Uniform Diffusion Models Revisited: Leave-One-Out Denoiser and Absorbing State Reformulation

Samson Gourevitch[*,1] **Yazid Janati**[*,2,4] **Dario Shariatian**[3]
Umut Simsekli[3] **Eric Moulines**[4,5] **Eric P. Xing**[2,4] **Alain Durmus**[1]
[1] CMAP, Ecole polytechnique [2] Institute of Foundation Models
[3] Inria, PSL Research University [4] MBZUAI [5] EPITA, LRE

Abstract

Discrete diffusion models are often trained through clean-data prediction, but the prediction can be used in different ways to define the reverse dynamics. In Masked Diffusion Models (MDM) these choices largely coincide, whereas in Uniform Diffusion Models (UDM) they do not. We show that the standard plug-in bridge parameterization for UDM is not optimized by the denoising posterior, but by a *leave-one-out* posterior that predicts each clean token without using its own noisy observation. This identifies a mismatch between the plug-in ELBO and the usual cross-entropy denoising objective. We characterize the leave-one-out target and derive exact conversions between the denoiser, the leave-one-out posterior, and the score. These conversions allow us to disentangle parameterization and training objective. Our results also lead to inference improvements without any additional training through an informed predictor-corrector sampler and improved temperature sampling based on the leave-one-out predictor. We further introduce an absorbing-state reformulation of uniform diffusion that preserves the UDM joint law while decomposing it into masked-diffusion-like sampling operations, with simpler denoising posteriors, carry-over unmasking, and a natural remasking mechanism. On language modeling and code generation, leave-one-out parameterizations consistently improve UDM generation, while the absorbing construction matches or surpasses masked diffusion. These results suggest that the empirical gap between masked and uniform diffusion is driven less by the choice of marginals themselves than by parameterization and sampling design. The code and models can be found at https://github.com/samsongourevitch/rev_udm.

1 Introduction

Discrete diffusion models have emerged as a strong alternative to autoregressive generation for language and other structured discrete domains [10, 15, 22]. Unlike left-to-right models, they support parallel token updates, controllable infilling, and a flexible trade-off between compute and sample quality at inference time. Recent models show that they can scale competitively on modern text benchmarks [18, 21, 25, 28]. Although early work emphasized Uniform Diffusion Models (UDM), in which each position is replaced by a uniformly sampled token [1, 13, 30], more recent approaches have relied on Masked Diffusion Models (MDM), where corruption replaces tokens by a separate

* Equal contribution
Corresponding authors: samson.gourevitch@polytechnique.edu, yazid.janati@mbzuai.ac.ae and dario.shariatian@inria.fr

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

mask symbol, motivated by their better perplexity at large vocabulary size [18, 28, 29]. This empirical shift left open fundamental questions regarding parameterization choices in UDMs. To make this question precise, we distinguish three common parameterizations: the network prediction can be used as (i) a score parameterizing a rate matrix [18, 26, 34], (ii) a denoiser, where the network assigns probabilities to possible clean $\mathbf{x}_0$, and the reverse transition is computed as the mixture of the $\mathbf{x}_0$-conditioned reverse transitions $q_{s|0,t}(\cdot|\mathbf{x}_0, \mathbf{x}_t)$, weighted by these probabilities [1, 11], or (iii) a direct clean-token proxy $\hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t)$, plugged instead of $\mathbf{x}_0$ in $q_{s|0,t}(\cdot|\mathbf{x}_0, \mathbf{x}_t)$ [13, 28, 30, 36].

Contributions. In this paper, we first show that for uniform diffusion the plug-in bridge parameterization, the third option above, does not reach its optimum for the standard ELBO loss at the denoising posterior, which has been the central quantity in diffusion models. We show that for this plug-in parameterization, the ELBO is instead minimized by the leave-one-out posterior: to predict the clean token at position ℓ, the model uses the noisy tokens at all positions except ℓ. This distinction is mostly hidden in masked diffusion because the parameterizations (ii) and (iii) coincide.

This observation exposes a mismatch between two standard objectives for UDMs: the plug-in ELBO targets a leave-one-out posterior, whereas the usual cross-entropy objective targets the plain denoising posterior. We use the conversion formulas between these two quantities to train either parameterization with either objective, namely a denoiser through the ELBO or a leave-one-out predictor through cross-entropy. Empirically, across objectives and samplers, we find that parameterizing the leave-one-out posterior consistently improves generative performance. These conversions are also useful at inference time. They turn a trained denoiser into the leave-one-out quantities needed for an informed predictor-corrector sampler, in the spirit of [40], without training an auxiliary model. They also indicate where to apply sampling heuristics: applying top-p or temperature to the leave-one-out predictor, rather than to the denoiser itself, improves generation at no additional training cost.

We finally revisit the gap between uniform and masked diffusion processes [28, 35, 36]. We ask whether this gap is caused by the difference in marginals themselves, or by the parameterization and sampling scheme usually paired with each process. To separate these effects, we propose two absorbing-state constructions of uniform diffusion that preserves the UDM joint law while inheriting structural simplifications of MDM. We coin these Absorbing State Uniform Diffusion Model (AUDM) and Masked Uniform Diffusion Model (MUDM). As detailed in Appendix C, MUDM reuses an MDM denoiser while recovering the UDM joint law. As for AUDM, it reaches perplexity and generation quality comparable to masked diffusion. The empirical advantage of masked diffusion therefore appears to stem not from masked versus uniform marginals, but from parameterization choices that can be transferred to a model with UDM marginals.

We validate our theoretical insights and methods through extensive experiments on large scale language modeling, Sudoku, and mathematical reasoning with TinyGSM. The leave-one-out parameterization improves over its denoiser counterpart, while our absorbing construction is competitive with masked diffusion.

Notation. We consider a discrete space $\mathsf{X} = \mathsf{V}^L$ with $\mathsf{V} = [\![1, K]\!]$, and write $\Delta_K = \{p \in \mathbb{R}^K : p_i \geq 0,\ \sum_i p_i = 1\}$ for the probability simplex. Tokens are identified with their one-hot vectors in $\mathbb{R}^K$ via the canonical basis $(e_1, \dots, e_K)$, so categorical distributions are points in Δ_K (or Δ_K^L in the multi-token case). For masked diffusion, the mask token is denoted m. We write $\mathbf{x} \in \mathsf{X}$, $x \in \mathsf{V}$, $\mathbf{x}^\ell$ for the ℓ-th token of $\mathbf{x}$, and $\mathbf{x}^{-\ell} \in \mathsf{V}^{L-1}$ for $\mathbf{x}$ with the ℓ-th token removed. For $\pi \in \Delta_K$, we denote by $\mathrm{Cat}(\pi)$ the categorical distribution with parameter π. For ease of notation, we also denote by $x \mapsto \mathrm{Cat}(x; \pi) = \langle x, \pi \rangle$ the associated density with respect to the counting measure.

2 Preliminaries on discrete diffusion models

We first briefly introduce discrete diffusion models [1, 13, 28, 30, 31]. Denoting by p_0 a data distribution on X, this class of models consists in transporting a reference distribution $p_1(\cdot) \coloneqq \prod_{\ell=1}^L \mathrm{Cat}(\cdot; \boldsymbol{\pi}^\ell)$, with $\boldsymbol{\pi} \in (\Delta_K)^L$, to the data distribution p_0 through successive transitions. We consider a Markov process $(X_t)_{t\in[0,1]}$ started at p_0 with transitions $q_{t|s}(\mathbf{x}_t|\mathbf{x}_s) = \prod_{\ell=1}^L q_{t|s}^\ell(\mathbf{x}_t^\ell|\mathbf{x}_s^\ell)$, and

$$q_{t|s}^\ell(\mathbf{x}_t^\ell|\mathbf{x}_s^\ell) = \mathrm{Cat}\big(\mathbf{x}_t^\ell; \alpha_{t|s}\mathbf{x}_s^\ell + (1-\alpha_{t|s})\boldsymbol{\pi}^\ell\big)\ , \quad 0 \leq s < t \leq 1\ . \tag{1}$$

Here $(\alpha_t)_{t\in[0,1]}$ is a monotone noise schedule with $\alpha_0 = 1$, $\alpha_1 \approx 0$, and $\alpha_{t|s} \coloneqq \alpha_t/\alpha_s$. We denote by p_t the marginal of X_t. By reversing the Markov process in time, we obtain a generative model for p_0 with source distribution p_1 and transitions given by

$$p_{s|t}(\mathbf{x}_s|\mathbf{x}_t) = \textstyle\sum_{\mathbf{x}_0} q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{x}_t)\, p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)\ , \tag{2}$$

with $p_{0|t}(\cdot|\mathbf{x}_t)$ the denoising posterior and $q_{s|0,t}$ being the conditional distribution of X_s given (X_0, X_t). We refer to this quantity, which is central to the paper, as the *bridge*. Note that for $s < t$, $q_{t|s}$ and $q_{s|0,t}$ factorize across tokens, *e.g.*, $q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{x}_t) = \prod_{\ell=1}^{L} q^{\ell}_{s|0,t}(\mathbf{x}^{\ell}_s|\mathbf{x}^{\ell}_0, \mathbf{x}^{\ell}_t)$ with

$$q^{\ell}_{s|0,t}(\cdot|x_0, x_t) = \mathrm{Cat}\left(\cdot;\ \frac{\left[\alpha_{t|s}x_t + (1-\alpha_{t|s})\langle x_t, \boldsymbol{\pi}^{\ell}\rangle \mathbf{1}_K\right] \odot \left[\alpha_s x_0 + (1-\alpha_s)\boldsymbol{\pi}^{\ell}\right]}{\alpha_t\langle x_t, x_0\rangle + (1-\alpha_t)\langle x_t, \boldsymbol{\pi}^{\ell}\rangle}\right)\ , \tag{3}$$

when $q^{\ell}_{t|0}(x_t|x_0) > 0$. To define a practical model, we approximate the reverse transitions. To this end, we consider the following joint law with factorized transitions parameterized by θ:

$$p^{\theta}_{0:n}(\mathbf{x}_{t_{0:n}}) = p_1(\mathbf{x}_{t_n}) \prod_{i=1}^{n} p^{\theta}_{s_i|t_i}(\mathbf{x}_{s_i}|\mathbf{x}_{t_i}) \quad \text{where} \quad p^{\theta}_{s|t}(\mathbf{x}_s|\mathbf{x}_t) \coloneqq \prod_{\ell=1}^{L} p^{\theta,\ell}_{s|t}(\mathbf{x}^{\ell}_s|\mathbf{x}_t)\ , \tag{4}$$

for a choice of timesteps $0 = t_0 < \ldots < t_n = 1$, setting $s_i = t_{i-1}$ for notational convenience. Discrete diffusion models are then trained by minimizing the *expected* NELBO $\mathbb{E}[\mathrm{L}_n(X_0;\theta)]$. For a fixed $\mathbf{x}_0$, $\mathrm{L}_n(\mathbf{x}_0;\theta)$ is given, up to an additive constant independent of θ, by:

$$\mathrm{L}_n(\mathbf{x}_0;\theta) = \mathbb{E}\left[-\log p^{\theta}_{0|t_1}(\mathbf{x}_0|X_{t_1}) + \sum_{i=2}^{n} \mathrm{KL}(q_{s_i|0,t_i}(\cdot|X_0, X_{t_i})\|p^{\theta}_{s_i|t_i}(\cdot|X_{t_i}))\,\middle|\, X_0 = \mathbf{x}_0\right]\ . \tag{5}$$

Uniform and masked diffusion models. In this paper we focus on Masked Diffusion Models (MDM) and Uniform Diffusion Models (UDM). For MDM [28, 31], we take $\boldsymbol{\pi}^{\ell} = \mathsf{m} \coloneqq e_K$ where m denotes the mask token and we assume that any state containing this token has zero data probability, *i.e.*, $p_0(\mathbf{x}) = 0$ whenever $\mathbf{x}^{\ell} = \mathsf{m}$ for some ℓ. For UDM [1, 13, 30], $\boldsymbol{\pi}^{\ell} \coloneqq \mathbf{1}/K$ where $\mathbf{1}$ is the vector with all ones, *i.e.*, the uniform distribution. The MDM bridge is given, for $\ell \in [\![1, L]\!]$, by:

$$q^{\ell}_{s|0,t}(x_s|x_0, x_t) = \begin{cases} \mathrm{Cat}\left(x_s; x_0\right) & \text{if } x_t = x_0 \\ \mathrm{Cat}\left(x_s; \frac{\alpha_s - \alpha_t}{1-\alpha_t} x_0 + \frac{1-\alpha_s}{1-\alpha_t}\mathsf{m}\right) & \text{if } x_t = \mathsf{m} \end{cases} \tag{6}$$

and is undefined for $x_t \notin \{\mathsf{m}, x_0\}$. Regarding UDM, the bridge has the explicit form

$$q^{\ell}_{s|0,t}(x_s|x_0, x_t) = \mathrm{Cat}\left(x_s; \frac{K\alpha_t\langle x_t, x_0\rangle x_t + (\alpha_{t|s} - \alpha_t)x_t + (\alpha_s - \alpha_t)x_0 + D_{s,t}\mathbf{1}_K/K}{K\alpha_t\langle x_t, x_0\rangle + 1 - \alpha_t}\right), \tag{7}$$

with $D_{s,t} \coloneqq (1-\alpha_{t|s})(1-\alpha_s)$.

3 Elucidating the parameterizations in discrete diffusion models

3.1 The different parameterizations of the reverse transitions

A central design choice is how to parameterize the family of reverse conditionals $\{p^{\theta,\ell}_{s|t}\}_{\ell=1}^{L}$. A natural approach is to let it be the direct output of a neural network with input $(\mathbf{x}_t, s, t)$. As observed in [1], this approach does not exploit the inductive bias associated with the forward process. Another option is to use a neural network that takes $(\mathbf{x}_t, t)$ as input and produces a simplex-valued prediction, denoted by $\hat{\mathbf{x}}^{\theta}_0(\mathbf{x}_t, t)$, from which the reverse conditionals are constructed through the bridge. There are two common instantiations of this idea. The *marginalization* parameterization [1, 11] interprets this prediction as the factorized distribution $p^{\theta}_{0|t}(\cdot|\mathbf{x}_t) \coloneqq \prod_{\ell=1}^{L} \mathrm{Cat}(\cdot; \hat{\mathbf{x}}^{\theta}_0(\mathbf{x}_t, t)^{\ell})$ and defines

$$\tilde{p}^{\theta}_{s|t}(\mathbf{x}_s|\mathbf{x}_t) = \textstyle\sum_{\mathbf{x}_0} q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{x}_t)\, p^{\theta}_{0|t}(\mathbf{x}_0|\mathbf{x}_t)\ , \tag{Marginalization}$$

whereas the *bridge plug-in* parameterization [13, 28, 30, 31, 36] uses the same simplex-valued prediction directly inside the bridge (3):

$$\hat{p}^{\theta}_{s|t}(\mathbf{x}_s|\mathbf{x}_t) = q_{s|0,t}(\mathbf{x}_s|\hat{\mathbf{x}}^{\theta}_0(\mathbf{x}_t, t), \mathbf{x}_t)\ . \tag{Bridge Plug-in}$$

