chains*. Number 2. Cambridge university press, 1998.

Oksendal, B. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

Ou, Z., Zhang, R., and Li, Y. Discrete neural flow samplers with locally equivariant transformer. *arXiv preprint arXiv:2505.17741*, 2025.

Phillips, A., Dau, H.-D., Hutchinson, M. J., De Bortoli, V., Deligiannidis, G., and Doucet, A. Particle denoising diffusion sampler. *arXiv preprint arXiv:2402.06320*, 2024.

Rector-Brooks, J., Hasan, M., Peng, Z., Quinn, Z., Liu, C., Mittal, S., Dziri, N., Bronstein, M., Bengio, Y., Chatterjee, P., et al. Steering masked discrete diffusion models via discrete denoising posterior prediction. *arXiv preprint arXiv:2410.08134*, 2024.

Sahoo, S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J., Rush, A., and Kuleshov, V. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Schiff, Y., Sahoo, S. S., Phung, H., Wang, G., Boshar, S., Dalla-torre, H., de Almeida, B. P., Rush, A., Pierrot, T., and Kuleshov, V. Simple guidance mechanisms for discrete diffusion models. *arXiv preprint arXiv:2412.10193*, 2024.

Shi, J., Han, K., Wang, Z., Doucet, A., and Titsias, M. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167, 2024.

Singhal, R., Horvitz, Z., Teehan, R., Ren, M., Yu, Z., McKeown, K., and Ranganath, R. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

Skreta, M., Akhound-Sadegh, T., Ohanesian, V., Bondesan, R., Aspuru-Guzik, A., Doucet, A., Brekelmans, R., Tong, A., and Neklyudov, K. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts. *arXiv preprint arXiv:2503.02819*, 2025.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Uehara, M., Zhao, Y., Biancalani, T., and Levine, S. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*, 2024.

Uehara, M., Zhao, Y., Wang, C., Li, X., Regev, A., Levine, S., and Biancalani, T. Reward-guided controlled generation for inference-time alignment in diffusion models: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025.

Vignac, C., Krawczuk, I., Siraudin, A., Wang, B., Cevher, V., and Frossard, P. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

Wallace, B., Dang, M., Rafailov, R., Zhou, L., Lou, A., Purushwalkam, S., Ermon, S., Xiong, C., Joty, S., and Naik, N. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Wang, C., Uehara, M., He, Y., Wang, A., Biancalani, T., Lal, A., Jaakkola, T., Levine, S., Wang, H., and Regev, A. Fine-tuning discrete diffusion models via reward optimization with applications to dna and protein design. *arXiv preprint arXiv:2410.13643*, 2024.

Wang, G., Schiff, Y., Sahoo, S. S., and Kuleshov, V. Remasking discrete diffusion models with inference-time scaling. *arXiv preprint arXiv:2503.00307*, 2025.

Wu, L., Trippe, B., Naesseth, C., Blei, D., and Cunningham, J. P. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023.

Yang, S., Cho, K., Merchant, A., Abbeel, P., Schuurmans, D., Mordatch, I., and Cubuk, E. D. Scalable diffusion for materials generation. *arXiv preprint arXiv:2311.09235*, 2023.

Yuan, H., Huang, K., Ni, C., Chen, M., and Wang, M. Reward-directed conditional diffusion: Provable distribution estimation and reward improvement. *Advances in Neural Information Processing Systems*, 36:60599–60635, 2023.

Zekri, O. and Boullé, N. Fine-tuning discrete diffusion models with policy gradient methods. *arXiv preprint arXiv:2502.01384*, 2025.

Zhang, L., Rao, A., and Agrawala, M. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

Zhang, R., Liu, X., and Liu, Q. A langevin-like sampler for discrete distributions. In *International Conference on Machine Learning*, pp. 26375–26396. PMLR, 2022.

Zhao, S., Gupta, D., Zheng, Q., and Grover, A. d1: Scaling reasoning in diffusion large language models via reinforcement learning. *arXiv preprint arXiv:2504.12216*, 2025.

Appendix for "Test-Time Alignment of Discrete Diffusion Models with Sequential Monte Carlo"

Contents

A. Other Discrete Diffusion Models

A major challenge in alignment or guidance for masked diffusion models is that latent variables become immutable once they are assigned a non-mask category at any point during the denoising process. As a result, any unmasking errors that occur are irreversible and persist in the final samples. In this section, we briefly introduce few other types of discrete diffusion models. Notably, these models allow latent variables to remain updatable throughout the entire denoising process.

A.1. Remasking Diffusion Model

Remasking Diffusion Model (ReMDM) (Wang et al., 2025) is a modification of masked diffusion model which allows remasking unmasked tokens during the denoising process. The posteriors are constructed in such a way that the forward marginals $q(\mathbf{z}_t|\mathbf{x})$ remain the same as masked diffusion.

$$q_\sigma(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x}) = \begin{cases} \mathrm{Cat}(\mathbf{z}_s;(1-\sigma_t)\mathbf{x}+\sigma_t\boldsymbol{m}) & \mathbf{z}_t \neq \boldsymbol{m} \\ \mathrm{Cat}(\mathbf{z}_s;\frac{\alpha_s-(1-\sigma_t)\alpha_t}{1-\alpha_t}\mathbf{x}+\frac{1-\alpha_s-\sigma_t\alpha_t}{1-\alpha_t}\boldsymbol{m}) & \mathbf{z}_t = \boldsymbol{m} \end{cases} \tag{19}$$

Here, σ_t is the remasking schedule, and to ensure that the posterior is valid, it must follow the constraints:

$$0 \leq \sigma_t \leq \min\left\{1, \frac{1-\alpha_s}{\alpha_t}\right\} =: \sigma_t^{max}$$

The reverse unmasking process is parametrized as,

$$p_\theta(\mathbf{z}_s|\mathbf{z}_t) = q_\sigma(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x}_\theta(\mathbf{z}_t,t)). \tag{20}$$

Finally, the ReMDM loss objective becomes a re-weighted version of the loss objective for masked diffusion models. Thus, we can take a pre-trained masked diffusion model, and use the ReMDM parametrized reverse process in Eq. (20) for inference.