Note that these two parameterizations coincide if $\Delta_K \ni \mathbf{x}_0^\ell \mapsto q^\ell_{s|0,t}(\mathbf{x}_s^\ell|\mathbf{x}_0^\ell, \mathbf{x}_t^\ell)$ is affine in $\mathbf{x}_0^\ell$. The support conventions and the extension of the bridge to simplex-valued predictions are specified in Remark 2. A recurring argument [28, 29, 36] used to motivate Bridge Plug-in is that, for a fixed $\mathbf{x}_0$, the minimizer of (5) is given for any t by $\hat{\mathbf{x}}_0^\theta(\cdot, t) \equiv \mathbf{x}_0$. However, this argument only characterizes the minimizer of the conditional objective for a fixed $\mathbf{x}_0$. The actual training objective of discrete diffusion models is instead the expected NELBO given by

$$\mathbb{E}[\mathrm{L}_n(X_0;\theta)] = \sum_{i=1}^{n}\sum_{\ell=1}^{L}\mathbb{E}\big[\mathrm{KL}(p^\ell_{s_i|t_i}(\cdot|X_{t_i})\|p^{\theta,\ell}_{s_i|t_i}(\cdot|X_{t_i}))\big] + \mathrm{Const}\ , \tag{8}$$

with $p^\ell_{s|t}(\mathbf{x}_s^\ell|\mathbf{x}_t) \coloneqq \sum_{\mathbf{x}_s^{-\ell}} p_{s|t}(\mathbf{x}_s|\mathbf{x}_t)$ the ℓ-th token marginal transition. Hence, we can see that the minimum is attained when $p^\theta_{s_i|t_i}(\cdot|\mathbf{x}_{t_i}) = \prod_{\ell=1}^L p^\ell_{s_i|t_i}(\cdot|\mathbf{x}_{t_i})$. Moreover, from (2) and (3) we have: $p^\ell_{s|t}(\mathbf{x}_s^\ell|\mathbf{x}_t) = \sum_{\mathbf{x}_0^\ell} q^\ell_{s|0,t}(\mathbf{x}_s^\ell|\mathbf{x}_0^\ell, \mathbf{x}_t^\ell)\, p^\ell_{0|t}(\mathbf{x}_0^\ell|\mathbf{x}_t)$, we conclude that the *marginalization* parameterization reaches the minimum of the loss by taking $p^\theta_{0|t}(\mathbf{x}_0|\mathbf{x}_t) = \prod_{\ell=1}^L p^\ell_{0|t}(\mathbf{x}_0^\ell|\mathbf{x}_t)$, or equivalently, $\hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t) = \mathbb{E}[X_0|X_t = \mathbf{x}_t]$. By contrast, the same conclusion is more delicate for Bridge Plug-in since the canonical extension of (3), as defined after (7), is not necessarily linear with respect to $\mathbf{x}_0^\ell$ so it cannot simply be expressed as the marginalization of a denoising posterior[1]. For MDM this distinction makes no difference, since (53) is token-wise affine in x_0 and thus $\tilde{p}^\theta_{s|t}(\mathbf{x}_s|\mathbf{x}_t) = q_{s|0,t}(\mathbf{x}_s|\hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t), \mathbf{x}_t)$, so the *marginalization* parameterization is itself the *bridge plug-in* parameterization, and a minimizer is the denoising distribution. For UDM however, the normalization term in (7) makes the bridge nonlinear as a function of x_0, so the equivalence between the two parameterizations does not hold. In that setting, it is no longer obvious which object the Bridge Plug-in parameterization recovers at optimum, nor whether it can still attain the minimum of the expected NELBO. The next subsection answers both questions.

3.2 Leave-one-out parameterization

We first recall the factorized *leave-one-out* (LOO) denoising distribution as introduced in [34, 40]:

$$p^{\mathrm{loo}}_{0|t}(\mathbf{x}_0|\mathbf{x}_t) := \prod_{\ell=1}^{L} p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}_0^\ell|\mathbf{x}_t^{-\ell}) \quad \text{where} \quad p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}_0^\ell|\mathbf{x}_t^{-\ell}) \propto \sum_{\mathbf{x}_0^{-\ell}} p_0(\mathbf{x}_0)\prod_{j\neq\ell}^{L} q^j_{t|0}(\mathbf{x}_t^j|\mathbf{x}_0^j)\ . \tag{9}$$

Equivalently, $p^{\mathrm{loo}}_{0|t}(\cdot|\mathbf{x}_t) = \prod_{\ell=1}^L \mathrm{Cat}(\hat{\mathbf{x}}^{\mathrm{loo}}_{0|t}(\mathbf{x}_t)^\ell)$ with $\hat{\mathbf{x}}^{\mathrm{loo}}_{0|t}(\mathbf{x}_t)^\ell = \mathbb{E}[X_0^\ell|X_t^{-\ell} = \mathbf{x}_t^{-\ell}]$. This quantity is relevant for the Bridge Plug-in parameterization since it allows to express the minimizer of the expected NELBO under such parameterization, as made precise in the following Proposition.

Proposition 1. *For the Bridge Plug-in parameterization, under Assumption (**A1**), a minimizer of (8) is $\hat{\mathbf{x}}_0^{\theta_\star}(\mathbf{x}_t, t) = \hat{\mathbf{x}}^{\mathrm{loo}}_{0|t}(\mathbf{x}_t)$. In addition, $p^\ell_{s|t}(\cdot|\mathbf{x}_t) = q^\ell_{s|0,t}(\cdot|\hat{\mathbf{x}}^{\mathrm{loo}}_{0|t}(\mathbf{x}_t)^\ell, \mathbf{x}_t^\ell)$. Furthermore, this minimizer is unique in the case of UDM.*

The proof is given in Appendix A.1. Thus, the Bridge Plug-in parameterization can attain the minimum of the ELBO, but the optimal prediction is, in general, the leave-one-out posterior rather than the denoising posterior, which can be substantially different quantities, see Appendix A.6. Masked diffusion is a special case in which the distinction between the leave-one-out posterior and the denoising posterior disappears on masked positions: if $\mathbf{x}_t^\ell = \mathsf{m}$, then $p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}_0^\ell|\mathbf{x}_t^{-\ell}) = p^\ell_{0|t}(\mathbf{x}_0^\ell|\mathbf{x}_t)$. On unmasked positions, by contrast, the denoiser is the Dirac mass at $\mathbf{x}_t^\ell$ whereas the leave-one-out posterior need not be.

The difference between the leave-one-out and denoiser is not an obstacle to using the usual training and sampling tools, because for UDM the leave-one-out posterior, the denoising posterior, and the score can be converted into one another explicitly, *i.e.*

$$p^\ell_{0|t}(\mathbf{x}_0^\ell|\mathbf{x}_t) \propto p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}_0^\ell|\mathbf{x}_t^{-\ell}) q^\ell_{t|0}(\mathbf{x}_t^\ell|\mathbf{x}_0^\ell) \tag{10}$$

and conversely this provides a conversion from denoiser to LOO denoiser since for UDM, $q^\ell_{t|0}(x_t|x_0) > 0$ for all x_0, so we have $p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}_0^\ell|\mathbf{x}_t^{-\ell}) \propto p^\ell_{0|t}(\mathbf{x}_0^\ell|\mathbf{x}_t)/q^\ell_{t|0}(\mathbf{x}_t^\ell|\mathbf{x}_0^\ell)$. These formulas hold more generally for other corruption processes; see Appendix A.2. We can also convert

[1] Interestingly, with an affine extension of the bridge, which always exists, the two parameterizations are equivalent by construction (see Appendix D for a more thorough discussion).

between the leave-one-out posterior and the concrete score [2, 18] $\mathbf{s}_t(\mathbf{x}_t) \in \mathbb{R}^{L\times K}$ defined by $\langle \mathbf{y}^\ell, \mathbf{s}_t(\mathbf{x}_t)^\ell\rangle = p_t(\mathbf{y})/p_t(\mathbf{x}_t)$ with $\mathbf{y}^\ell \neq \mathbf{x}_t^\ell$ and $\mathbf{y}^{-\ell} = \mathbf{x}_t^{-\ell}$. We then have that

$$\langle \mathbf{y}^\ell, \mathbf{s}_t(\mathbf{x}_t)^\ell\rangle = q_{t|0}(\mathbf{y}^\ell|\hat{\mathbf{x}}^{\text{loo}}_{0|t}(\mathbf{x}_t)^\ell)/q_{t|0}(\mathbf{x}_t^\ell|\hat{\mathbf{x}}^{\text{loo}}_{0|t}(\mathbf{x}_t)^\ell)\ . \tag{11}$$

We now focus on the practical consequences of these conversion formulas:

Training with the leave-one-out. A first consequence is that recent works on UDMs that use the Bridge Plug-in parameterization [13, 29, 30] therefore train a model whose target is the leave-one-out posterior rather than the denoising posterior. A structural property of this target is that $\hat{\mathbf{x}}^{\text{loo}}_{0|t}(\mathbf{x}_t)^\ell$ does not depend on $\mathbf{x}_t^\ell$. This invariance can be enforced architecturally with a Hollow Transformer [34, 40], whose ℓ-th output is prevented from attending to the ℓ-th input position. However, for text modeling this architecture is harder to train and performs worse than standard attention [40], which leaves the invariance to be learned. In that case, the sensitivity of the ℓ-th prediction to $\mathbf{x}_t^\ell$ provides a direct diagnostic for suboptimality of the learned leave-one-out model or a trained denoiser, provided that we convert it first to a leave-one-out using (10); see Appendix I.

The conversion (10) allows us to train a leave-one-out model with the usual denoising cross-entropy. Indeed, if a network outputs logits $\boldsymbol{f}_\theta(\mathbf{x}_t, t)$ and we set the parameterized prediction as $\hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t)^\ell = \text{softmax}(\boldsymbol{f}_\theta(\mathbf{x}_t, t)^\ell)$, then the associated denoiser is

$$\boldsymbol{d}_\theta(\mathbf{x}_t, t)^\ell = \text{softmax}\Big(\boldsymbol{f}_\theta(\mathbf{x}_t, t)^\ell + \log\Big(1 + \tfrac{K\alpha_t}{1-\alpha_t}\Big)\cdot \mathbf{x}_t^\ell\Big)\ . \tag{12}$$

Thus, for UDM, one may optimize the standard cross-entropy loss

$$-\int_0^1 \mathbb{E}\big[\textstyle\sum_{\ell=1}^L \log\langle X_0^\ell, \boldsymbol{d}_\theta(X_t, t)^\ell\rangle\big]\,\mathrm{d}t\ , \tag{13}$$

while the underlying prediction $\hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t)$ parameterizes the leave-one-out posterior. At optimality $\boldsymbol{d}_\theta(\mathbf{x}_t, t) = \mathbb{E}[X_0|X_t = \mathbf{x}_t]$ and since the conversion formulas define a one-to-one correspondence between the denoiser and leave-one-out denoiser *in the case of UDM*, we have $\hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t) = \hat{\mathbf{x}}^{\text{loo}}_{0|t}(\mathbf{x}_t)$. Experimentally we find that this loss outperforms the standard cross-entropy loss in which the model parameterizes a plain denoiser and performs as well as the continuous-time UDM plug-in NELBO used in [30] while being simpler.

Predictor-corrector sampling. The leave-one-out denoiser also gives access to the one-coordinate conditional of p_t given by $p_t^\ell(\mathbf{x}_t^\ell|\mathbf{x}_t^{-\ell}) := p_t(\mathbf{x}_t)/\sum_{\tilde{\mathbf{x}}_t^\ell} p_t(\tilde{\mathbf{x}}_t)$. Following [34] it holds that:

$$p_t^\ell(\cdot|\mathbf{x}_t^{-\ell}) = q_{t|0}^\ell(\cdot|\hat{\mathbf{x}}^{\text{loo}}_{0|t}(\mathbf{x}_t)^\ell) = \text{Cat}(\alpha_t\hat{\mathbf{x}}^{\text{loo}}_{0|t}(\mathbf{x}_t)^\ell + (1-\alpha_t)\boldsymbol{\pi}^\ell)\ . \tag{14}$$

See Appendix A.5 for a proof. This observation yields a predictor-corrector [2, 37] in the same spirit as [40] without needing to train an additional model. Concretely, iterating the corresponding Gibbs updates gives a corrector kernel that preserves p_t, making it a valid corrector step. If one trains a denoiser or score function instead, the conversion formulas above recover the leave-one-out posterior needed to evaluate the same corrector. In practice, we use the margin-based heuristic of [40], with parallel updates on low-confidence positions; the full sampler is given in Appendix E.

4 Bridging Masked and Uniform Diffusion

In the previous section we clarified the difference between existing parameterizations of UDMs. We now give an alternative view: uniform diffusion can be represented as an absorbing-state diffusion after conditioning on a random sequence of absorbing states. This perspective is useful since it bridges uniform and masked diffusion. A complementary construction in Appendix C preserves the UDM joint law while using an MDM denoiser.

4.1 Absorbing state uniform diffusion

Let $U \sim \upsilon := \text{Uniform}([\![1, K]\!])^L$ be an auxiliary noise variable. Conditionally on a realization $U = \mathbf{u}$, define a Markov process $(X_t)_{t\in[0,1]}$ with transition kernel $q_{t|s}(\mathbf{x}_t|\mathbf{x}_s, \mathbf{u}) = \prod_{\ell=1}^L \text{Cat}(\mathbf{x}_t^\ell; \alpha_{t|s}\mathbf{x}_s^\ell + (1-\alpha_{t|s})\mathbf{u}^\ell)$. Thus, conditioned on $U = \mathbf{u}$, each coordinate evolves

as an absorbing-state diffusion whose absorbing state is fixed to $\mathbf{u}^\ell$. In contrast with the usual absorbing diffusion, the absorbing state is not a distinguished mask token, but a random token sampled independently at each position.

Conditionally on $U = \mathbf{u}$, the law of X_t is $p_t(\mathbf{x}_t|\mathbf{u}) \coloneqq \sum_{\mathbf{x}_0} q_{t|0}(\mathbf{x}_t|\mathbf{x}_0, \mathbf{u})\, p_0(\mathbf{x}_0)$. Averaging over the auxiliary variable recovers the usual uniform-diffusion marginal: $\sum_{\mathbf{u}} p_t(\mathbf{x}_t|\mathbf{u})\, \upsilon(\mathbf{u}) = p_t(\mathbf{x}_t)$, which follows from the identity $\sum_{\mathbf{u}} q_{t|s}(\mathbf{x}_t|\mathbf{x}_s, \mathbf{u})\, \upsilon(\mathbf{u}) = q_{t|s}(\mathbf{x}_t|\mathbf{x}_s)$ which states that the UDM forward transition is a mixture of absorbing-state transitions. Therefore, the joint process (U, X_t) has the same marginal law for X_t as the original UDM, while its conditional dynamics given U are those of an absorbing-state process with position-dependent absorbing states.

This conditional viewpoint gives a direct analogue of the masked-diffusion carry-over structure. Indeed, since U is fixed in the conditional process, the bridge $q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{x}_t, \mathbf{u})$ is token-wise and, for each coordinate ℓ, is given by

$$q^\ell_{s|0,t}(\mathbf{x}^\ell_s|\mathbf{x}^\ell_0, \mathbf{x}^\ell_t, \mathbf{u}^\ell) = \begin{cases} \mathrm{Cat}\left(\mathbf{x}^\ell_s; \mathbf{x}^\ell_t\right) & \text{if } \mathbf{x}^\ell_t \neq \mathbf{u}^\ell \\ \mathrm{Cat}\left(\mathbf{x}^\ell_s; \frac{\alpha_s - \alpha_t}{1-\alpha_t}\mathbf{x}^\ell_0 + \frac{1-\alpha_s}{1-\alpha_t}\mathbf{u}^\ell\right) & \text{otherwise.} \end{cases} \tag{15}$$

The same dichotomy appears in the conditional denoiser. By definition:

$$p_{0|t}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{u}) \propto p_0(\mathbf{x}_0) \prod_{\ell:\mathbf{x}^\ell_t = \mathbf{u}^\ell} \left[\alpha_t \mathbb{1}_{\mathbf{x}^\ell_0 = \mathbf{x}^\ell_t} + (1-\alpha_t)\right] \prod_{\ell:\mathbf{x}^\ell_t \neq \mathbf{u}^\ell} \mathbb{1}_{\mathbf{x}^\ell_0 = \mathbf{x}^\ell_t}. \tag{16}$$

Consequently, for every coordinate ℓ with $\mathbf{x}^\ell_t \neq \mathbf{u}^\ell$, the posterior is degenerate at $\mathbf{x}^\ell_t$; meaning that $p^\ell_{0|t}(\cdot|\mathbf{x}_t, \mathbf{u}) = \mathrm{Cat}(\mathbf{x}^\ell_t)$. In contrast, the coordinates with $\mathbf{x}^\ell_t = \mathbf{u}^\ell$ remain ambiguous and must be predicted from the full noisy sequence. However, not all convenient properties of MDM carry over to this setting. In particular, unlike in MDMs [23, 41], the noise-conditioned denoiser remains explicitly time-dependent. For example, if we consider the single token case $L = 1$, then when $\mathbf{x}^\ell_t = \mathbf{u}$, we have that $p_{0|t}(\mathbf{x}_t|\mathbf{x}_t, \mathbf{u}) = p_0(\mathbf{x}_t)/(1 - \alpha_t \sum_{\tilde{\mathbf{x}}_0 \neq \mathbf{x}_t} p_0(\tilde{\mathbf{x}}_0))$, showing that the noise-conditioned denoiser is indeed time-dependent.