A.2. Uniform Noise Diffusion Models

Uniform Noise Diffusion Models (UDLM) (Schiff et al., 2024) is a discrete diffusion model where the fixed prior $\boldsymbol{\pi} = \boldsymbol{u} := \mathbf{1}/V$, where V is the vocabulary size or the number of possible categorical values. The resulting posterior is,

$$q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x}) = \text{Cat}\left(\mathbf{z}_s; \frac{V\alpha_t \mathbf{z}_t \odot \mathbf{x} + \left(\frac{\alpha_t}{\alpha_s} - \alpha_t\right)\mathbf{z}_t + (\alpha_s - \alpha_t)\mathbf{x} + \frac{(\alpha_s - \alpha_t)(1-\alpha_s)}{V\alpha_s}\mathbf{1}}{V\alpha_t\langle \mathbf{z}_t, \mathbf{x}\rangle + 1 - \alpha_t}\right) \tag{21}$$

The reverse process is parametrized similar to masked diffusion models as:

$$p_\theta(\mathbf{z}_s|\mathbf{z}_t) = q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x}_\theta(\mathbf{z}_t, t))$$

Unlike masked diffusion models, the denoising model $\mathbf{x}_\theta(\mathbf{z}_t, t)$ has no additional constraints except that it should output a valid categorical distribution, i.e., $\mathbf{x}_\theta(\mathbf{z}_t, t) \in \Delta^V$. Thus, the continuous formulation introduced in Sec. 3.2 is not relevant for computing the gradients for the first order Taylor approximation in case of UDLM.

B. Extending Discrete-time SMC to Continuous-time SMC

In this section, we establish the connection between the discrete-time SMC formulation in Eq. (4) and its continuous-time counterparts, as studied in (Ou et al., 2025; Holderrieth et al., 2025; Lee et al., 2025). We begin by recapitulating continuous-time Markov chains (CTMC) (Norris, 1998) in Appendix B.1 and then discuss the connection in Appendix B.2.

B.1. Background of CTMC

A CTMC (Norris, 1998) is characterised by a time-dependent transition rate matrix R_t, defined as

$$R_t(\mathbf{x}, \mathbf{y}) = \lim_{\Delta t \to 0} \frac{p_{t+\Delta t|t}(\mathbf{y}|\mathbf{x}) - \delta_{\mathbf{y}=\mathbf{x}}}{\Delta t}. \tag{22}$$

By definition, the transition probability is given by

$$p_{t+\Delta t|t}(\mathbf{y}|\mathbf{x}) = \delta_{\mathbf{y}=\mathbf{x}} + R_t(\mathbf{x}, \mathbf{y})\Delta t + \mathcal{O}(\Delta t). \tag{23}$$

To ensure the transition probability is valid, the rate matrix R_t should satisfy

$$R_t(\mathbf{x}, \mathbf{y}) \geq 0, \forall \mathbf{y} \neq \mathbf{x}, \quad R_t(\mathbf{x}, \mathbf{x}) = -\sum_{\mathbf{y}\neq\mathbf{x}} R_t(\mathbf{x}, \mathbf{y}). \tag{24}$$

The CTMC transition probabilities satisfy the Kolmogorov forward and backward equations (Oksendal, 2013). For $t > s$

$$\text{Kolmogorov forward equation:} \quad \partial_t q_{t|s}(\mathbf{x}|\tilde{\mathbf{x}}) = \sum_{\mathbf{y}} q_{t|s}(\mathbf{y}|\tilde{\mathbf{x}}) R_t(\mathbf{y}, \mathbf{x})$$

$$\text{Kolmogorov backward equation:} \quad \partial_s q_{t|s}(\mathbf{x}|\tilde{\mathbf{x}}) = -\sum_{\mathbf{y}} R_t(\tilde{\mathbf{x}}, \mathbf{y}) q_{t|s}(\mathbf{x}|\mathbf{y})$$

The forward equation also induces a PDE for the marginals of CTMC

$$\partial_t q_t(\mathbf{x}) = \sum_{\mathbf{y}} q_t(\mathbf{y}) R_t(\mathbf{y}, \mathbf{x}). \tag{25}$$

Given a test function h of interest, define $u_t(\mathbf{x}) = \mathbb{E}_{q_{1|t}(\mathbf{z}|\mathbf{x})}[h(\mathbf{z})]$, the backward equation gives

$$\begin{aligned}\partial_t u_t(\mathbf{x}) &= \sum_{\mathbf{z}} h(\mathbf{z}) \partial_t q_{1|t}(\mathbf{z}|\mathbf{x}) \\ &= \sum_{\mathbf{z}} h(\mathbf{z}) - \sum_{y} R_t(\mathbf{x}, \mathbf{y}) q_{1|t}(\mathbf{z}, \mathbf{y}) \\ &= -\sum_{y} R_t(\mathbf{x}, \mathbf{y}) \sum_{\mathbf{z}} q_{1|t}(\mathbf{z}, \mathbf{y}) h(\mathbf{z}) \\ &= -\sum_{y} R_t(\mathbf{x}, \mathbf{y}) u_t(\mathbf{y}).\end{aligned}$$

B.2. CTMC Formulation of SMC

Recall Eq. (4), where the importance weight is given by

$$w_s(\mathbf{x}_s) = w_t(\mathbf{x}_t) \frac{\pi_s(\mathbf{x}_s)}{\pi_t(\mathbf{x}_t)} \frac{\mathcal{B}_s(\mathbf{x}_t|\mathbf{x}_s)}{\mathcal{F}_s(\mathbf{x}_s|\mathbf{x}_t)}. \tag{26}$$

We now extend this formulation to the continuous-time setting. Let R_t and $\hat{R}_t$ denote the rate matrices corresponding to the forward proposal $\mathcal{F}_t$ and the backward transition $\mathcal{B}_t$, respectively. Consider a discretization with N denoising steps, indexed by time points $s = t_N < \cdots < t_i \cdots < t_0 = 1$, where each interval satisfies $t_{i-1} - t_i = \frac{1}{N}$. The discrete-time importance weight at step time s is then computed as

$$\log w_s = \log \frac{\pi_s(\mathbf{x}_s)}{\pi_1(\mathbf{x}_1)} + \sum_{i=1}^{N} \log \frac{\mathcal{B}_{t_i}(\mathbf{x}_{t_{i-1}}|\mathbf{x}_{t_i})}{\mathcal{F}_{t_i}(\mathbf{x}_{t_i}|\mathbf{x}_{t_{i-1}})}. \tag{27}$$

The second term in the RHS can be computed as

$$\begin{aligned}\log \frac{\mathcal{B}_{t_i}(\mathbf{x}_{t_{i-1}}|\mathbf{x}_{t_i})}{\mathcal{F}_{t_i}(\mathbf{x}_{t_i}|\mathbf{x}_{t_{i-1}})} &= \log\left(\delta_{\mathbf{x}_{t_{i-1}}=\mathbf{x}_{t_i}} + \hat{R}_{t_i}(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i-1}})\frac{1}{N}\right) - \log\left(\delta_{\mathbf{x}_{t_i}=\mathbf{x}_{t_{i-1}}} + R_{t_i}(\mathbf{x}_{t_{i-1}}, \mathbf{x}_{t_i})\frac{1}{N}\right) \\ &= \sum_{i, t_i = t_{i-1}} \log\left(1 + \hat{R}_{t_i}(\mathbf{x}_{t_i}, \mathbf{x}_{t_i})\frac{1}{N}\right) - \log\left(1 + R_{t_i}(\mathbf{x}_{t_i}, \mathbf{x}_{t_i})\frac{1}{N}\right) + \sum_{i, t_i \neq t_{i-1}} \hat{R}_{t_i}(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i-1}}) - R_{t_i}(\mathbf{x}_{t_{i-1}}, \mathbf{x}_{t_i}) \\ &= \sum_{i, t_i = t_{i-1}} \hat{R}_{t_i}(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i-1}})\frac{1}{N} - R_{t_i}(\mathbf{x}_{t_{i-1}}, \mathbf{x}_{t_i})\frac{1}{N} + \mathcal{O}(\frac{1}{N}) + \sum_{i, t_i \neq t_{i-1}} \hat{R}_{t_i}(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i-1}}) - R_{t_i}(\mathbf{x}_{t_{i-1}}, \mathbf{x}_{t_i})\end{aligned}$$

Taking the limit as $N \to +\infty$, the importance weight becomes:

$$\log w_s = \log \frac{\pi_s(\mathbf{x}_s)}{\pi_1(\mathbf{x}_1)} + \int_1^s R_t(\mathbf{x}_t, \mathbf{x}_t) - \hat{R}_t(\mathbf{x}_t, \mathbf{x}_t)\, \mathrm{d}t + \sum_{s \leq t, \mathbf{x}_{t^+} \neq \mathbf{x}_t} \log \hat{R}_t(\mathbf{x}_t, \mathbf{x}_{t^+}) - \log R_t(\mathbf{x}_{t^+}, \mathbf{x}_t). \tag{28}$$

By the fundamental theorem of calculus for piecewise differentiable functions, we also have:

$$\log \frac{\pi_s(\mathbf{x}_s)}{\pi_1(\mathbf{x}_1)} = \int_1^s -\partial_t \log \pi_t(\mathbf{x}_t) + \sum_{s \leq t, \mathbf{x}_{t^+} \neq \mathbf{x}_t} \log \pi_t(\mathbf{x}_t) - \log \pi_t(x_{t^+}). \tag{29}$$

If the backward process is chosen such that the rate matrix satisfies the detailed balance condition: $\hat{R}_t(\mathbf{x}, \mathbf{y}) p_t(\mathbf{x}) = R_t(\mathbf{y}, \mathbf{x}) p_t(\mathbf{y})$, then the importance weight simplifies accordingly

$$\begin{aligned}\log w_s &= \int_1^s -\partial_t \log \pi_t(\mathbf{x}_t) + R_t(\mathbf{x}_t, \mathbf{x}_t) - \hat{R}_t(\mathbf{x}_t, \mathbf{x}_t)\, \mathrm{d}t \\ &= \int_1^s -\partial_t \log \pi_t(\mathbf{x}_t) + \sum_{\mathbf{y}} R_t(\mathbf{x}_t, \mathbf{y}) \frac{\pi_t(\mathbf{y})}{\pi_t(\mathbf{x}_t)}.\end{aligned}$$

This coincides with the importance weights used in the SMC proposed in (Ou et al., 2025; Holderrieth et al., 2025). In these works, the intermediate target distribution is defined as a geometric interpolation between the base and target distributions: $\pi_t = p_{\text{base}}^t p_{\text{target}}^{1-t}$. The proposal rate matrix R_t is then trained to satisfy the Kolmogorov forward equation. Alternatively, one may use an arbitrary proposal, such as a pretrained rate matrix, but this typically leads to suboptimal performance.

In Lee et al. (2025), the intermediate target is defined as

$$\pi_t(\mathbf{x}_t) = p_t(\mathbf{x}_t)p_t(\zeta|\mathbf{x}_t)^\alpha, \quad p_t(\zeta|\mathbf{x}_t) = \mathbb{E}_{p_t(\mathbf{x}_0|\mathbf{x}_t)}[p(\zeta|\mathbf{x}_0)], \tag{30}$$

where $p(\zeta|\mathbf{x}_0)$ denotes the tilting reward function. To align with the notation used in Lee et al. (2025), we denote the proposal rate matrix by Q_t, the forward rate matrix generating p_t by R_t, and its corresponding backward rate matrix by $\hat{R}_t$. With this notation, the importance weight is given by:

$$\begin{aligned}\log w_t = &\underbrace{\int_1^s -\partial_t \log \pi_t(\mathbf{x}_t) + Q_t(\mathbf{x}_t, \mathbf{x}_t) - \hat{R}_t(\mathbf{x}_t, \mathbf{x}_t)\,\mathrm{d}t}_{①} \\ &+ \underbrace{\sum_{s\le t, \mathbf{x}_{t^+}\neq\mathbf{x}_t} \log \pi_t(\mathbf{x}_t) - \log \pi_t(\mathbf{x}_{t^+}) + \log \hat{R}_t(\mathbf{x}_t, \mathbf{x}_{t^+}) - \log Q_t(\mathbf{x}_{t^+}, \mathbf{x}_t)}_{②}.\end{aligned} \tag{31}$$