Model and objective. The structure above suggests the following parameterization:

$$\hat{\mathbf{x}}^\theta_0(\mathbf{x}_t, t; \mathbf{u})^\ell \coloneqq \begin{cases} \mathrm{softmax}(\boldsymbol{f}_\theta(\mathbf{x}_t, t; \mathbf{u})^\ell) & \text{if } \mathbf{x}^\ell_t = \mathbf{u}^\ell \\ \mathbf{x}^\ell_t & \text{if } \mathbf{x}^\ell_t \neq \mathbf{u}^\ell\,. \end{cases} \tag{17}$$

We now define the corresponding generative model. The sampler first draws a latent absorbing sequence $\mathbf{u} \sim \upsilon$, initializes the terminal state from the point mass at $\mathbf{u}$, and then runs the learned reverse process. The joint model is $p^\theta_{0:n}(\mathbf{x}_{t_{0:n}}) \coloneqq \sum_{\mathbf{u}} p_{t_n}(\mathbf{x}_{t_n}|\mathbf{u}) \prod_{i=1}^n p^\theta_{s_i|t_i}(\mathbf{x}_{s_i}|\mathbf{x}_{t_i}, \mathbf{u})\, \upsilon(\mathbf{u})$. with $p^\theta_{s|t}(\mathbf{x}_s|\mathbf{x}_t, \mathbf{u}) \coloneqq q_{s|0,t}(\mathbf{x}_s|\hat{\mathbf{x}}^\theta_0(\mathbf{x}_t, t; \mathbf{u}), \mathbf{x}_t, \mathbf{u})$; *i.e.* we use the Bridge Plug-in parameterization which coincides here with Marginalization. Applying Jensen's inequality to $-\log p^\theta_0(\mathbf{x}_0)$ where p^θ_0 is the time-zero marginal of the joint model above and taking the limit $n \to \infty$ in the conditional NELBO (5) appearing in the upper bound gives the following continuous-time objective. A proof of the next Proposition is provided in Appendix B.

Proposition 2. *A continuous time NELBO for* $-\log p^\theta_0(\mathbf{x}_0)$ *is*

$$\mathrm{L}^{\mathrm{AUDM}}_\infty(\mathbf{x}_0; \theta) \coloneqq -\int_0^1 \tfrac{\alpha'_t}{1-\alpha_t}\, \mathbb{E}\big[\textstyle\sum_{\ell=1}^L \mathbb{1}_{X^\ell_t = U^\ell} \big\{1 - \langle U^\ell, \hat{\mathbf{x}}^\theta_0(X_t, t, U)^\ell\rangle \\ - \mathbb{1}_{\mathbf{x}^\ell_0 \neq U^\ell} \cdot \big(1 + \log \langle \mathbf{x}^\ell_0, \hat{\mathbf{x}}^\theta_0(X_t, t, U)^\ell\rangle\big)\big\}\big]\mathrm{d}t, \tag{18}$$

where the expectation is w.r.t. $U \sim$ *Uniform*$(\llbracket 1, K \rrbracket)^{\otimes L}$, *and* $X_t \sim q_{t|0}(\cdot|\mathbf{x}_0, U)$.

This loss can be viewed as a generalization of the continuous-time ELBO for MDMs. Indeed, upon taking $U^\ell \sim \mathrm{Cat}(\mathsf{m})$ for all $\ell \in \llbracket 1, L \rrbracket$ we recover the MDM ELBO by further imposing that $\langle \mathsf{m}, \hat{\mathbf{x}}^\theta_0(\mathbf{x}, t)^\ell\rangle = 0$ for any $\mathbf{x}$, which is referred to as the zero-remasking property as in [28].

4.2 AUDM with resampling recovers UDM

The construction above matches the single-time marginals of uniform diffusion. We now show that, by resampling the absorbing sequence along the reverse chain, one can recover the full UDM

Algorithm 1 Remasked AUDM sampler

Require: time grid $0 = t_0 < t_1 < \cdots < t_n = 1$
Draw $U_{t_n} \sim \mathrm{Uniform}([\![1, K]\!])^{\otimes L}$ and $X_{t_n} \sim p_{t_n}(\cdot|U_{t_n})$
for $i = n-1, n-2, \ldots, 0$ **do**
Draw $X_0 \sim p^{\theta}_{0|t_{i+1}}(\cdot|X_{t_{i+1}}, U_{t_{i+1}})$
Draw $X_{t_i} \sim q_{t_i|0,t_{i+1}}(\cdot|X_0, X_{t_{i+1}})$
Draw $U^{\ell}_{t_i} \sim \begin{cases} \delta_{X^{\ell}_{t_i}}, & X^{\ell}_{t_i} \neq X^{\ell}_0, \\ \mathrm{Cat}\left(\frac{X^{\ell}_0 + \alpha_{t_i}(\mathbf{1} - X^{\ell}_0)}{1+(K-1)\alpha_{t_i}}\right), & X^{\ell}_{t_i} = X^{\ell}_0, \end{cases}$ ▷ in parallel over ℓ
end for
return X_{t_0}

reverse-chain:

$$p_{0:n}(\mathbf{x}_{t_{0:n}}) = p_{t_n}(\mathbf{x}_{t_n}) \prod_{i=1}^{n} p_{s_i|t_i}(\mathbf{x}_{s_i}|\mathbf{x}_{t_i}) \,, \tag{19}$$

We describe the resampling procedure in Algorithm 1. It is a discrete-time augmented state Markov chain with one step transition $\bar{p}_{s|t}(\mathbf{x}_s, \mathbf{u}_s|\mathbf{x}_t, \mathbf{u}_t) = \sum_{\mathbf{x}_0} \upsilon_{s|0}(\mathbf{u}_s|\mathbf{x}_0, \mathbf{x}_s) q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{x}_t)\, p_{0|t}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{u}_t)$. Equivalently, given $(\mathbf{x}_t, \mathbf{u}_t)$, the sampler first draws X_0 from the absorbing-state denoiser, then draws X_s from the UDM bridge, and finally refreshes the absorbing sequence by drawing U_s conditionally on X_0, X_s. Thus, compared with the fixed-U sampler described above, the only additional operation is the final resampling step. The resampling distribution is defined as

$$\upsilon_{s|0}(\mathbf{u}_s|\mathbf{x}_0, \mathbf{x}_s) := \frac{\upsilon(\mathbf{u}_s) q_{s|0}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{u}_s)}{q_{s|0}(\mathbf{x}_s|\mathbf{x}_0)} = \prod_{\ell=1}^{L} \begin{cases} \mathrm{Cat}(\mathbf{u}^{\ell}_s; \mathbf{x}^{\ell}_s) & \mathbf{x}^{\ell}_s \neq \mathbf{x}^{\ell}_0, \\ \mathrm{Cat}\left(\mathbf{u}^{\ell}_s; \frac{\alpha_s \mathbf{1} + (1-\alpha_s)\mathbf{x}^{\ell}_0}{1+(K-1)\alpha_s}\right), & \mathbf{x}^{\ell}_s = \mathbf{x}^{\ell}_0, \end{cases} \tag{20}$$

The first case says that, if a coordinate changed between $\mathbf{x}^{\ell}_0$ and $\mathbf{x}^{\ell}_s$, then the absorbing value, or noise, must be the observed value $\mathbf{x}^{\ell}_s$. The second case accounts for the ambiguity when the coordinate did not change. This sampler preserves the UDM joint law (19) marginally as stated in the next Proposition.

Proposition 3. *The trajectory generated in Algorithm 1 satisfies Law*$(X_{t_0}, \ldots, X_{t_n}) = p_{0:n}$*, where* $p_{0:n}$ *is the uniform UDM reverse-chain law defined in* (19)*.*

The proof is given in Appendix B.2. This result essentially shows that uniform diffusion can be decomposed into simple sampling operations: first sampling from an absorbing-state denoiser, then the uniform bridge, and then resampling the absorbing-state according to (20). In practice, we take as approximate model one that uses the factorized transitions of the joint law on the augmented states generated by Algorithm 1, which is equivalent to replacing the exact $p_{0|t}(\cdot|\mathbf{x}_t, \mathbf{u}_t)$ by the approximation introduced in the previous section. For the initialization we set $X_{t_n} = U_{t_n}$.

5 Experiments

This section tests the practical implications of our analysis for UDM parameterization, sampling, and reverse process design. We compare the LOO denoiser and plain denoiser as primitive learned objects, evaluate the predictor-corrector sampler of Section 3.2, and compare AUDM and ReAUDM from Section 4 against UDMs and MDMs. We do not benchmark against wrapper methods or more elaborate UDM variants, which act at another level of the modeling pipeline; our conclusions are complementary to such methods and can guide the parameterization on which they build. We evaluate language modeling, Sudoku, and TinyGSM reasoning, where LOO improves GSM8K accuracy at all temperatures (Appendix J.3). All the experimental details are provided in Appendix J.

Language modeling. We experiment on One Billion Word (LM1B [4]) and OpenWebText (OWT [9]), following [18, 28, 29]. LM1B uses the bert-base-uncased tokenizer with $L = 128$, while OWT uses the gpt2 tokenizer with $L = 1024$. All models use the same DiT-based architecture as [18, 29] and are trained from scratch for 1M steps with batch size 512. To assess sample quality, we report GPT-2 Large generative perplexity (Gen-PPL) with the OWT-trained model. Similarly to [17, 27] we rely on the *generative frontier* where we plot the Gen-PPL as a function of 1-gram

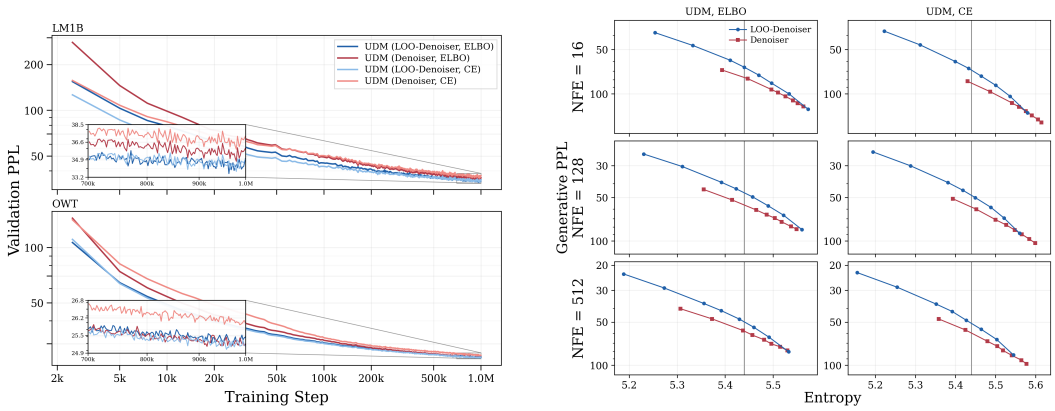


Figure 1: Comparison between the denoiser and leave-one-out parameterizations. **Left:** validation perplexity during training. **Right:** top-p sampling Gen-PPL frontier, obtained by sweeping ($p \in [0.8, 1.0]$). Top$-p$ is applied to the LOO denoiser and denoiser respectively. The curves are shown only up to entropy 5.6. The vertical line corresponds to the dataset entropy

Table 1: Perplexity (↓). OWT is the validation distribution; other columns report zero-shot transfer. Bold marks the overall best value, and orange the best within each method family.

Method	Loss	Param.	OWT	AG News	Lambada	LM1B	PTB	PubMed	WikiText
Absorbing Uniform Diffusion									
	ELBO	Denoiser	23.32	**61.37**	**45.87**	**72.02**	**72.21**	**43.01**	**33.22**
Uniform Diffusion									
	Cross-Entropy	Denoiser	25.45	74.03	49.04	83.43	84.13	44.66	39.17
	Cross-Entropy	LOO-Denoiser	25.18	72.66	46.91	81.54	87.79	43.72	36.67
	ELBO	Denoiser	25.20	73.33	46.80	80.33	84.94	45.75	37.44
	ELBO	LOO-Denoiser	25.30	73.82	46.22	82.03	83.24	44.52	37.23
Max Coupling Uniform Diffusion									
	ELBO	Denoiser	32.96	112.64	64.71	117.21	110.78	65.82	51.40
	ELBO	LOO-Denoiser	32.21	109.97	62.44	112.70	106.53	61.20	49.41
Masked Diffusion									
	ELBO	Denoiser	**22.90**	64.57	48.79	73.06	74.99	43.13	35.16

entropy. We vary the entropy using either temperature or nucleus sampling. This evaluation is substantially more informative than reporting the performance at a single entropy, as emphasized by [27] since different entropies may reverse the Gen-PPL ordering of two models.

Sudoku. We also run a 9×9 Sudoku completion experiment to test the same modeling choices on a different structured discrete task. We use a similar setup to [38]. The model is a smaller DiT with about 6M parameters, trained on a 50k-puzzle Sudoku split. We sweep the learning rate independently for every method family and report, for each method, the learning rate that maximizes the average validation solve rate across seeds and NFEs. The full setup is summarized in Table 5.

5.1 Leave-one-out parameterization

LOO denoiser vs denoiser. We first isolate the effect of the parameterization by training UDMs with either the denoiser or the LOO denoiser parameterization under matched architectures. We consider both cross-entropy and continuous-time ELBO training for both parameterizations, using the standard UDM ELBO of [30] and the Maximal Coupling[2] ELBO derived in Proposition 10. The configuration ELBO with the denoiser parameterization is the continuous-time analogue of D3PM Uniform trained without the auxiliary cross-entropy term in the hybrid objective [1, Equation 5]. Although [30] compare against D3PM Uniform, their comparison also changes the training objective: the D3PM baseline is trained with a discrete-time loss, while the plug-in UDM model uses a continuous-time loss, which already improves perplexity; see [30, Figure 2]. We therefore

[2]This model is related to the DDIM sampler for UDMs derived in [32, Appendix A] and is used to confirm that our findings carry over to different processes. This process is discussed extensively in Appendix G.3

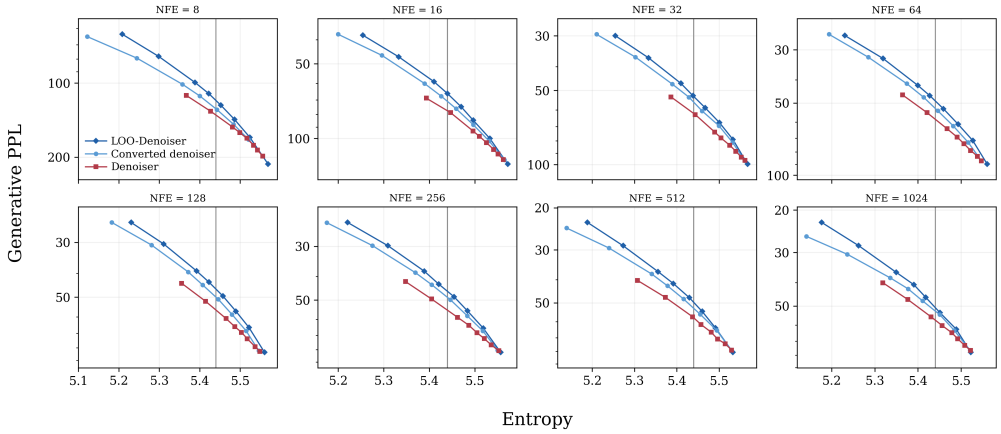


Figure 2: Top-p sampling Gen-PPL frontier, obtained by sweeping ($p \in [0.8, 1.0]$). Top-p sampling is applied to the denoiser, denoiser converted into LOO denoiser and LOO denoiser.

train both the denoiser and LOO parameterizations with the same continuous-time loss, isolating the effect of the parameterization itself.