To connect with the derivation in Lee et al. (2025), we use the following identities:

$$\hat{R}_t(\mathbf{x}_t, \mathbf{x}_{t^+}) = R_t(\mathbf{x}_{t^+}, \mathbf{x}_t)\frac{p_t(\mathbf{x}_{t^+})}{p_t(\mathbf{x}_t)}. \tag{32}$$

$$\hat{R}_t(\mathbf{x}_t, \mathbf{x}_t) = -\sum_{y\neq\mathbf{x}_t} \hat{R}_t(\mathbf{x}_t, \mathbf{y}) = -\sum_{\mathbf{y}\neq\mathbf{x}_t} R(\mathbf{y}, \mathbf{x}_t)\frac{p_t(\mathbf{y})}{p_t(\mathbf{x})}. \tag{33}$$

$$\partial_t p_t(\mathbf{x}_t) = \sum_{\mathbf{y}} R_t(\mathbf{y}, \mathbf{x}_t)p_t(\mathbf{y}). \tag{34}$$

$$\partial_t \log p_t(\mathbf{x}_t) = \frac{1}{p_t(\mathbf{x}_t)}\sum_{\mathbf{y}} R_t(\mathbf{y}, \mathbf{x}_t)p_t(\mathbf{y}) = \sum_{\mathbf{y}\neq\mathbf{x}} R_t(\mathbf{y}, \mathbf{x}_t)\frac{p_t(\mathbf{y})}{p_t(\mathbf{x}_t)} + R_t(\mathbf{x}_t, \mathbf{x}_t). \tag{35}$$

$$\begin{aligned}\partial_t \log p_t(\zeta|\mathbf{x}_t) = \frac{\partial_t p_t(\zeta|\mathbf{x}_t)}{p_t(\zeta|\mathbf{x}_t)} &= -\frac{1}{p_t(\zeta|\mathbf{x}_t)}\sum_{\mathbf{y}} R_t(\mathbf{x}_t, \mathbf{y})p_t(\zeta|\mathbf{y}) \\ &= -\sum_{\mathbf{y}\neq\mathbf{x}_t} R_t(\mathbf{x}_t, \mathbf{y})\frac{p_t(\zeta|\mathbf{y})}{p_t(\zeta|\mathbf{x}_t)} - R_t(\mathbf{x}_t, \mathbf{x}_t) \\ &\triangleq R_t^{\alpha=1}(\mathbf{x}_t, \mathbf{x}_t) - R_t(\mathbf{x}_t, \mathbf{x}_t),\end{aligned} \tag{36}$$

where we denote $R_t^{\alpha=1}(\mathbf{x}_t, \mathbf{x}_t) = R_t(\mathbf{x}_t, \mathbf{y})\frac{p_t(\zeta|\mathbf{y})}{p_t(\zeta|\mathbf{x}_t)}$, which induces the identity:

$$\log p_t(\zeta|\mathbf{y}) - \log p_t(\zeta|\mathbf{x}_t) = \log R_t^{\alpha=1}(\mathbf{x}_t, \mathbf{x}_t) - \log R_t(\mathbf{x}_t, \mathbf{y}).$$

Substituting into ①, we obtain:

$$\begin{aligned}① &= \int_1^s -\left(\alpha\partial_t \log p_t(\zeta|\mathbf{x}_t) + \sum_{\mathbf{y}\neq\mathbf{x}} R_t(\mathbf{y}, \mathbf{x}_t)\frac{p_t(\mathbf{y})}{p_t(\mathbf{x}_t)} + R_t(\mathbf{x}_t, \mathbf{x}_t)\right) + Q_t(\mathbf{x}_t, \mathbf{x}_t) + \sum_{\mathbf{y}\neq\mathbf{x}_t} R(\mathbf{y}, \mathbf{x}_t)\frac{p_t(\mathbf{y})}{p_t(\mathbf{x})}\,\mathrm{d}t \\ &= \int_1^s -\alpha\partial_t \log p_t(\zeta|\mathbf{x}_t) - R_t(\mathbf{x}_t, \mathbf{x}_t) + Q_t(\mathbf{x}_t, \mathbf{x}_t)\,\mathrm{d}t \\ &= \int_1^s \alpha\left(R_t(\mathbf{x}_t, \mathbf{x}_t) - R_t^{\alpha=1}(\mathbf{x}_t, \mathbf{x}_t)\right) - R_t(\mathbf{x}_t, \mathbf{x}_t) + Q_t(\mathbf{x}_t, \mathbf{x}_t)\,\mathrm{d}t;\end{aligned}$$

Algorithm 1 Twisted SMC for Reward-aligned Discrete Diffusion Sampling

Input: Pre-trained discrete diffusion model p_θ, Prior distribution $\boldsymbol{\pi}$, Proposal $\mathcal{F}$, Reward function $r(\cdot)$, KL weight α, Number of particles N, Number of discrete time steps T, Temperature schedule $0 = \lambda_1 \leq \cdots \leq \lambda_0 = 1$, Minimum ESS threshold ESS_{min}, Resampling scheme RESAMPLE
Output: Weighted particle set $\{X^{(i),1:L}, W^{(i)}\}_{i=1}^N$ approximating p_{tar}