The validation curves on LM1B and OWT, on the left plot of Figure 1 show that the LOO parameterization improves optimization, with better perplexity throughout the training. The zero-shot results in Table 1 show that this trend transfers beyond OWT where in the three settings the LOO improves over its denoiser counterpart on almost all the datasets considered. We then evaluate generation with top-p frontiers, sweeping $p \in [0.8, 1.0]$ and plotting Gen-PPL against 1-gram entropy. The right panel of Figure 1 shows that the LOO parameterization improves the frontier in the relevant low-entropy regime.

Cross-entropy training. Figure 1 shows that cross-entropy training with the denoiser is the weakest combination, despite its simplicity. Using the LOO parameterization largely removes this drawback: cross-entropy becomes competitive with ELBO training while remaining simpler to implement. Our practical takeaway is therefore to use the LOO denoiser as the default UDM prediction target, especially with cross-entropy training.

Applying top-p to the LOO denoiser. In the comparison in Figure 1, top-p is applied to the learned primitive itself: to the LOO denoiser before using Bridge Plug-in and to the denoiser before using Marginalization. A useful consequence of our conversion formulas is that one can also start from a denoiser-trained model, convert the denoiser to the LOO representation, apply top-p there, and then sample with the bridge plug-in parameterization. This gives a non-negligible **improvement of the frontier at no additional cost** when using top-p sampling and reduces the gap with the LOO parameterization, as shown in Figure 2.

Predictor-corrector We next evaluate the predictor-corrector sampler (Algorithm 5) described at the end of Section 3.2. As detailed in Appendix E, the LOO parameterization gives access to a corrector step without training an auxiliary model, in contrast to [40]. We use the confidence-based resampling rule of [40], with $M \in [\![1, 5]\!]$ corrector steps per predictor step and $k \in [\![1, 15]\!]$ parallel token updates per corrector step. We show the results for a trained LOO denoiser on the left panel of Figure 3 and for a trained denoiser converted into a LOO denoiser on the right panel. The resulting predictor-corrector frontiers Pareto-dominate the ancestral temperature frontiers in both cases. In addition to being substantially better, a key advantage is that this correction is cheap: in our implementation, the predictor-corrector sampler has the same runtime as the regular ancestral sampler. For completeness, we also report the comparison with top-p sampling in Figure 9. Thus, the LOO parameterization is useful not only as a training target, but also as a training-free mechanism for improving inference.

5.2 Absorbing-state uniform diffusion

We finally evaluate the absorbing-state uniform diffusion model of Section 4. AUDM uses the same 170M-parameter DiT backbone as the UDM and MDM baselines, and adds a 7M-parameter module

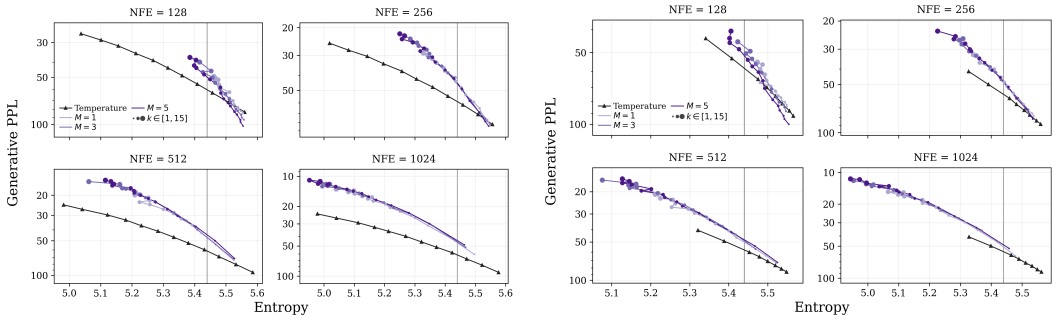


Figure 3: Gen-PPL frontiers. Temperature sampling Gen-PPL frontier, obtained by sweeping the temperature in $[0.8, 1.10]$. **Left:** predictor-corrector sampling using a trained LOO denoiser against ancestral sampling using the same LOO denoiser. **Right:** predictor-corrector sampling using a trained plain denoiser converted at inference time to a LOO against ancestral sampling using the same plain denoiser. The vertical line corresponds to the dataset entropy

to process the auxiliary noise variable U, corresponding to a 4.1% parameter overhead. AUDM gives the best likelihood results among the uniform-diffusion variants in Table 1. It improves over UDM on the OWT validation distribution and on every zero-shot dataset. It also outperforms MDM on all zero-shot datasets, although MDM remains slightly better on OWT validation perplexity. At sampling time, Figure 6 shows that AUDM remains competitive along the generative frontier, and is often slightly better than MDM, uniformly across NFEs. Thus, the absorbing-state construction preserves the marginals of UDMs while improving its likelihood.

The Sudoku results in Figure 6 give a smaller-scale check of the same conclusions outside language modeling. The best solve rates are obtained by UDMs trained with the leave-one-out target, with cross-entropy and ELBO both reaching nearly perfect accuracy. UDM also outperforms MDM on this task. We see that AUDM improves over MDM even though they share structural properties, and interestingly, the ReAUDM construction improves over AUDM even though they use the same checkpoint and the sampling scheme does not add any additional cost or learned component.

6 Conclusion

We revisited uniform discrete diffusion from two complementary perspectives: the parameterization of the reverse kernel and the design of the forward process. On the parameterization side, we showed that the bridge plug-in objective of UDMs is optimized not by the standard denoising posterior but by a leave-one-out posterior. We derived exact conversions between the denoiser, the leave-one-out predictor, and the score, which separate the choice of training target from the choice of sampling parameterization. In practice, this viewpoint leads to a simple cross-entropy objective for the leave-one-out predictor that improves over the standard cross-entropy parameterization, a way to apply top-p or temperature in the most effective representation, and a predictor-corrector sampler that can be obtained from a trained model without any auxiliary network. On the process-design side, we introduced an absorbing-state reformulation of uniform diffusion, together with a resampled version that recovers the full UDM reverse chain, expliciting a direct bridge between masked and uniform diffusion. Empirically, leave-one-out parameterizations consistently improve optimization and the generative frontier of UDMs, while AUDM and ReAUDM match or surpass masked diffusion on several frontier evaluations while improving likelihood within the uniform-diffusion family. Altogether, our results indicate that the practical gap between masked and uniform diffusion is driven less by the corruption marginals themselves than by the learned representation and the inference scheme built on top of them.

Limitations and future work. Our evidence for the advantage of the leave-one-out target remains mainly empirical, and a theoretical explanation for its improved generative frontier remains open. In addition, although AUDM and ReAUDM clarify how masked-style structure can be transferred to uniform diffusion and substantially improve likelihood, the best generative frontiers are still obtained by leave-one-out UDMs. Improving these absorbing-state models through better parameterizations or sampling schemes is therefore an important direction for future work. Finally, when training the AUDM denoiser, we condition on one-hot absorbing tokens. A natural extension would be to replace

this with a simplex-valued conditioning vector, possibly with mass on the mask, so as to interpolate within a single network between AUDM-type posteriors, the standard UDM denoiser obtained at $\mathbf{1}/K$, and more general hybrid corruption mechanisms.

Acknowledgments

The authors thank Jawad Chemaou for carefully proofreading the paper and providing helpful comments.

AD received funding from the Fondation de l'École polytechnique through its Servir la science campaign. AD and SG are supported by the France 2030 program with the reference ANR-25-PEIA-0001 (THEOREM project) AD and SG are supported by Hi! Paris and Agence Nationale de la Recherche (Grant 11-LABX-0047) This work received government funding administered by the National Research Agency (ANR) under the France 2030 program "Hi! PARIS", grant number ANR-23-IACL-0005.

References

[1] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

[2] Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

[3] Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design, 2024. URL https://arxiv.org/abs/2402.04997.

[4] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling, 2014. URL https://arxiv.org/abs/1312.3005.

[5] Zixiang Chen, Huizhuo Yuan, Yongqian Li, Yiwen Kou, Junkai Zhang, and Quanquan Gu. Fast sampling via discrete non-markov diffusion models with predetermined transition time. *Advances in Neural Information Processing Systems*, 37:106870–106905, 2024.

[6] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

[7] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents, 2018. URL https://arxiv.org/abs/1804.05685.

[8] Justin Deschenaux, Caglar Gulcehre, and Subham Sekhar Sahoo. The diffusion duality, chapter ii: ψ-samplers and efficient curriculum, 2026. URL https://arxiv.org/abs/2602.21185.

[9] Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

[10] Google DeepMind. Gemini Diffusion: Google DeepMind's experimental research model. https://blog.google/technology/google-deepmind/gemini-diffusion/, May 2025. Accessed: 2026-05-06.

[11] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis, 2022. URL https://arxiv.org/abs/2111.14822.

[12] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

[13] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows: Learning categorical distributions with normalizing flows. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2021.

[14] Matthew J Johnson, James Saunderson, and Alan Willsky. Analyzing hogwild parallel gaussian gibbs sampling. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/b51a15f382ac914391a58850ab343b00-Paper.pdf.

[15] Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, Aditya Grover, and Volodymyr Kuleshov. Mercury: Ultra-fast language models based on diffusion, 2025. URL https://arxiv.org/abs/2506.17298.

[16] Bingbin Liu, Sebastien Bubeck, Ronen Eldan, Janardhan Kulkarni, Yuanzhi Li, Anh Nguyen, Rachel Ward, and Yi Zhang. TinyGSM: Achieving $> 80\%$ on GSM8K with small language models. *arXiv preprint arXiv:2312.09241*, 2023. URL https://arxiv.org/abs/2312.09241.

[17] Sulin Liu, Juno Nam, Andrew Campbell, Hannes Stärk, Yilun Xu, Tommi Jaakkola, and Rafael Gómez-Bombarelli. Think while you generate: Discrete diffusion with planned denoising. *arXiv preprint arXiv:2410.06264*, 2024.

[18] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, pages 32819–32848. PMLR, 2024.

[19] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL https://aclanthology.org/J93-2004/.

[20] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016. URL https://arxiv.org/abs/1609.07843.

[21] Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=WNvvwK0tut.

[22] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models, 2025. URL https://arxiv.org/abs/2502.09992.

[23] Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.

[24] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context, 2016. URL https://arxiv.org/abs/1606.06031.

[25] William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

[26] Le-Tuyet-Nhi PHAM, Dario Shariatian, Antonio Ocello, Giovanni Conforti, and Alain Oliviero Durmus. Discrete markov probabilistic models: An improved discrete score-based framework with sharp convergence bounds under minimal assumptions. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=biJiSMLGOV.

[27] Patrick Pynadath, Jiaxin Shi, and Ruqi Zhang. Generative frontiers: Why evaluation matters for diffusion language models, 2026. URL https://arxiv.org/abs/2604.02718.

[28] Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

[29] Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin Chiu, and Volodymyr Kuleshov. The diffusion duality. *arXiv preprint arXiv:2506.10892*, 2025.

[30] Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dallatorre, Bernardo P de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. *arXiv preprint arXiv:2412.10193*, 2024.

[31] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167, 2024.

[32] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

[33] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[34] Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=BYWWwSY2G5s.

[35] Dimitri von Rütte, Janis Fluri, Omead Pooladzandi, Bernhard Schölkopf, Thomas Hofmann, and Antonio Orvieto. Scaling behavior of discrete diffusion language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=GDYaNzxt9T.

[36] Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion, 2025. URL https://arxiv.org/abs/2503.04482.

[37] Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling, 2026. URL https://arxiv.org/abs/2503.00307.

[38] Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning, 2025. URL https://arxiv.org/abs/2410.14157.

[39] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification, 2016. URL https://arxiv.org/abs/1509.01626.

[40] Yixiu Zhao, Jiaxin Shi, Feng Chen, Shaul Druckmann, Lester Mackey, and Scott Linderman. Informed correctors for discrete diffusion models, 2025. URL https://arxiv.org/abs/2407.21243.

[41] Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *arXiv preprint arXiv:2409.02908*, 2024.

[42] Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation, 2024. URL https://arxiv.org/abs/2302.05737.

Appendix Contents

A Leave-one-out (LOO) denoiser **16**

A.1 Proof of Proposition 1 . . . 16

A.2 Conversion Formula . . . 18

A.3 Score and leave-one-out denoiser . . . 19

A.4 Categorical ratio matching . . . 20

A.5 Conditional marginal in terms of the LOO denoiser . . . 21

A.6 An example of leave-one-out and denoiser differences . . . 21

B Absorbing State Uniform Diffusion **22**

B.1 AUDM NELBO . . . 22

B.2 Resampled AUDM . . . 23

B.3 Remarks on some related constructions . . . 26

C Masked Uniform Diffusion **29**

C.1 Transition-time construction . . . 29

C.2 Recovering the UDM joint distribution . . . 29

C.3 Derivations and sampling algorithm . . . 30

D On the role of the bridge extension in the plug-in parameterization **32**

E Leave-one-out and predictor-corrector **34**

E.1 On the distinction between LOO and denoiser in [8] . . . 36

F Continuous time perspective **36**

G Uniform Diffusion and Maximal Coupling **38**

G.1 Maximal couplings . . . 38

G.2 Maximal couplings and masked diffusion . . . 38

G.3 Maximum coupling bridge and ELBO for uniform diffusion . . . 38

H Related Works **41**

I Leave-one-out and Hollow Transformers **41**

J Experimental Details **43**

J.1 Setup and hyperparameters . . . 43

J.2 Additional results . . . 44

J.3 Mathematical reasoning with TinyGSM . . . 47

J.4 Generated samples . . . 49

A Leave-one-out (LOO) denoiser

A.1 Proof of Proposition 1

The proof of Proposition 1 relies on the *plug-in* of the prediction of the neural network into the bridge. This assumes a certain extension of the bridge, initially defined for the one-hots to the simplex. To this end, we consider the following assumption:

(**A1**) *Bayes-compatible simplex extension.* For every $x_s, x_t \in \mathsf{V}$, and $\nu \in \Delta_K$ such that $q^\ell_{t|0}(x_t|\nu) > 0$, the bridge extension used in Bridge Plug-in satisfies

$$q^\ell_{s|0,t}(x_s|\nu, x_t) = \frac{q^\ell_{t|s}(x_t|x_s)\, q^\ell_{s|0}(x_s|\nu)}{q^\ell_{t|0}(x_t|\nu)} \,.$$

Proposition 1 (Generalized form)**.** *For the Bridge Plug-in parameterization, under (**A1**), a minimizer of (8) is obtained by setting $\hat{\mathbf{x}}_0^{\theta_\star}(\mathbf{x}_t, t) = \hat{\mathbf{x}}^{\mathrm{loo}}_{0|t}(\mathbf{x}_t)$ on the support of p_t. In particular, for such $\mathbf{x}_t$,*

$$p^\ell_{s|t}(\cdot|\mathbf{x}_t) = q^\ell_{s|0,t}(\cdot|\hat{\mathbf{x}}^{\mathrm{loo}}_{0|t}(\mathbf{x}_t)^\ell, \mathbf{x}^\ell_t) \,.$$

Furthermore, this minimizer is unique in the case of UDM.

Proof. Fix $0 < s < t$, $\ell \in [\![1, L]\!]$ and $\mathbf{x}_t \in \mathsf{X}$ such that $p_t(\mathbf{x}_t) > 0$. We write $\nu^\ell_\star := \hat{\mathbf{x}}^{\mathrm{loo}}_{0|t}(\mathbf{x}_t)^\ell$. We first prove that

$$p^\ell_{s|t}(\cdot|\mathbf{x}_t) = q^\ell_{s|0,t}(\cdot|\nu^\ell_\star, \mathbf{x}^\ell_t) \,. \tag{21}$$

Since (8) is a sum of token-wise KL divergences, this identity implies that $\hat{\mathbf{x}}_0^{\theta_\star}(\mathbf{x}_t, t) = \hat{\mathbf{x}}^{\mathrm{loo}}_{0|t}(\mathbf{x}_t)$ is one minimizer. For every $\mathbf{x}^\ell_s \in \mathsf{V}$, the definition of the marginal reverse transition and the factorization of the bridge across the dimensions yields

$$\begin{aligned} p^\ell_{s|t}(\mathbf{x}^\ell_s|\mathbf{x}_t) &= \textstyle\sum_{\mathbf{x}_s^{-\ell}} p_{s|t}(\mathbf{x}_s|\mathbf{x}_t) \\ &= \textstyle\sum_{\mathbf{x}_s^{-\ell}, \mathbf{x}_0} q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{x}_t)\, p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) \\ &= \textstyle\sum_{\mathbf{x}^\ell_0} q^\ell_{s|0,t}(\mathbf{x}^\ell_s|\mathbf{x}^\ell_0, \mathbf{x}^\ell_t)\, p^\ell_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}_t). \end{aligned}$$

Moreover, for every $(\mathbf{x}^\ell_0, \mathbf{x}^\ell_t)$, the product identity

$$q^\ell_{t|0}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_0) q^\ell_{s|0,t}(\mathbf{x}^\ell_s|\mathbf{x}^\ell_0, \mathbf{x}^\ell_t) = q^\ell_{t|s}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_s)\, q^\ell_{s|0}(\mathbf{x}^\ell_s|\mathbf{x}^\ell_0) \,.$$

holds. This is the Bayes' formula if $q^\ell_{t|0}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_0) > 0$. Otherwise the left hand side vanishes by the Chapman–Kolmogorov identity and we still get equality. We also write the posterior marginal at position ℓ as

$$p^\ell_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}_t) = \frac{p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}_t^{-\ell})\, q^\ell_{t|0}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_0)}{\sum_{\tilde{\mathbf{x}}^\ell_0 \in \mathsf{V}} p^{\mathrm{loo},\ell}_{0|t}(\tilde{\mathbf{x}}^\ell_0|\mathbf{x}_t^{-\ell})\, q^\ell_{t|0}(\mathbf{x}^\ell_t|\tilde{\mathbf{x}}^\ell_0)} \,.$$

Since $x_0 \mapsto q^\ell_{t|0}(\mathbf{x}^\ell_t|x_0)$ is linear in its first argument, the denominator is exactly $q^\ell_{t|0}(\mathbf{x}^\ell_t|\nu^\ell_\star)$. It is strictly positive since it is the conditional probability of $\mathbf{x}^\ell_t$ given $\mathbf{x}_t^{-\ell}$ under p_t by Proposition 5, and the fixed state $\mathbf{x}_t$ is in the support of p_t. Substituting the two previous identities yields

$$p^\ell_{s|t}(\mathbf{x}^\ell_s|\mathbf{x}_t) = \frac{q^\ell_{t|s}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_s) \sum_{\mathbf{x}^\ell_0 \in \mathsf{V}} q^\ell_{s|0}(\mathbf{x}^\ell_s|\mathbf{x}^\ell_0)\, p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}_t^{-\ell})}{q^\ell_{t|0}(\mathbf{x}^\ell_t|\nu^\ell_\star)} \,.$$

Using again the linearity of the forward transition,

$$\sum_{\mathbf{x}^\ell_0 \in \mathsf{V}} q^\ell_{s|0}(\mathbf{x}^\ell_s|\mathbf{x}^\ell_0)\, p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}_t^{-\ell}) = q^\ell_{s|0}(\mathbf{x}^\ell_s|\nu^\ell_\star)$$

and therefore

$$p^\ell_{s|t}(\mathbf{x}^\ell_s|\mathbf{x}_t) = \frac{q^\ell_{t|s}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_s)\, q^\ell_{s|0}(\mathbf{x}^\ell_s|\nu^\ell_\star)}{q^\ell_{t|0}(\mathbf{x}^\ell_t|\nu^\ell_\star)} \,.$$

By (**A1**), applied with $\nu = \nu_\star^\ell$ and $x_t = \mathbf{x}_t^\ell$, we have

$$p_{s|t}^\ell(\mathbf{x}_s^\ell|\mathbf{x}_t) = \frac{q_{t|s}^\ell(\mathbf{x}_t^\ell|\mathbf{x}_s^\ell)\, q_{s|0}^\ell(\mathbf{x}_s^\ell|\nu_\star^\ell)}{q_{t|0}^\ell(\mathbf{x}_t^\ell|\nu_\star^\ell)} = q_{s|0,t}^\ell(\mathbf{x}_s^\ell|\nu_\star^\ell, \mathbf{x}_t^\ell),$$

which is exactly (21).

We now prove uniqueness in the UDM case. Fix ℓ and $\mathbf{x}_t$. The corresponding token-wise term in (8) is minimized if and only if the model marginal transition coincides with $p_{s|t}^\ell(\cdot|\mathbf{x}_t)$. Under the plug-in parameterization, this marginal is of the form $q_{s|0,t}^\ell(\cdot|\nu, \mathbf{x}_t^\ell)$ with $\nu = \hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t)^\ell \in \Delta_K$. It is therefore enough to prove that, for UDM, the map

$$\nu \longmapsto q_{s|0,t}^\ell(\cdot|\nu, \mathbf{x}_t^\ell)$$

is injective.

Write $\mathbf{x}_t^\ell = e_k$. By (7), for every $\nu \in \Delta_K$,

$$q_{s|0,t}^\ell(\cdot|\nu, e_k) = \mathrm{Cat}\left(\cdot; \frac{K\alpha_t\nu_k e_k + (\alpha_{t|s} - \alpha_t)e_k + (\alpha_s - \alpha_t)\nu + D_{s,t}\mathbf{1}/K}{K\alpha_t\nu_k + 1 - \alpha_t}\right).$$

Hence, for every $j \neq k$,

$$\langle e_j, q_{s|0,t}^\ell(\cdot|\nu, e_k)\rangle = \frac{(\alpha_s - \alpha_t)\nu_j + D_{s,t}/K}{K\alpha_t\nu_k + 1 - \alpha_t}\,. \tag{22}$$

Summing (22) over $j \neq k$ gives

$$1 - \langle e_k, q_{s|0,t}^\ell(\cdot|\nu, e_k)\rangle = \frac{(\alpha_s - \alpha_t)(1 - \nu_k) + (K-1)D_{s,t}/K}{K\alpha_t\nu_k + 1 - \alpha_t}\,.$$

Since $0 < s < t$, we have $\alpha_s - \alpha_t > 0$ and $D_{s,t} > 0$. The numerator is therefore strictly decreasing in ν_k, whereas the denominator is strictly increasing in ν_k. The ratio is thus strictly decreasing and determines ν_k uniquely from $q_{s|0,t}^\ell(\cdot|\nu, e_k)$. Once ν_k is known, (22) yields, for every $j \neq k$,

$$\nu_j = \frac{(K\alpha_t\nu_k + 1 - \alpha_t)\langle e_j, q_{s|0,t}^\ell(\cdot|\nu, e_k)\rangle - D_{s,t}/K}{\alpha_s - \alpha_t}\,.$$

Hence every coordinate of ν is uniquely determined by $q_{s|0,t}^\ell(\cdot|\nu, e_k)$, so the map is injective. Since $\nu_\star^\ell = \hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)^\ell$ attains the minimum by the first part of the proof, it is the unique minimizer in the UDM case. □

Assumption (**A1**) is a condition on the simplex extension used by the plug-in parameterization. When $q_{t|0}^\ell(x_t|x_0) > 0$, the bridge is defined via the Bayes' formula as

$$q_{s|0,t}^\ell(x_s|x_0, x_t) = \frac{q_{t|s}^\ell(x_t|x_s)\, q_{s|0}^\ell(x_s|x_0)}{q_{t|0}^\ell(x_t|x_0)}\,.$$

When $q_{t|0}^\ell(x_t|x_0) = 0$, the value of the one-hot bridge must be completed by convention. This first completion does not determine the second extension to a simplex-valued first argument, and (**A1**) requires this second extension to keep the Bayes-ratio form on the simplex.

For the MDM bridge used in (53), this compatibility holds on the simplex, where $\langle \mathsf{m}, \nu\rangle = 0$. If $\mathbf{x}_t^\ell = e_k \neq \mathsf{m}$ and $\langle e_k, \nu\rangle > 0$, then

$$\frac{q_{t|s}^\ell(e_k|\mathbf{x}_s^\ell)\, q_{s|0}^\ell(\mathbf{x}_s^\ell|\nu)}{q_{t|0}^\ell(e_k|\nu)} = \frac{\alpha_{t|s}\alpha_s\langle e_k, \nu\rangle\, \mathbb{1}\{\mathbf{x}_s^\ell = e_k\}}{\alpha_t\langle e_k, \nu\rangle} = \mathbb{1}\{\mathbf{x}_s^\ell = e_k\},$$

so the Bayes-ratio expression gives $\mathrm{Cat}(\cdot; \mathbf{x}_t^\ell)$, which is the first branch of (53). If $\mathbf{x}_t^\ell = \mathsf{m}$, then

$$\frac{q_{t|s}^\ell(\mathsf{m}|\mathbf{x}_s^\ell)\, q_{s|0}^\ell(\mathbf{x}_s^\ell|\nu)}{q_{t|0}^\ell(\mathsf{m}|\nu)} = \frac{\alpha_s - \alpha_t}{1 - \alpha_t}\langle \mathbf{x}_s^\ell, \nu\rangle \mathbb{1}\{\mathbf{x}_s^\ell \neq \mathsf{m}\} + \frac{1 - \alpha_s}{1 - \alpha_t}\mathbb{1}\{\mathbf{x}_s^\ell = \mathsf{m}\},$$

or, equivalently,

$$\mathrm{Cat}\left(\cdot; \frac{\alpha_s - \alpha_t}{1 - \alpha_t}\nu + \frac{1 - \alpha_s}{1 - \alpha_t}\mathsf{m}\right),$$

which is the second branch of (53) with x_0 replaced by ν. Therefore (**A1**) is verified and Proposition 1 holds.

However, the assumption is not automatic for an arbitrary simplex extension. For example, suppose that, on unsupported visible pairs, one completes the bridge by the rule $\mathrm{Cat}(\cdot; \frac{\alpha_s-\alpha_t}{1-\alpha_t}x_0 + \frac{1-\alpha_s}{1-\alpha_t}x_t)$ and then extends this rule to ν as $\mathrm{Cat}(\cdot; \frac{\alpha_s-\alpha_t}{1-\alpha_t}\nu + \frac{1-\alpha_s}{1-\alpha_t}x_t)$, without the MDM case split. If $x_t = e_k$, $\langle e_k, \nu\rangle > 0$, and $j \neq k$, then the Bayes-ratio extension satisfies

$$\frac{q^\ell_{t|s}(e_k|e_j)\, q^\ell_{s|0}(e_j|\nu)}{q^\ell_{t|0}(e_k|\nu)} = 0,$$

whereas the naive simplex extension gives

$$\langle e_j, \frac{\alpha_s-\alpha_t}{1-\alpha_t}\nu + \frac{1-\alpha_s}{1-\alpha_t}e_k\rangle = \frac{\alpha_s-\alpha_t}{1-\alpha_t}\langle e_j, \nu\rangle.$$

Thus (**A1**) fails in general. If the leave-one-out posterior has positive mass on such a token e_j, plugging it into this naive extension does not reproduce the exact marginal reverse transition, and the conclusion of Proposition 1 can be false. Therefore, depending on the extension that one uses for the bridge, Proposition 1 may or may not hold. In the paper however, both the bridge considered satisfy (**A1**). And the assumption stated in the main paper: $q^\ell_{t|0}(x_t|x_0) > 0$ for all $x_0, x_t \in \mathsf{V}$, is a sufficient condition for the canonical bridge extension to verify (**A1**).

Even when (**A1**) holds for the bridge in (53), uniqueness of the minimizer is not automatic because the map $\nu \mapsto q^\ell_{s|0,t}(\cdot|\nu, \mathbf{x}^\ell_t)$ may not be injective. For instance, the map is not injective for MDMs. Indeed, if $\mathbf{x}^\ell_t \neq \mathsf{m}$, then by (53), $q^\ell_{s|0,t}(\cdot|\nu, \mathbf{x}^\ell_t) = \mathrm{Cat}(\cdot; \mathbf{x}^\ell_t)$ for every ν in the simplex. Therefore the loss does not identify a unique value of ν on visible positions. In particular, the denoiser is also a minimizer.

A.2 Conversion Formula

In the next Proposition we provide the conversion formulas for arbitrary $\boldsymbol{\pi}$. Note that we are only interested in recovering $p^\ell_{0|t}(\mathbf{x}_0|\mathbf{x}_t)$ and $p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}^{-\ell}_t)$ for x_t satisfying $p_t(\mathbf{x}_t) > 0$ and $p_t(\mathbf{x}^{-\ell}_t) > 0$. Outside this support, the conditional densities may be defined arbitrarily.

Proposition 4. *It holds for any $\mathbf{x}_t$ such that $p_t(\mathbf{x}_t) > 0$,*

$$p^\ell_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}_t) = p_t(\mathbf{x}^{-\ell}_t) q^\ell_{t|0}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_0)\, p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}^{-\ell}_t)/p_t(\mathbf{x}_t)\ . \tag{23}$$

Conversely, suppose that $q^\ell_{t|0}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_0) > 0$ for any $\mathbf{x}_0$ and $\mathbf{x}_t$, it holds for any $\mathbf{x}^{-\ell}_t$, $p_t(\mathbf{x}^{-\ell}_t) > 0$,

$$p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}^{-\ell}_t) = \frac{p_t(\mathbf{x}_t) p^\ell_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}_t)}{p_t(\mathbf{x}^{-\ell}_t) q^\ell_{t|0}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_0)}\ . \tag{24}$$

Proof. By definition, for every $\ell \in [\![1, L]\!]$, every $\mathbf{x}_t \in \mathsf{X}$, and every $\mathbf{x}^\ell_0 \in \mathsf{V}$, we have

$$p^\ell_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}_t) = \frac{\sum_{\mathbf{x}^{-\ell}_0} p_0(\mathbf{x}_0) q^\ell_{t|0}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_0) \prod_{j\neq\ell} q^j_{t|0}(\mathbf{x}^j_t|\mathbf{x}^j_0)}{p_t(\mathbf{x}_t)}.$$

Since $q^\ell_{t|0}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_0)$ does not depend on $\mathbf{x}^{-\ell}_0$, this can be rewritten as

$$p^\ell_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}_t) = \frac{p_t(\mathbf{x}^{-\ell}_t) q^\ell_{t|0}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_0) p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}^{-\ell}_t)}{p_t(\mathbf{x}_t)} \propto q^\ell_{t|0}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_0)\, p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}^{-\ell}_t).$$

This is the conversion identity stated in Equation (10). In other words, the denoiser is obtained from the leave-one-out posterior by reinserting the local likelihood term $q^\ell_{t|0}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_0)$ associated with the observation at position ℓ.

Conversely, assume that for the considered $(t, \mathbf{x}_t, \ell)$, we have $q^\ell_{t|0}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_0) > 0$ for every $\mathbf{x}^\ell_0 \in \mathsf{V}$. Dividing the previous identity by $q^\ell_{t|0}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_0)$ and normalizing over $\mathbf{x}^\ell_0$ yields

$$p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}^{-\ell}_t) = \frac{p^\ell_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}_t)/q^\ell_{t|0}(\mathbf{x}^\ell_t|\mathbf{x}^\ell_0)}{\sum_{\tilde{\mathbf{x}}^\ell_0\in\mathsf{V}} p^\ell_{0|t}(\tilde{\mathbf{x}}^\ell_0|\mathbf{x}_t)/q^\ell_{t|0}(\mathbf{x}^\ell_t|\tilde{\mathbf{x}}^\ell_0)}\ . \tag{25}$$

□

Therefore, the conversion from the denoiser to the leave-one-out posterior is available exactly when the forward has full support. In UDM this condition is satisfied for every $t > 0$, so the two representations can always be converted into one another. In MDM, by contrast, the condition fails for unmasked positions. If $\mathbf{x}_t^\ell = \mathsf{m}$, the likelihood is constant in $\mathbf{x}_0^\ell$, so the two representations coincide. If $\mathbf{x}_t^\ell \neq \mathsf{m}$, the denoiser collapses to the Dirac mass at $\mathbf{x}_t^\ell$, and the leave-one-out posterior cannot be recovered from it.