for $i = 1 \dots N$ **do**
  $Z_1^{(i)l} \sim \text{Cat}(\mathbf{z}; \boldsymbol{\pi})$ for $l = 1 \dots L$ {Initialize particles according to prior $\boldsymbol{\pi}$ at $t = 1$}
  $W_1^{(i)} \leftarrow 1/N$
end for
for $\tau = T \dots 1$ **do**
  $t \leftarrow \tau/T$
  $s \leftarrow (\tau - 1)/T$
  for $i = 1 \dots N$ **do**
    $Z_s^{(i),1:L} \sim \mathcal{F}_s(\mathbf{z}_s^{1:L} | Z_t^{(i),1:L})$
    $W_s^{(i)} \leftarrow w_s(Z_s^{(i),1:L})$ {Using Eq. (9)}
  end for
  $W_s^{(i)} \leftarrow W_s^{(i)} / \sum_{j=1}^N W_s^{(j)}$ for $i = 1 \dots N$ {Normalize weights}
  $\text{ESS} \leftarrow \left(\sum_{i=1}^N (W_s^{(i)})^2\right)^{-1}$
  if $\text{ESS} < \text{ESS}_{\text{min}}$ **then**
    $\{Z_s^{(i),1:L}, W_s^{(i)}\}_{i=1}^N \leftarrow \text{RESAMPLE}(\{Z_s^{(i),1:L}, W_s^{(i)}\}_{i=1}^N)$
  end if
end for
$\{X^{(i),1:L}, W^{(i)}\}_{i=1}^N \leftarrow \{Z_0^{(i),1:L}, W_0^{(i)}\}_{i=1}^N$
return $\{X^{(i),1:L}, W^{(i)}\}_{i=1}^N$

For ②, we similarly have:

$$\begin{aligned}
② &= \sum_{s \leq t, \mathbf{x}_{t^+} \neq \mathbf{x}_t} \alpha \left(\log p_t(\zeta|\mathbf{x}_t) - \log p_t(\zeta|\mathbf{x}_{t^+})\right) + \log R_t(\mathbf{x}_{t^+}, \mathbf{x}_t) - \log Q_t(\mathbf{x}_{t^+}, \mathbf{x}_t) \\
&= \sum_{s \leq t, \mathbf{x}_{t^+} \neq \mathbf{x}_t} \alpha \left(\log R_t^{\alpha=1}(\mathbf{x}_{t^+}|\mathbf{x}_t) - \log R_t(\mathbf{x}_{t^+}|\mathbf{x}_{t^+})\right) + \log R_t(\mathbf{x}_{t^+}, \mathbf{x}_t) - \log Q_t(\mathbf{x}_{t^+}, \mathbf{x}_t).
\end{aligned}$$

Combining both terms, the full expression for the importance weight becomes:

$$\begin{aligned}
\log w_s &= ① + ② \\
&= \int_1^s Q_t(\mathbf{x}_t, \mathbf{x}_t) - R_t(\mathbf{x}_t, \mathbf{x}_t)\, \mathrm{d}t + \sum_{s \leq t, \mathbf{x}_{t^+} \neq \mathbf{x}_t} \log R_t(\mathbf{x}_{t^+}, \mathbf{x}_t) - \log Q_t(\mathbf{x}_{t^+}, \mathbf{x}_t) \\
&\quad + \int_1^s \alpha \left(R_t(\mathbf{x}_t, \mathbf{x}_t) - R_t^{\alpha=1}(\mathbf{x}_t, \mathbf{x}_t)\right) \mathrm{d}t + \sum_{s \leq t, \mathbf{x}_{t^+} \neq \mathbf{x}_t} \alpha \left(\log R_t^{\alpha=1}(\mathbf{x}_{t^+}|\mathbf{x}_t) - \log R_t(\mathbf{x}_{t^+}|\mathbf{x}_{t^+})\right),
\end{aligned}$$

which recovers the expression for the importance weight in (Lee et al., 2025, Theorem 2).

C. Sampling Algorithm

The final algorithm for reward aligned sampling for discrete diffusion models using twisted SMC is detailed in Alg. 1. For the proposal $\mathcal{F}_s$, we can choose any one of proposals from Sec. 3. For the resampling scheme, one can use stratified or systematic multinomial sampling, and do either a full or a partial resample.

D. Gradient analysis of the Denoising model

In Sec. 3.2, we proposed a continuous formulation (Eq. (16)) of the denoising model $\mathbf{x}_\theta$ to address the discontinuity induced by the *Carry-Over Unmasking* constraint in masked diffusion models. In this section, we analyse the resulting gradients,

providing motivation for the proposed formulation.

To do so, we first provide a slightly modified version of Eq. (16):

$$\mathbf{x}_\theta(\mathbf{z}_t^{1:L}, t)^l = \gamma \tilde{\mathbf{x}}_\theta(\tilde{\mathbf{z}}_t^{1:L}, t)^l + (\mathbf{1} - \boldsymbol{m}) \odot \mathbf{z}_t^l \tag{37}$$

where $\tilde{\mathbf{z}}_t^{1:L} := \left[\mathbf{z}_t^{1:l-1}, \boldsymbol{m}, \mathbf{z}_t^{l+1,L}\right]$. Notice that this is also a completely valid continuous formulation of the *Carry-Over Unmasking* constraint. However, in this case, $\tilde{\mathbf{x}}_\theta(\tilde{\mathbf{z}}_t^{1:L}, t)^l$ is not dependent on $\mathbf{z}_t^l$, as it is not part of the input.

Let $m \in \{1, \dots, V\}$ denote the index of the mask category. We can write Eq. (37) in the index notation as,

$$\mathbf{x}_\theta(\mathbf{z}_t^{1:L}, t)_i^l = \gamma \tilde{\mathbf{x}}_\theta(\tilde{\mathbf{z}}_t^{1:L}, t)_i^l + \mathbf{z}_{t,i}^l \cdot \mathbf{1}[i \neq m]. \tag{38}$$

Similarly, γ can be written as,

$$\gamma = 1 - \sum_{k \neq m} \mathbf{z}_{t,k}^l.$$

We want to compute,

$$\frac{\partial \mathbf{x}_\theta(\mathbf{z}_t^{1:L}, t)_i^l}{\partial \mathbf{z}_{t,j}^{l'}}$$

for $l, l' \in \{1, \dots, L\}$ and $i, j \in \{1, \dots, V\}$. First, note that both terms in Eq. (38) are always zero for $i = m$, since $\tilde{\mathbf{x}}_\theta(\tilde{\mathbf{z}}_t^{1:L}, t)_m^l = 0$ by the *Zero Masking Probabilities* constraint. Therefore, $\frac{\partial \mathbf{x}_\theta(\mathbf{z}_t^{1:L}, t)_i^l}{\partial \mathbf{z}_{t,j}^{l'}} = 0$ when $i = m$, and we restrict our analysis to $i \neq m$ from here on. Second, in masked diffusion models, we are only interested in gradients with respect to masked tokens, i.e., $\mathbf{z}_t^{l'} = \boldsymbol{m}$, because for unmasked tokens the factorized proposal in Eq. (18) is deterministic and independent of the gradient. Thus, we will only consider cases where $\mathbf{z}_t^{l'} = \boldsymbol{m}$ in the following analysis.

Now, we can look at cases $l' = l$ and $l' \neq l$ separately.

Case 1: $l' = l$

We have, $\frac{\partial \gamma}{\partial \mathbf{z}_{t,j}^{l'}} = \begin{cases} -1 & j \neq m \\ 0 & j = m \end{cases}$, and $\frac{\partial \tilde{\mathbf{x}}_\theta(\tilde{\mathbf{z}}_t^{1:L}, t)_i^l}{\partial \mathbf{z}_{t,j}^{l'}} = 0$ as $\tilde{\mathbf{x}}_\theta(\tilde{\mathbf{z}}_t^{1:L}, t)^l$ is not dependent on $\mathbf{z}_t^{l'} = \mathbf{z}_t^l$ as noted earlier. The final gradients are given as,

$$\frac{\partial \mathbf{x}_\theta(\mathbf{z}_t^{1:L}, t)_i^l}{\partial \mathbf{z}_{t,j}^{l'}} = \begin{cases} \begin{cases} -\tilde{\mathbf{x}}_\theta(\tilde{\mathbf{z}}_t^{1:L}, t)_i^l + 1 & j = i \\ -\tilde{\mathbf{x}}_\theta(\tilde{\mathbf{z}}_t^{1:L}, t)_i^l & j \neq i \end{cases} & j \neq m \\ 0 & j = m. \end{cases}$$

We can see that these gradients are exactly equal to the finite difference around $\mathbf{z}_t^l = \boldsymbol{m}$, i.e.,

$$\frac{\partial \mathbf{x}_\theta(\mathbf{z}_t^{1:L}, t)_i^l}{\partial \mathbf{z}_{t,j}^{l'}} = \mathbf{x}_\theta(\left[\mathbf{z}_t^{1:l-1}, \mathbf{e}_j, \mathbf{z}_t^{l+1,L}\right], t)_i^l - \mathbf{x}_\theta(\left[\mathbf{z}_t^{1:l-1}, \boldsymbol{m}, \mathbf{z}_t^{l+1,L}\right], t)_i^l$$

where $\mathbf{e}_j$ is the one hot vector corresponding to a non-mask category j. Thus, they provide accurate values when using a first order approximation.

Case 2: $l' \neq l$

The gradients in this case are given simply as,

$$\frac{\partial \mathbf{x}_\theta(\mathbf{z}_t^{1:L}, t)_i^l}{\partial \mathbf{z}_{t,j}^{l'}} = \gamma \frac{\partial \tilde{\mathbf{x}}_\theta(\tilde{\mathbf{z}}_t^{1:L}, t)_i^l}{\partial \mathbf{z}_{t,j}^{l'}}.$$

$\mathbf{x}_\theta(\mathbf{z}_t^{1:L}, t)_i^l$ has a gradient with respect to $\mathbf{z}_t^{l'}$, however it is gated by γ. This makes sense because if $\mathbf{z}_t^l$ is unmasked, i.e., $\gamma = 0$, then $\mathbf{x}_\theta(\mathbf{z}_t^{1:L}, t)^l = \mathbf{z}_t^l$ and it is not dependent on the value of any other token $\mathbf{z}_t^{l'}$.

We note that throughout this paper, we have used the formulation provided in Eq. (16), not the modified version in Eq. (37). The original formulation has similar but slightly different gradients - notably, the gradients for the original formulation include an additional term for $l' = l$. We used the gradients of the modified version for our analysis as they are simple and provide intuition for choosing this specific type of continuous formulation.

E. Related Work

SMC. SMC has been proposed for guidance or alignment of diffusion models by Wu et al. (2023); Dou & Song (2024); Cardoso et al. (2023); Phillips et al. (2024); Kim et al. (2025); Singhal et al. (2025); Skreta et al. (2025); He et al. (2025). In the context of discrete diffusion models, Lee et al. (2025) use SMC to sample from the unnormalized tempered distribution $p_0(x_0)p_0(\zeta|x_0)^\alpha$, where ζ is a conditioning variable and α is the guidance scale. They show that guided rate matrix R_t^α normally used for guidance based sampling is equal to the true tempered rate matrix $R_t^{\alpha,\text{true}}$ only for $\alpha = 1$. Thus, to sample from $p_0(x_0)p_0(\zeta|x_0)^\alpha$ where $\alpha \neq 1$, they propose to decouple the rate matrix used as R_t^β from α. SMC is used to sample from the target distribution $p_0(x_0)p_0(\zeta|x_0)^\alpha$, while the distributions induced by the rate matrix R_t^β serve as the proposal. Usually, β is set close to 1. In contrast to our method, their method necessitates additional learning or fine-tuning to first obtain the rate guided rate matrix $R_t^{\alpha=1}$ which is needed for the proposal. Additionally, the proposal used is not necessarily optimal to minimize weight variance. As α and β diverge, SMC resampling will be doing most of the heavy-lifting (instead of the guidance) and this may lead to high resampling frequency and low sample diversity.