For UDM, these same relations can be rewritten in matrix form. Since

$$q_{t|0}^\ell(\mathbf{x}_t^\ell|\mathbf{x}_0^\ell) = \alpha_t \langle \mathbf{x}_t^\ell, \mathbf{x}_0^\ell \rangle + \frac{1-\alpha_t}{K},$$

the first identity above yields

$$p_{0|t}^\ell(\cdot|\mathbf{x}_t) = \mathrm{Cat}\left(\frac{(1-\alpha_t)\hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)^\ell + K\alpha_t \langle \mathbf{x}_t^\ell, \hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)^\ell \rangle \mathbf{x}_t^\ell}{1-\alpha_t + K\alpha_t \langle \mathbf{x}_t^\ell, \hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)^\ell \rangle} \right) . \tag{26}$$

Conversely, the inverse relation can be written as

$$\hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)^\ell = \frac{(1+(K-1)\alpha_t)\,\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)^\ell - K\alpha_t \langle \mathbf{x}_t^\ell, \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)^\ell \rangle \mathbf{x}_t^\ell}{1+(K-1)\alpha_t - K\alpha_t \langle \mathbf{x}_t^\ell, \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)^\ell \rangle} . \tag{27}$$

Where $\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)^\ell = p_{0|t}^\ell(\cdot|\mathbf{x}_t)$.

For MDM, the conversion is explicit only on the support of the forward process. If $\mathbf{x}_t^\ell = \mathsf{m}$, then $q_{t|0}^\ell(\mathsf{m}|\mathbf{x}_0^\ell) = 1-\alpha_t$ for every $\mathbf{x}_0^\ell \in \mathsf{V}$, so

$$p_{0|t}^\ell(\cdot|\mathbf{x}_t) = \mathrm{Cat}(\hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)^\ell) .$$

If instead $\mathbf{x}_t^\ell \neq \mathsf{m}$, then $q_{t|0}^\ell(\mathbf{x}_t^\ell|\mathbf{x}_0^\ell) = \alpha_t \mathbb{1}\{\mathbf{x}_0^\ell = \mathbf{x}_t^\ell\}$, hence

$$p_{0|t}^\ell(\cdot|\mathbf{x}_t) = \mathrm{Cat}(\mathbf{x}_t^\ell) .$$

Therefore, on unmasked positions, the denoiser no longer contains enough information to reconstruct the leave-one-out posterior, so the inverse formula is not available.

Remark 1. *When these conversion formulas hold in both directions, as they do for UDM, the uniqueness of the denoiser as a minimizer of the loss is equivalent to the uniqueness of the leave-one-out denoiser as a minimizer of the loss.*

A.3 Score and leave-one-out denoiser

Recall that the score matrix $\mathbf{s}_t(\mathbf{x}_t) \in \mathbb{R}^{L\times K}$ is defined by

$$\langle \mathbf{y}^\ell, \mathbf{s}_t(\mathbf{x}_t)^\ell \rangle = p_t(\mathbf{y})/p_t(\mathbf{x}_t),$$

for every $\mathbf{y} \in \mathsf{X}$ such that $\mathbf{y}^{-\ell} = \mathbf{x}_t^{-\ell}$. This quantity appears naturally in the CTMC time-reversal; see Appendix F. By [2, Proposition 1], it is related to the denoiser through

$$\langle \mathbf{y}^\ell, \mathbf{s}_t(\mathbf{x}_t)^\ell \rangle = \sum_{\mathbf{x}_0^\ell} \frac{q_{t|0}^\ell(\mathbf{y}^\ell|\mathbf{x}_0^\ell)}{q_{t|0}^\ell(\mathbf{x}_t^\ell|\mathbf{x}_0^\ell)} p_{0|t}^\ell(\mathbf{x}_0^\ell|\mathbf{x}_t) . \tag{28}$$

This already shows that the score can be parameterized from the denoiser. The same quantity can also be expressed directly in terms of the leave-one-out denoiser:

$$\langle \mathbf{y}^\ell, \mathbf{s}_t(\mathbf{x}_t)^\ell \rangle = \frac{q_{t|0}^\ell(\mathbf{y}^\ell|\hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)^\ell)}{q_{t|0}^\ell(\mathbf{x}_t^\ell|\hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)^\ell)} , \tag{29}$$

for every $\mathbf{y} \in \mathsf{X}$ such that $\mathbf{y}^{-\ell} = \mathbf{x}_t^{-\ell}$.

Proof. By definition of the score, for every $\mathbf{y} \in \mathsf{X}$ such that $\mathbf{y}^{-\ell} = \mathbf{x}_t^{-\ell}$,

$$\langle \mathbf{y}^\ell, \mathbf{s}_t(\mathbf{x}_t)^\ell \rangle = \frac{p_t(\mathbf{y})}{p_t(\mathbf{x}_t)}.$$

Expanding the numerator using the factorization of the forward process gives

$$p_t(\mathbf{y}) = \sum_{\mathbf{x}_0} p_0(\mathbf{x}_0) q_{t|0}(\mathbf{y}|\mathbf{x}_0) = \sum_{\mathbf{x}_0} p_0(\mathbf{x}_0) q^\ell_{t|0}(\mathbf{y}^\ell|\mathbf{x}_0^\ell) \prod_{j \neq \ell} q^j_{t|0}(\mathbf{x}_t^j|\mathbf{x}_0^j).$$

Summing first over $\mathbf{x}_0^{-\ell}$, we obtain

$$p_t(\mathbf{y}) = \sum_{\mathbf{x}_0^\ell \in \mathsf{V}} q^\ell_{t|0}(\mathbf{y}^\ell|\mathbf{x}_0^\ell) \sum_{\mathbf{x}_0^{-\ell}} p_0(\mathbf{x}_0) \prod_{j \neq \ell} q^j_{t|0}(\mathbf{x}_t^j|\mathbf{x}_0^j).$$

Using the definition of the leave-one-out posterior, the inner sum can be written as

$$\sum_{\mathbf{x}_0^{-\ell}} p_0(\mathbf{x}_0) \prod_{j \neq \ell} q^j_{t|0}(\mathbf{x}_t^j|\mathbf{x}_0^j) = p_t(\mathbf{x}_t^{-\ell}) p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}_0^\ell|\mathbf{x}_t^{-\ell}).$$

Therefore,

$$p_t(\mathbf{y}) = p_t(\mathbf{x}_t^{-\ell}) \sum_{\mathbf{x}_0^\ell \in \mathsf{V}} q^\ell_{t|0}(\mathbf{y}^\ell|\mathbf{x}_0^\ell) p^{\mathrm{loo},\ell}_{0|t}(\mathbf{x}_0^\ell|\mathbf{x}_t^{-\ell}).$$

Since the map $\nu \mapsto q^\ell_{t|0}(\mathbf{y}^\ell|\nu)$ is affine in its first argument, this becomes

$$p_t(\mathbf{y}) = p_t(\mathbf{x}_t^{-\ell}) q^\ell_{t|0}(\mathbf{y}^\ell|\hat{\mathbf{x}}^{\mathrm{loo}}_{0|t}(\mathbf{x}_t)^\ell).$$

Taking $\mathbf{y} = \mathbf{x}_t$ yields in the same way

$$p_t(\mathbf{x}_t) = p_t(\mathbf{x}_t^{-\ell}) q^\ell_{t|0}(\mathbf{x}_t^\ell|\hat{\mathbf{x}}^{\mathrm{loo}}_{0|t}(\mathbf{x}_t)^\ell).$$

Dividing the two equations gives

$$\langle \mathbf{y}^\ell, \mathbf{s}_t(\mathbf{x}_t)^\ell \rangle = \frac{q^\ell_{t|0}(\mathbf{y}^\ell|\hat{\mathbf{x}}^{\mathrm{loo}}_{0|t}(\mathbf{x}_t)^\ell)}{q^\ell_{t|0}(\mathbf{x}_t^\ell|\hat{\mathbf{x}}^{\mathrm{loo}}_{0|t}(\mathbf{x}_t)^\ell)},$$

which is exactly (29). □

A.4 Categorical ratio matching

In this section we review *categorical ratio matching* as introduced in [34]. For $\mathbf{x}_t, \mathbf{y}$ that differ only at position ℓ, the concrete score used in the main text satisfies

$$\langle \mathbf{y}^\ell, \mathbf{s}_t(\mathbf{x}_t)^\ell \rangle = \frac{p_t(\mathbf{y})}{p_t(\mathbf{x}_t)} = \frac{p^\ell_t(\mathbf{y}^\ell|\mathbf{x}_t^{-\ell})}{p^\ell_t(\mathbf{x}_t^\ell|\mathbf{x}_t^{-\ell})}\,. \tag{30}$$

Thus it is enough to learn the one-position conditional marginal matrix $\boldsymbol{p}^*_t$ which is such that $\boldsymbol{p}^*_t(\mathbf{x}_t) \in \mathbb{R}^{L \times K}$ and $\langle x, \boldsymbol{p}^*_t(\mathbf{x}_t)^\ell \rangle = p^\ell_t(x|\mathbf{x}_t^{-\ell})$ since the score is recovered by probing this matrix at $\mathbf{y}^\ell$ and $\mathbf{x}_t^\ell$ and taking the ratio. In its population form, *categorical ratio matching* (CRM) trains a network $\boldsymbol{p}_\theta$ by minimizing

$$\mathcal{L}^{\mathrm{CRM}}_t(\theta) = \sum_{\ell=1}^{L} \mathbb{E}_{p_t}\left[\mathrm{KL}(p^\ell_t(\cdot|X_t^{-\ell}) \| \mathrm{Cat}(\boldsymbol{p}_\theta(X_t,t)^\ell))\right] \tag{31}$$

$$= -\sum_{\ell=1}^{L} \sum_{\mathbf{x}_t, \tilde{\mathbf{x}}_t^\ell} \log \langle \tilde{\mathbf{x}}_t^\ell, \boldsymbol{p}_\theta(\mathbf{x}_t,t)^\ell \rangle\, p^\ell_t(\tilde{\mathbf{x}}_t^\ell|\mathbf{x}_t^{-\ell}) p_t(\mathbf{x}_t) + C\,. \tag{32}$$

The minimizer of (31) is therefore $\boldsymbol{p}_\theta = \boldsymbol{p}^*_t$. This loss is however intractable as we cannot sample from $p^\ell_t(\cdot|\mathbf{x}_t^{-\ell})$ for any $\mathbf{x}_t^{-\ell}$. By assuming that the network $\boldsymbol{p}_\theta$ is leave-one-out; *i.e.* the output $\boldsymbol{p}_\theta(\mathbf{x}_t,t)^\ell$ is independent of the input $\mathbf{x}_t^\ell$, the previous loss then becomes equal, up to a constant, to

$$\widetilde{\mathcal{L}}^{\mathrm{CRM}}_t(\theta) = -\sum_{\ell=1}^{L} \mathbb{E}_{p_t}\left[\log \langle X_t^\ell, \boldsymbol{p}_\theta(X_t,t)^\ell \rangle\right]\,. \tag{33}$$

Without the leave-one-out assumption, this second loss is minimized when $\boldsymbol{p}_\theta(\mathbf{x}_t, t)^\ell = \mathbf{x}_t^\ell$.

[34] also considers a LOO denoiser parameterization which follows from Proposition 5. Therefore, if $\hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t)$ is a model for the leave-one-out denoiser, one may use the parameterization

$$\boldsymbol{p}_\theta(\mathbf{x}_t, t) = \alpha_t \hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t) + (1 - \alpha_t)\boldsymbol{\pi}^\ell \ . \tag{34}$$

The same restriction is still needed after this reparameterization. If $\hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t)^\ell$ may depend on $\mathbf{x}_t^\ell$, the minimizer of (33) maximizes the probability of $\mathbf{x}_t$ under $\mathrm{Cat}(\mathbf{p}_\theta(\mathbf{x}_t, t))$, hence at optimum $\hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t)^\ell = \mathbf{x}_t^\ell$ and $\boldsymbol{p}_\theta(\mathbf{x}_t, t)^\ell = \alpha_t \mathbf{x}_t^\ell + (1 - \alpha_t)\boldsymbol{\pi}^\ell$. If instead the admissible functions satisfy $\hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t)^\ell = \hat{\mathbf{x}}_0^\theta(\tilde{\mathbf{x}}_t, t)^\ell$ whenever $\mathbf{x}_t^{-\ell} = \tilde{\mathbf{x}}_t^{-\ell}$, then the unique minimizer is $\hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t) = \hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)$.

A.5 Conditional marginal in terms of the LOO denoiser

Proposition 5. *For every $\ell \in [\![1, L]\!]$, it holds*

$$p_t^\ell(\cdot|\mathbf{x}_t^{-\ell}) = \mathrm{Cat}\big(\alpha_t \hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)^\ell + (1 - \alpha_t)\boldsymbol{\pi}^\ell\big) \ . \tag{35}$$

Proof. Fix $\ell \in [\![1, L]\!]$ and let $\mathbf{y} \in \mathsf{X}$ satisfy $\mathbf{y}^{-\ell} = \mathbf{x}_t^{-\ell}$. Expanding the marginal of $\mathbf{y}$ and summing first over $\mathbf{x}_0^{-\ell}$ gives

$$p_t(\mathbf{y}) = \sum_{\mathbf{x}_0^\ell \in \mathsf{V}} q_{t|0}^\ell(\mathbf{y}^\ell|\mathbf{x}_0^\ell) \sum_{\mathbf{x}_0^{-\ell}} p_0(\mathbf{x}_0) \prod_{j \neq \ell} q_{t|0}^j(\mathbf{x}_t^j|\mathbf{x}_0^j).$$

By definition of the leave-one-out posterior,

$$\sum_{\mathbf{x}_0^{-\ell}} p_0(\mathbf{x}_0) \prod_{j \neq \ell} q_{t|0}^j(\mathbf{x}_t^j|\mathbf{x}_0^j) = p_t(\mathbf{x}_t^{-\ell}) p_{0|t}^{\mathrm{loo},\ell}(\mathbf{x}_0^\ell|\mathbf{x}_t^{-\ell}).$$

Therefore

$$p_t(\mathbf{y}) = p_t(\mathbf{x}_t^{-\ell}) \sum_{\mathbf{x}_0^\ell \in \mathsf{V}} q_{t|0}^\ell(\mathbf{y}^\ell|\mathbf{x}_0^\ell) p_{0|t}^{\mathrm{loo},\ell}(\mathbf{x}_0^\ell|\mathbf{x}_t^{-\ell}).$$

Using the affine extension of the forward kernel in its first argument, this becomes

$$p_t(\mathbf{y}) = p_t(\mathbf{x}_t^{-\ell}) q_{t|0}^\ell(\mathbf{y}^\ell|\hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)^\ell).$$

Dividing by $p_t(\mathbf{x}_t^{-\ell})$ yields

$$p_t^\ell(\mathbf{y}^\ell|\mathbf{x}_t^{-\ell}) = q_{t|0}^\ell(\mathbf{y}^\ell|\hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)^\ell).$$

Taking $\mathbf{y} = \mathbf{x}_t$ gives Proposition 5. □

A.6 An example of leave-one-out and denoiser differences

In order to confirm some intuition on the difference between the leave-one-out and the denoiser, we train a uniform diffusion model with the Bridge Plug-in parameterization that reaches a leave-one-out denoiser at optimality and a model with the Marginalization parameterization that recovers a denoiser. We then plot a heatmap corresponding to the probabilities output by the leave-one-out and the denoiser before passing them into their respective updates during sampling. We show the results in Figure 4.

In this case, the difference is mainly on the edges of the digits, where the denoiser is more confident than the leave-one-out posterior. This is because, on the edges of the digits, the local observation $\mathbf{x}_t^\ell$ is comparatively more informative about $\mathbf{x}_0^\ell$ than on the inner pixels where the value of $\mathbf{x}_0^\ell$ is already determined by its neighbors.

However, this difference varies with time, for large t, so at the start of sampling, (26) shows that the denoiser smoothes the prediction of the loo by convolving them with the marginal of the ℓ-th token. This blurring effect is observed in practice earlier in the sampling.