Classifier-free and Classifier-based Guidance. Classifier-based (Dhariwal & Nichol, 2021; Song et al., 2020) and classifier-free (Ho & Salimans, 2022; Zhang et al., 2023; Yuan et al., 2023) guidance has been widely used for continuous diffusion models. Li et al. (2022); Lovelace et al. (2023) perform classfier-based and classfier-free guidance using diffusion models on the continuous latent representations of discrete data. Schiff et al. (2024) propose classifier-free (D-CFG) and classifier-based guidance (D-CBG) for discrete diffusion models. For D-CBG, they propose to use the first order Taylor approximation of the classifier output logits $\log p_\phi$ to make sampling from the classifier guided backward process tractable for datasets with large vocabularies or high number of dimensions. Nisonoff et al. (2024) derive classifier-based and classifier-free guidance for CTMC-based discrete diffusion models. They use a similar Taylor approximation to estimate the predictor-guided rate matrices. Vignac et al. (2022) propose DiGress - a discrete diffusion model for graphs, where they also use a Taylor approximation for regressor guidance. Ninniri et al. (2024) propose classifier-free guidance for DiGress.

Reinforcement Learning. GLID^{2}E (Cao et al., 2025) and SEPO (Zekri & Boullé, 2025) extend reinforcement learning based fine tuning methods (Black et al., 2023; Uehara et al., 2024) to discrete diffusion models. Wang et al. (2024) aim to optimize the same objective presented in Eq. (6). They fine-tune the parametrized generator Q^θ of the pre-trained discrete diffusion model by back-propagating through the rewards. To enable differentiability during sampling at each step of the CTMC, they use the Gumbel-Softmax trick. DPO has been used to align continuous diffusion models (Wallace et al., 2024) to human preferences. D3PO (Borso et al., 2025) extends DPO to discrete diffusion models.

F. Experimental Setting and Additional Results

F.1. Baselines

Oracle. For the mixture of Gaussian toy datasets, we use twisted SMC with the locally optimal proposal presented in Sec. 3.1 as the oracle. The locally optimal proposal is calculated using a brute force approach for the entire 64^2-sized state space at each time step.

Approximate guidance. We intend to compare the proposed tempered SMC method with approximate guidance. For approximate guidance, we need to sample using

$$p_\theta^{\text{guidance}}(\mathbf{z}_s^{1:L}|\mathbf{z}_t^{1:L}) \propto p_\theta(\mathbf{z}_s^{1:L}|\mathbf{z}_t^{1:L})\exp\left(\frac{\hat{r}(\mathbf{z}_s^{1:L})}{\alpha}\right)$$

where $\frac{1}{\alpha}$ acts as the guidance strength. Since sampling from the above is intractable for large vocabulary sizes or high-dimensional data, we can use the first order Taylor approximation from Eq. (12) again, to rewrite $p_\theta^{\text{guidance}}(\mathbf{z}_s^{1:L}|\mathbf{z}_t^{1:L})$ as

proportional to:

$$p_\theta(\mathbf{z}_s^{1:L}|\mathbf{z}_t^{1:L})\exp\left(\frac{1}{\alpha}\sum_{l=1}^{L}\langle\nabla_{\mathbf{z}_t^l}\hat{r}(\mathbf{z}_t^{1:L}),\mathbf{z}_s^l\rangle\right) \tag{39}$$

This process becomes very similar to using the first order Taylor approximation of the locally optimal proposal, except that there is no tempering and no resampling.

F.2. Mixture of Gaussian datasets

For all experiments on these datasets, we use 2000 particles and 100 discretized time steps. The reward function computation details are provided in the Appendix F.2.1. For resampling, we have used ESS threshold $\text{ESS}_{\text{min}} = 1000$, and we perform a full resample using systematic multinomial sampling. The KL weight α is 1, and the temperature schedule is set as $\lambda_t = 1 - t$. For the diffusion model, we use a linear noise schedule $\alpha_t = 1 - t$.

F.2.1. Reward computation

The rewards specified in Fig. 1 are computed in a differentiable manner for one hot inputs $\mathbf{x}, \mathbf{y}$ corresponding to X, Y as follows. For the top row:

$$r(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{V}\left(-\mathbf{x}_i \cdot \frac{\hat{i}^2}{100} - \mathbf{y}_i \cdot \hat{i}^2\right)$$

and, similarly for the bottom row:

$$r(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{V}\left(-\mathbf{x}_i \cdot \hat{i}^2 - \mathbf{y}_i \cdot \frac{(\hat{i}-1)^2}{10}\right)$$

where $\hat{i} = 12(i/63 - 1/2)$.

F.2.2. Reward and ESS traces for different proposals

In Fig. 3, we visualize traces of the mean (estimated) reward and ESS of the particle set when using tempered SMC with different proposals on the Gaussian mixture datasets from Sec. 4.1.

The first order approximation of the locally optimal proposal required fewer resampling steps compared to using the reverse as proposal for the first dataset, but for the second dataset, the number of resampling steps needed is the same for both. In both datasets, however, using the locally optimal proposal produces significantly fewer resamples.

It is also worth noting that the reverse process relies completely on the SMC resampling steps to increase the rewards for the particles. This is expected as the reverse process proposal in Eq. (10) is completely independent of the reward function. In contrast, using both the locally optimal proposal and its first order approximation results in a steady increase in rewards as the particles are iteratively sampled using reward-guided proposals (Eq. (11), Eq. (17)).

F.3. Binarized MNIST dataset

Applying the twisted SMC-based guidance to masked diffusion models trained on the binarised MNIST dataset was tricky because the reward-guided proposals become ineffective once a pixel is unmasked. This necessitated careful selections of tempering schedule, KL weight, ESS threshold, etc. to get good results. To test this theory, we redid our experiments using ReMDM and UDLM, both of which allow injecting guidance throughout the entire sampling process.

For all experiments on this dataset, we use 20 particles and 100 discretized time steps. The tempering schedules used are visualised in Fig. 6. For resampling, we have used ESS threshold $\text{ESS}_{\text{min}} = 15$, and we perform partial resampling to resample only half of the particles. For all the diffusion models, we use a linear noise schedule $\alpha_t = 1 - t$.

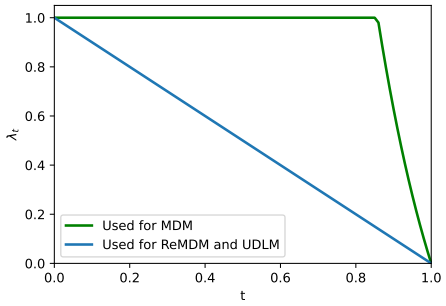

Figure 6. Tempering schedules used for twisted SMC on the binarised MNIST dataset.

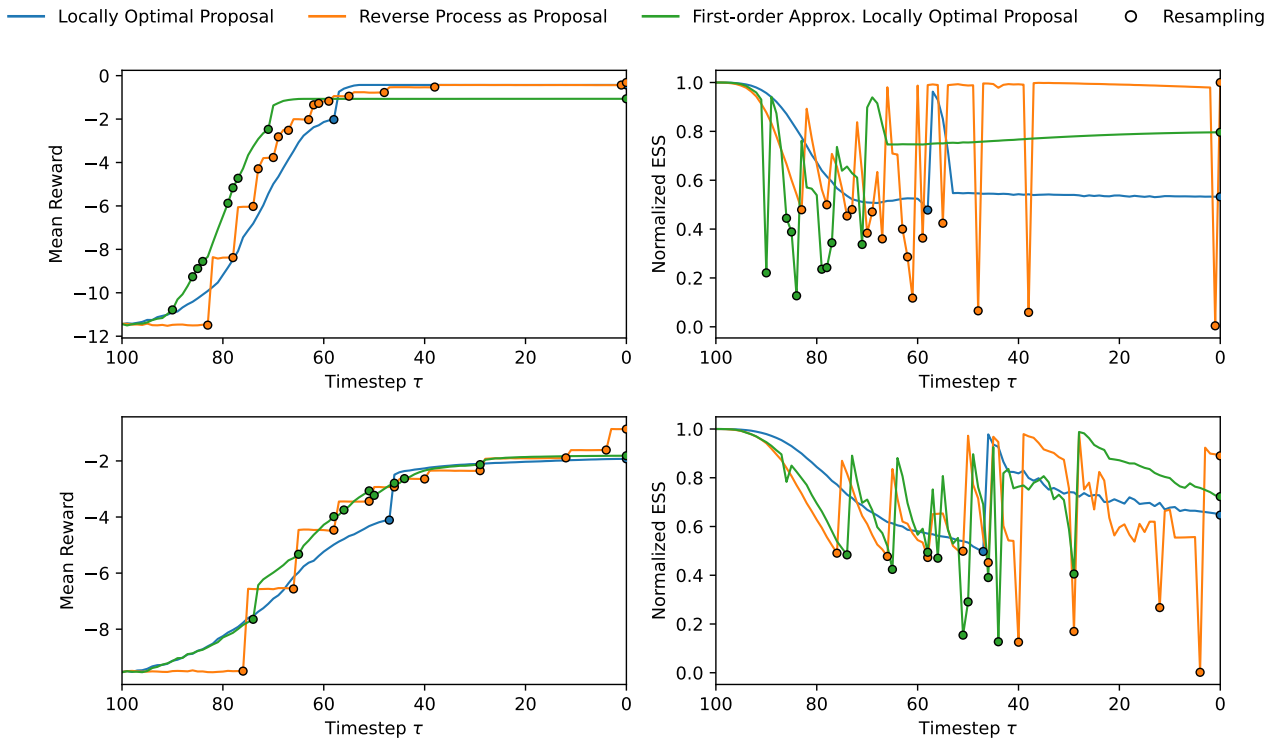

Figure 3. Mean reward and normalized effective sample size (ESS) of the particle sets across timesteps using tempered SMC with different proposals. The first and second rows correspond to the mixture of Gaussians datasets shown in the first and second rows of Fig. 1, respectively.

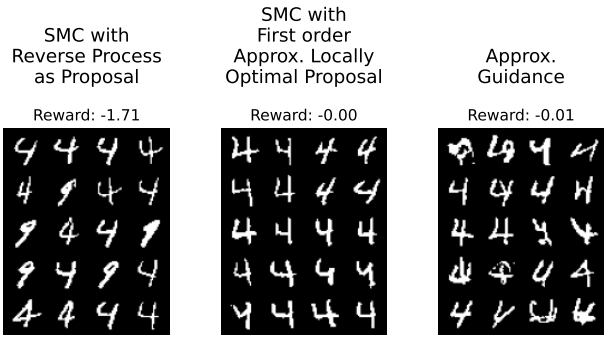

Figure 4. Binarised MNIST samples generated for $y_{\text{target}} = 4$, $\alpha = 1$ using ReMDM with tempered SMC (left and middle), and approximate guidance (right).

Figure 5. Binarised MNIST samples generated for $y_{\text{target}} = 4$, $\alpha = 1$ using UDLM with tempered SMC (left and middle), and approximate guidance (right).

F.3.1. Using ReMDM

Setup. We reuse the masked diffusion model trained in Sec. 4.2, only replacing the reverse process parametrisation to that of ReMDM for inference. For the remasking schedule, we use the max-capped schedule (Wang et al., 2025), with $\eta_{\text{cap}} = 0.1$. We use the same reward function. For the tempered SMC, we change the tempering schedule to a simple linear schedule, $\lambda_t = 1 - t$.

Result. Fig. 4 shows the resulting particles. The benefit of allowing remasking is clearly evident when comparing the samples generated using approximate guidance in Fig. 4 with those in Fig. 2. The amount of corrupted samples have decreased significantly. We observed that, SMC with ReMDM is much more flexible compared to masked diffusion in terms of the choice of tempering schedule, and other parameters.

F.3.2. Using UDLM

Setup. We train a UDLM on the binarised MNIST dataset from Sec. 4.2. For the denoising model, we use a U-Net with attention applied at lower resolutions, similar to (Ho et al., 2020). We use a linear tempering schedule $\lambda_t = 1 - t$ for SMC.

Result. The resulting samples are shown in Fig. 5.

G. Limitations and Future Work

While our method shows good results on small datasets, its effectiveness on more complex tasks, such as math reasoning, protein design, remains to be evaluated. Additionally, we have yet to benchmark our approach against a broader set of reinforcement learning and guidance methods, as outlined in Appendix E. The temperature schedule λ_t used for intermediate target distributions is presently hand-crafted; future work will explore automating this component through adaptive tempering techniques.