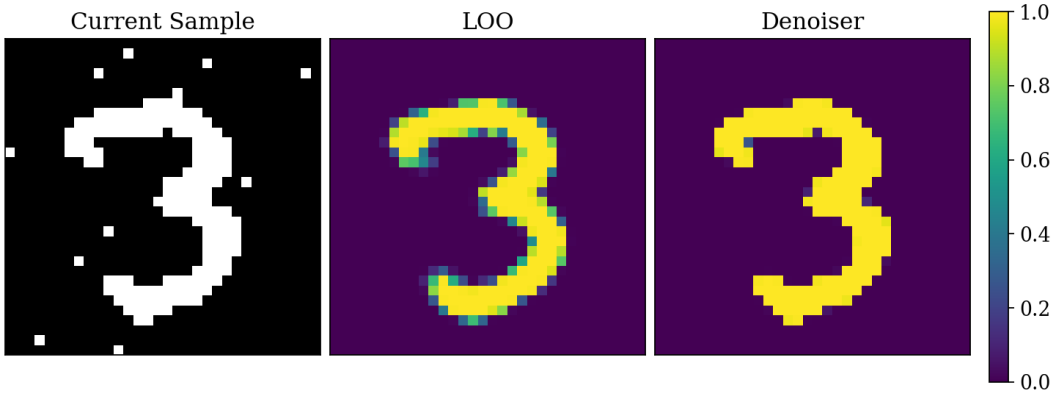


Figure 4: Comparison between the leave-one-out posterior $\hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)$ and the denoising posterior $\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)$.

Algorithm 2 AUDM sampler

Require: grid $0 = t_0 < \cdots < t_n = 1$
  Draw $U \sim \upsilon$
  Draw $X_{t_n} \sim p_{t_n}(\cdot|U)$
  for $i = n{-}1, \ldots, 0$ **do**
    $\hat{\mathbf{x}}_{0,i} \leftarrow \hat{\mathbf{x}}_0^\theta(X_{t_{i+1}}, t_{i+1}; U)$
    In parallel over ℓ:
      if $X_{t_{i+1}}^\ell \neq U^\ell$: $X_{t_i}^\ell \leftarrow X_{t_{i+1}}^\ell$
      else: $X_{t_i}^\ell \sim \mathrm{Cat}\Big(\frac{\alpha_{t_i} - \alpha_{t_{i+1}}}{1-\alpha_{t_{i+1}}} \hat{\mathbf{x}}_{0,i}^\ell + \frac{1-\alpha_{t_i}}{1-\alpha_{t_{i+1}}} U^\ell\Big)$
  end for
  return X_{t_0}

B Absorbing State Uniform Diffusion

B.1 AUDM NELBO

The following proposition gives the NELBO form for $-\log p_0^\theta$ defined in Section 4.1. See Appendix F for background on CTMCs. We write $\tilde{p}_0^\theta(\cdot|\mathbf{u})$ for the time-zero marginal of the CTMC reverse model conditioned on $U = \mathbf{u}$, so that $p_0^\theta(\mathbf{x}_0) = \sum_{\mathbf{u}} \upsilon(\mathbf{u}) \tilde{p}_0^\theta(\mathbf{x}_0|\mathbf{u})$.

Proposition 2. *A continuous-time NELBO for $-\log p_0^\theta(\mathbf{x}_0)$ is*

$$\mathrm{L}_\infty^{\mathrm{AUDM}}(\mathbf{x}_0;\theta) := -\int_0^1 \frac{\alpha_t'}{1-\alpha_t}\, \mathbb{E}\Big[\sum_{\ell=1}^L \mathbb{1}_{X_t^\ell = U^\ell} \big\{1 - \langle U^\ell, \hat{\mathbf{x}}_0^\theta(X_t, t; U)^\ell\rangle - \mathbb{1}_{\mathbf{x}_0^\ell \neq U^\ell} \cdot \big(1 + \log \langle \mathbf{x}_0^\ell, \hat{\mathbf{x}}_0^\theta(X_t, t; U)^\ell\rangle\big)\big\}\Big] \mathrm{d}t, \quad (36)$$

where the expectation is w.r.t. $U \sim$ Uniform$([\![1, K]\!])^{\otimes L}$, and $X_t \sim q_{t|0}(\cdot|\mathbf{x}_0, U)$.

Proof. For every $\mathbf{x}_0$, Jensen's inequality gives

$$-\log p_0^\theta(\mathbf{x}_0) = -\log \textstyle\sum_{\mathbf{u}} \upsilon(\mathbf{u}) p_0^\theta(\mathbf{x}_0|\mathbf{u}) \leq \sum_{\mathbf{u}} \upsilon(\mathbf{u}) \big[-\log p_0^\theta(\mathbf{x}_0|\mathbf{u})\big].$$

It is therefore enough to upper bound each conditional negative log-likelihood by the CTMC ELBO (63), with $\pi^\ell = U^\ell$, and then average over U.

Since the prefactor is $\langle X_t^\ell, U^\ell\rangle$, only the position ℓ such that $X_t^\ell = U^\ell$ contribute to the loss. Let ℓ denote such a position and let $\tilde{x} \neq U^\ell$. We first compute the two arguments of Φ in (63). By

definition of the conditional score (60),

$$\langle \tilde{x}, s_t(U^\ell | X_0^\ell) \rangle = \frac{\langle \tilde{x}, \alpha_t X_0^\ell + (1-\alpha_t) U^\ell \rangle}{q_{t|0}(U^\ell | X_0^\ell, U^\ell)} = \frac{\alpha_t}{1-\alpha_t} \mathbb{1}_{\tilde{x}=X_0^\ell} \mathbb{1}_{X_0^\ell \neq U^\ell} .$$

Next, by the definition (62) and since $X_t^\ell = U^\ell$, we have for every $\tilde{x} \neq U^\ell$,

$$\langle \tilde{x}, \mathbf{s}_t^\theta(X_t)^\ell \rangle = \frac{\alpha_t}{1-\alpha_t} \langle \tilde{x}, \hat{\mathbf{x}}_0^\theta(X_t, t; U)^\ell \rangle .$$

Substituting both expressions into (63), the contribution of position ℓ at time t becomes

$$\beta_t \mathbb{1}\{X_t^\ell = U^\ell\} \sum_{\tilde{x} \neq U^\ell} \Phi\left(\frac{\alpha_t}{1-\alpha_t} \mathbb{1}_{\tilde{x}=X_0^\ell} \mathbb{1}_{X_0^\ell \neq U^\ell}, \frac{\alpha_t}{1-\alpha_t} \langle \tilde{x}, \hat{\mathbf{x}}_0^\theta(X_t, t; U)^\ell \rangle \right) .$$

If $X_0^\ell = U^\ell$, then the first argument of Φ is zero for every $\tilde{x} \neq U^\ell$, so the sum reduces to

$$\beta_t \frac{\alpha_t}{1-\alpha_t} \mathbb{1}_{X_t^\ell = U^\ell} \sum_{\tilde{x} \neq U^\ell} \langle \tilde{x}, \hat{\mathbf{x}}_0^\theta(X_t, t; U)^\ell \rangle = \frac{\alpha_t'}{1-\alpha_t} \mathbb{1}_{X_t^\ell = U^\ell} \big(1 - \langle U^\ell, \hat{\mathbf{x}}_0^\theta(X_t, t; U)^\ell \rangle\big).$$

If $X_0^\ell \neq U^\ell$, only the term $\tilde{x} = X_0^\ell$ has a nonzero first argument, and therefore

$$\begin{aligned}
&\beta_t \mathbb{1}\{X_t^\ell = U^\ell\} \sum_{\tilde{x} \neq U^\ell} \Phi\left(\frac{\alpha_t}{1-\alpha_t} \mathbb{1}_{\tilde{x}=X_0^\ell}, \frac{\alpha_t}{1-\alpha_t} \langle \tilde{x}, \hat{\mathbf{x}}_0^\theta(X_t, t; U)^\ell \rangle \right) \\
&\quad = \frac{-\alpha_t'}{1-\alpha_t} \mathbb{1}_{X_t^\ell = U^\ell} \left(\sum_{\tilde{x} \neq U^\ell} \langle \tilde{x}, \hat{\mathbf{x}}_0^\theta(X_t, t; U)^\ell \rangle - 1 - \log \langle X_0^\ell, \hat{\mathbf{x}}_0^\theta(X_t, t; U)^\ell \rangle \right) \\
&\quad = \frac{-\alpha_t'}{1-\alpha_t} \mathbb{1}_{X_t^\ell = U^\ell} \left(1 - \langle U^\ell, \hat{\mathbf{x}}_0^\theta(X_t, t; U)^\ell \rangle - 1 - \log \langle X_0^\ell, \hat{\mathbf{x}}_0^\theta(X_t, t; U)^\ell \rangle \right) .
\end{aligned}$$

Combining the two cases gives exactly the integrand in (36). □

B.2 Resampled AUDM

In the rest of this section we write $\mathbf{z} = (\mathbf{x}, \mathbf{u})$. The UDM denoiser admits the mixture decomposition

$$p_{0|t}(\mathbf{x}_0 | \mathbf{x}_t) = \textstyle\sum_{\mathbf{u}_t} p_{0|t}(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{u}_t)\, \upsilon_t(\mathbf{u}_t | \mathbf{x}_t) , \qquad \upsilon_t(\mathbf{u}_t | \mathbf{x}_t) \propto \upsilon(\mathbf{u}_t) p_t(\mathbf{x}_t | \mathbf{u}_t) . \tag{37}$$

A first exact lifted sampler is obtained by augmenting the state as $\tilde{Z}_t = (X_t, \tilde{U}_t)$ with conditional law $\tilde{U}_t \sim \upsilon_t(\cdot | X_t)$. The conditional law of $\tilde{U}_t$ belongs to the current lifted marginal, so the backward transition can be written as

$$\tilde{p}_{s|t}(\mathbf{z}_s | \mathbf{z}_t) \coloneqq \upsilon_s(\mathbf{u}_s | \mathbf{x}_s) \textstyle\sum_{\mathbf{x}_0} q_{s|0,t}(\mathbf{x}_s | \mathbf{x}_0, \mathbf{x}_t) p_{0|t}(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{u}_t) .$$

Using (37), this transition satisfies the one-step marginalization identity

$$\textstyle\sum_{\mathbf{u}_t} \tilde{p}_{s|t}(\mathbf{z}_s | \mathbf{z}_t) \upsilon_t(\mathbf{u}_t | \mathbf{x}_t) = p_{s|t}(\mathbf{x}_s | \mathbf{x}_t) \upsilon_s(\mathbf{u}_s | \mathbf{x}_s) .$$

This is the same propagation identity as in Lemma 1 that we show next: after additionally summing over $\mathbf{u}_s$, the remaining transition is the UDM reverse transition. Iterating this identity over the time grid shows that the ideal lifted chain preserves the UDM marginal path law. It is not directly useful, however, because drawing $\mathbf{u}_s$ from $\upsilon_s(\cdot | \mathbf{x}_s)$ is intractable.

The resampled AUDM sampler in Algorithm 1 avoids this step by resampling the absorbing state after drawing (X_0, X_s). It is the Markov chain on $Z_t = (X_t, U_t)$ whose one-step transition is

$$\bar{p}_{s|t}(\mathbf{z}_s | \mathbf{z}_t) = \sum_{\mathbf{x}_0} \upsilon_{s|0}(\mathbf{u}_s | \mathbf{x}_0, \mathbf{x}_s) q_{s|0,t}(\mathbf{x}_s | \mathbf{x}_0, \mathbf{x}_t)\, p_{0|t}(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{u}_t) .$$

This equation is exactly the inline algorithm in the main text: given $(\mathbf{x}_t, \mathbf{u}_t)$, first draw $\mathbf{x}_0$ from the absorbing-state denoiser, then draw $\mathbf{x}_s$ from the UDM bridge, and finally resample the absorbing state conditionally on $(\mathbf{x}_0, \mathbf{x}_s)$. The last step is tractable because

$$\upsilon_{s|0}(\mathbf{u}_s | \mathbf{x}_0, \mathbf{x}_s) = \frac{\upsilon(\mathbf{u}_s) q_{s|0}(\mathbf{x}_s | \mathbf{x}_0, \mathbf{u}_s)}{q_{s|0}(\mathbf{x}_s | \mathbf{x}_0)} . \tag{38}$$

The exact lifted generative model is

$$\bar{p}_{0:n}(\mathbf{z}_{t_0:t_n}) = \bar{p}_{t_n}(\mathbf{z}_{t_n}) \prod_{i=1}^{n} \bar{p}_{s_i|t_i}(\mathbf{z}_{s_i}|\mathbf{z}_{t_i}) \;, \quad \bar{p}_{t_n}(\mathbf{z}_{t_n}) \coloneqq \upsilon_{t_n}(\mathbf{u}_{t_n}|\mathbf{x}_{t_n}) p_{t_n}(\mathbf{x}_{t_n}) \tag{39}$$

with $\bar{p}_{0|t_1}(\cdot|\mathbf{z}_{t_1}) = \delta_{\mathbf{u}_{t_1}} \otimes p_{0|t_1}(\cdot|\mathbf{x}_{t_1}, \mathbf{u}_{t_1})$. The next lemma states the one-step marginalization identity. Iterating it over the grid gives Proposition 6, hence the trajectory produced by the sampler has the same marginal law on $(X_{t_0}, \dots, X_{t_n})$ as UDM.

Lemma 1. *For any $0 < s < t \leq 1$,*

$$\sum_{\mathbf{u}_t} \bar{p}_{s|t}(\mathbf{z}_s|\mathbf{z}_t)\, \upsilon_t(\mathbf{u}_t|\mathbf{x}_t) = p_{s|t}(\mathbf{x}_s|\mathbf{x}_t) \upsilon_s(\mathbf{u}_s|\mathbf{x}_s) \;.$$

Proof. We have that

$$\begin{aligned}
\sum_{\mathbf{u}_t} \bar{p}_{s|t}(\mathbf{z}_s|\mathbf{z}_t)\, \upsilon_t(\mathbf{u}_t|\mathbf{x}_t) &= \sum_{\mathbf{x}_0} \upsilon_{s|0}(\mathbf{u}_s|\mathbf{x}_0, \mathbf{x}_s) q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{x}_t) \sum_{\mathbf{u}_t} p_{0|t}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{u}_t) \upsilon_t(\mathbf{u}_t|\mathbf{x}_t) \\
&= \sum_{\mathbf{x}_0} \upsilon_{s|0}(\mathbf{u}_s|\mathbf{x}_0, \mathbf{x}_s) q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{x}_t)\, p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) \;.
\end{aligned}$$

where we have used the identity (37) in the second line. Next, using the definition $p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) = p_0(\mathbf{x}_0) q_{t|0}(\mathbf{x}_t|\mathbf{x}_0) / p_t(\mathbf{x}_t)$ and (38), we get

$$\begin{aligned}
\sum_{\mathbf{u}_t} \bar{p}_{s|t}(\mathbf{z}_s|\mathbf{z}_t)\, \upsilon_t(\mathbf{u}_t|\mathbf{x}_t) &= \frac{\upsilon(\mathbf{u}_s)}{p_t(\mathbf{x}_t)} \sum_{\mathbf{x}_0} \frac{q_{s|0}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{u}_s)}{q_{s|0}(\mathbf{x}_s|\mathbf{x}_0)} q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{x}_t) q_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\, p_0(\mathbf{x}_0) \\
&= \frac{\upsilon(\mathbf{u}_s)}{p_t(\mathbf{x}_t)} q_{t|s}(\mathbf{x}_t|\mathbf{x}_s) \sum_{\mathbf{x}_0} q_{s|0}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{u}_s)\, p_0(\mathbf{x}_0) \\
&= \frac{\upsilon(\mathbf{u}_s)}{p_t(\mathbf{x}_t)} q_{t|s}(\mathbf{x}_t|\mathbf{x}_s) p_s(\mathbf{x}_s|\mathbf{u}_s) \\
&= \frac{\upsilon(\mathbf{u}_s) p_s(\mathbf{x}_s|\mathbf{u}_s)}{p_s(\mathbf{x}_s)} \frac{p_s(\mathbf{x}_s) q_{t|s}(\mathbf{x}_t|\mathbf{x}_s)}{p_t(\mathbf{x}_t)}
\end{aligned}$$

where in the second line we have used the bridge identity $q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{x}_t) q_{t|0}(\mathbf{x}_t|\mathbf{x}_0) = q_{s|0}(\mathbf{x}_s|\mathbf{x}_0) q_{t|s}(\mathbf{x}_t|\mathbf{x}_s)$. □

Proposition 6. *It holds that*

$$\sum_{\mathbf{u}_{t_0:t_n}} \bar{p}_{0:n}(\mathbf{z}_{t_0:t_n}) \coloneqq p_{t_n}(\mathbf{x}_{t_n}) \prod_{i=1}^{n} p_{s_i|t_i}(\mathbf{x}_{s_i}|\mathbf{x}_{t_i}) \;. \tag{40}$$

Proof. We define $\bar{p}_{0:k|k+1}(\mathbf{z}_{t_0:t_k}|\mathbf{z}_{t_{k+1}}) \coloneqq \prod_{i=1}^{k+1} \bar{p}_{s_i|t_i}(\mathbf{z}_{s_i}|\mathbf{z}_{t_i})$ and proceed by induction.

First, by the definition (39) and Lemma 1 we have that

$$\begin{aligned}
\sum_{\mathbf{u}_{t_0:t_n}} \bar{p}_{0:n}(\mathbf{z}_{t_0:t_n}) &= \sum_{\mathbf{u}_{t_0:t_n}} p_{t_n}(\mathbf{x}_{t_n}) \upsilon_{t_n}(\mathbf{u}_{t_n}|\mathbf{x}_{t_n}) \bar{p}_{s_n|t_n}(\mathbf{z}_{s_n}|\mathbf{z}_{t_n}) \bar{p}_{0:n-2|n-1}(\mathbf{z}_{t_0:t_{n-1}}|\mathbf{z}_{s_n}) \\
&= \sum_{\mathbf{u}_{t_0:t_{n-1}}} p_{t_n}(\mathbf{x}_{t_n}) p_{s_n|t_n}(\mathbf{x}_{s_n}|\mathbf{x}_{t_n}) \\
&\qquad \times \upsilon_{s_n}(\mathbf{u}_{s_n}|\mathbf{x}_{s_n}) \bar{p}_{0:n-2|n-1}(\mathbf{z}_{t_0:t_{n-1}}|\mathbf{z}_{s_n}) \;.
\end{aligned}$$

Next, assuming that for $k \in [\![2, n-1]\!]$,

$$\sum_{\mathbf{u}_{t_0:t_n}} \bar{p}_{0:n}(\mathbf{z}_{t_0:t_n}) = \sum_{\mathbf{u}_{t_0:t_k}} p_{t_n}(\mathbf{x}_{t_n}) \prod_{i=k+1}^{n} p_{s_i|t_i}(\mathbf{x}_{s_i}|\mathbf{x}_{t_i}) \upsilon_{t_k}(\mathbf{u}_{t_k}|\mathbf{x}_{t_k}) \bar{p}_{0:k-1|k}(\mathbf{z}_{t_0:t_{k-1}}|\mathbf{z}_{t_k})$$

we get by applying Lemma 1 again,

$$\begin{aligned}
\sum_{\mathbf{u}_{t_0:t_n}} \bar{p}_{0:n}(\mathbf{z}_{t_0:t_n}) &= \sum_{\mathbf{u}_{t_0:t_{k-1}}} p_{t_n}(\mathbf{x}_{t_n}) \prod_{i=k+1}^{n} p_{s_i|t_i}(\mathbf{x}_{s_i}|\mathbf{x}_{t_i}) \\
&\qquad \times \sum_{\mathbf{u}_{t_k}} \upsilon_{t_k}(\mathbf{u}_{t_k}|\mathbf{x}_{t_k}) \bar{p}_{s_k|t_k}(\mathbf{z}_{s_k}|\mathbf{z}_{t_k}) \bar{p}_{0:k-2|k-1}(\mathbf{z}_{t_0:t_{k-2}}|\mathbf{z}_{s_k}) \\
&= \sum_{\mathbf{u}_{t_0:t_{k-1}}} p_{t_n}(\mathbf{x}_{t_n}) \prod_{i=k+1}^{n} p_{s_i|t_i}(\mathbf{x}_{s_i}|\mathbf{x}_{t_i}) \\
&\qquad \times p_{t_{k-1}|t_k}(\mathbf{x}_{t_{k-1}}|\mathbf{x}_{t_k}) \upsilon_{t_{k-1}}(\mathbf{u}_{t_{k-1}}|\mathbf{x}_{t_{k-1}}) \bar{p}_{0:k-2|k-1}(\mathbf{z}_{t_0:t_{k-2}}|\mathbf{z}_{t_{k-1}}) \\
&= \sum_{\mathbf{u}_{t_0:t_{k-1}}} p_{t_n}(\mathbf{x}_{t_n}) \prod_{i=k}^{n} p_{s_i|t_i}(\mathbf{x}_{s_i}|\mathbf{x}_{t_i}) \\
&\qquad \times \upsilon_{t_{k-1}}(\mathbf{u}_{t_{k-1}}|\mathbf{x}_{t_{k-1}}) \bar{p}_{0:k-2|k-1}(\mathbf{z}_{t_0:t_{k-2}}|\mathbf{z}_{t_{k-1}})
\end{aligned}$$

which finishes the induction. We hence get for $k = 2$, upon using the identity (37),

$$\begin{aligned}
\sum_{\mathbf{u}_{t_0:t_n}} \bar{p}_{0:n}(\mathbf{z}_{t_0:t_n}) &= \sum_{\mathbf{u}_{t_0},\mathbf{u}_{t_1}} p_{t_n}(\mathbf{x}_{t_n}) \prod_{i=2}^{n} p_{s_i|t_i}(\mathbf{x}_{s_i}|\mathbf{x}_{t_i}) \upsilon_{t_1}(\mathbf{u}_{t_1}|\mathbf{x}_{t_1}) \bar{p}_{0|t_1}(\mathbf{z}_{t_0}|\mathbf{z}_{t_1}) \\
&= p_{t_n}(\mathbf{x}_{t_n}) \prod_{i=2}^{n} p_{s_i|t_i}(\mathbf{x}_{s_i}|\mathbf{x}_{t_i}) \sum_{\mathbf{u}_{t_1}} \upsilon_{t_1}(\mathbf{u}_{t_1}|\mathbf{x}_{t_1}) p_{0|t_1}(\mathbf{x}_0|\mathbf{x}_{t_1}, \mathbf{u}_{t_1}) \\
&= p_{t_n}(\mathbf{x}_{t_n}) \prod_{i=1}^{n} p_{s_i|t_i}(\mathbf{x}_{s_i}|\mathbf{x}_{t_i}) \ ,
\end{aligned}$$

which gives the result. □

The exact lifted model above is useful conceptually, but it still contains the intractable denoiser $p_{0|t}(\cdot|\mathbf{x}_t, \mathbf{u}_t)$. Its true one-position reverse transition is obtained by marginalizing the exact lifted kernel over all other positions. Since the bridge and the resampling law are token-wise conditionally on $\mathbf{x}_0$, this gives

$$\bar{p}^{\ell}_{s|t}(\mathbf{x}^{\ell}_s, \mathbf{u}^{\ell}_s|\mathbf{z}_t) \coloneqq \textstyle\sum_{\mathbf{z}_s^{-\ell}} \bar{p}_{s|t}(\mathbf{z}_s|\mathbf{z}_t) = \sum_{\mathbf{x}_0^{\ell}} \upsilon_{s|0}(\mathbf{u}^{\ell}_s|\mathbf{x}^{\ell}_0, \mathbf{x}^{\ell}_s) q^{\ell}_{s|0,t}(\mathbf{x}^{\ell}_s|\mathbf{x}^{\ell}_0, \mathbf{x}^{\ell}_t)\, p^{\ell}_{0|t}(\mathbf{x}^{\ell}_0|\mathbf{x}_t, \mathbf{u}_t),$$

where $p^{\ell}_{0|t}(\mathbf{x}^{\ell}_0|\mathbf{x}_t, \mathbf{u}_t) \coloneqq \sum_{\mathbf{x}_0^{-\ell}} p_{0|t}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{u}_t)$. ReAUDM replaces this exact one-position posterior marginal with the factorized approximation induced by (17),

$$p^{\theta}_{0|t}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{u}_t) \coloneqq \textstyle\prod_{\ell=1}^{L} \mathrm{Cat}\big(\mathbf{x}^{\ell}_0; \hat{\mathbf{x}}^{\theta}_0(\mathbf{x}_t, t; \mathbf{u}_t)^{\ell}\big) \ ,$$

while keeping the UDM bridge and the resampling law (38) exact. This gives the parameterized token-wise reverse transition

$$\bar{p}^{\theta,\ell}_{s|t}(\mathbf{x}^{\ell}_s, \mathbf{u}^{\ell}_s|\mathbf{z}_t) \coloneqq \textstyle\sum_{\mathbf{x}_0^{\ell}} \upsilon_{s|0}(\mathbf{u}^{\ell}_s|\mathbf{x}^{\ell}_0, \mathbf{x}^{\ell}_s) q^{\ell}_{s|0,t}(\mathbf{x}^{\ell}_s|\mathbf{x}^{\ell}_0, \mathbf{x}^{\ell}_t)\, \mathrm{Cat}\big(\mathbf{x}^{\ell}_0; \hat{\mathbf{x}}^{\theta}_0(\mathbf{x}_t, t; \mathbf{u}_t)^{\ell}\big) \ ,$$

and therefore the full transition factorizes as

$$\bar{p}^{\theta}_{s|t}(\mathbf{z}_s|\mathbf{z}_t) \coloneqq \textstyle\prod_{\ell=1}^{L} \bar{p}^{\theta,\ell}_{s|t}(\mathbf{x}^{\ell}_s, \mathbf{u}^{\ell}_s|\mathbf{z}_t).$$

Following the same argument as in the main paper, this factorized transition is optimal when the factorized denoiser matches the one-position marginals of the exact noise-conditioned posterior. Indeed, the associated joint model is

$$\bar{p}^{\theta}_{0:n}(\mathbf{z}_{t_0:t_n}) = \bar{p}_{t_n}(\mathbf{z}_{t_n}) \textstyle\prod_{i=1}^{n} \bar{p}^{\theta}_{s_i|t_i}(\mathbf{z}_{s_i}|\mathbf{z}_{t_i}).$$

As in the main paper, its discrete time NELBO decomposes into KL divergences between backward transitions:

$$\begin{aligned}
&\mathcal{L}^{\mathrm{ReAUDM}}_n(\theta; \mathbf{x}_0) \\
&\qquad = \mathbb{E}\left[-\log p^{\theta}_{0|t_1}(\mathbf{x}_0|X_{t_1}, U_{t_1}) + \sum_{i=1}^{n} \mathrm{KL}(\bar{p}_{s_i|t_i}(\cdot|Z_{t_i}) \| \bar{p}^{\theta}_{s_i|t_i}(\cdot|Z_{t_i})) \,\middle|\, X_0 = \mathbf{x}_0 \right] . \quad (41)
\end{aligned}$$

Since $\bar{p}_{s|t}$ and $\bar{p}^{\theta}_{s|t}$ have the same backward decomposition except for the absorbing-state denoiser, these KL terms are minimized by matching $p_{0|t}(\cdot|\mathbf{x}_t, \mathbf{u}_t)$ with its factorized approximation. Equivalently, the optimal factorized approximation is obtained by matching the noise-conditioned posterior marginals at every position, namely $\hat{\mathbf{x}}^{\theta}_0(\mathbf{x}_t, t; \mathbf{u}_t)^{\ell} = \mathbb{E}[X_0^{\ell} \mid X_t = \mathbf{x}_t, U_t = \mathbf{u}_t]$, with the carry-over constraint in (17). In particular, positions satisfying $\mathbf{x}_t^{\ell} \neq \mathbf{u}_t^{\ell}$ are already optimal by setting $\hat{\mathbf{x}}^{\theta}_0(\mathbf{x}_t, t; \mathbf{u}_t)^{\ell} = \mathbf{x}_t^{\ell}$. The corresponding sampler is:

Algorithm 3 ReAUDM sampler

Require: grid $0 = t_0 < \cdots < t_n = 1$
  Draw $U_{t_n} \sim \upsilon_{t_n}$ and $X_{t_n} = U_{t_n}$
  for $i = n{-}1, \ldots, 0$ **do**
    Draw $X_0 \sim p^{\theta}_{0|t_{i+1}}(\cdot|X_{t_{i+1}}, U_{t_{i+1}})$
    Draw $X_{t_i} \sim q_{t_i|0,t_{i+1}}(\cdot|X_0, X_{t_{i+1}})$
    In parallel over ℓ:
      if $X^{\ell}_{t_i} \neq X^{\ell}_0$: $U^{\ell}_{t_i} \leftarrow X^{\ell}_{t_i}$
      else: $U^{\ell}_{t_i} \sim \mathrm{Cat}\Big(\frac{X^{\ell}_0 + \alpha_{t_i}(\mathbf{1} - X^{\ell}_0)}{1 + (K-1)\alpha_{t_i}}\Big)$
  end for
  return X_{t_0}

B.3 Remarks on some related constructions

Planned denoising [17]. We now relate AUDM to the planned-denoising parameterization of [17]. The connection is that both constructions start from the same UDM denoiser, but they expose different latent variables behind it: AUDM keeps the full absorbing state U, while planned denoising keeps only the binary mask indicating whether a position has been corrupted. This mask decomposition leads naturally to a planner-denoiser factorization as we now see.

We may also write an analogous identity to (37) using a binary noise mask rather than with the absorbing state U. Let $N_t \in \{0,1\}^L$ with p.m.f.

$$\nu_t(\mathbf{n}_t) \coloneqq \prod_{\ell=1}^{L} \mathrm{Bern}(\mathbf{n}_t^{\ell}; 1 - \alpha_t) \; .$$

Then the UDM forward transition admits the augmented representation

$$q_{t|0}(\mathbf{x}_t|\mathbf{x}_0) \coloneqq \sum_{\mathbf{n}_t} \underbrace{\prod_{\ell=1}^{L} \mathrm{Cat}\big(\mathbf{x}_t^{\ell}; (1 - \mathbf{n}_t^{\ell})\mathbf{x}_0^{\ell} + \mathbf{n}_t^{\ell}\mathbf{1}/K\big)}_{\coloneqq q_{t|0}(\mathbf{x}_t, \mathbf{n}_t|\mathbf{x}_0)} \; \nu_t(\mathbf{n}_t) \; . \tag{42}$$

We can thus get an equivalent of the identity (37):

$$p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) \coloneqq \sum_{\mathbf{n}_t} p_{0|t}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{n}_t)\, \nu_t(\mathbf{n}_t|\mathbf{x}_t) \; ,$$

where

$$\nu_t(\mathbf{n}_t|\mathbf{x}_t) \coloneqq \frac{\sum_{\tilde{\mathbf{x}}_0} p_0(\tilde{\mathbf{x}}_0) q_{t|0}(\mathbf{x}_t, \mathbf{n}_t|\tilde{\mathbf{x}}_0)}{p_t(\mathbf{x}_t)}$$

and

$$p_{0|t}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{n}_t) \coloneqq \frac{p_0(\mathbf{x}_0) q_{t|0}(\mathbf{x}_t, \mathbf{n}_t|\mathbf{x}_0)}{\sum_{\tilde{\mathbf{x}}_0} p_0(\tilde{\mathbf{x}}_0) q_{t|0}(\mathbf{x}_t, \mathbf{n}_t|\tilde{\mathbf{x}}_0)} = \frac{p_0(\mathbf{x}_0) \prod_{\ell:\mathbf{n}_t^{\ell}=0} \mathbb{1}_{\mathbf{x}_0^{\ell} = \mathbf{x}_t^{\ell}}}{\sum_{\tilde{\mathbf{x}}_0} p_0(\tilde{\mathbf{x}}_0) \prod_{\ell:\mathbf{n}_t^{\ell}=0} \mathbb{1}_{\tilde{\mathbf{x}}_0^{\ell} = \mathbf{x}_t^{\ell}}} \; . \tag{43}$$

The mask-conditioned posterior has thus the carry-over unmasking property and, in contrast with (16), it is time-independent. Conditioning on $\mathbf{n}_t$ instead removes the ambiguity that remains when one only conditions on U and observes $\mathbf{x}_t^{\ell} = U^{\ell}$.

We now focus on the factorized denoiser $\prod_{\ell=1}^{L} p^{\ell}_{0|t}(\cdot|\mathbf{x}_t)$. The ℓ-th posterior marginal $p^{\ell}_{0|t}(\cdot|\mathbf{x}_t)$ is obtained by marginalizing only the ℓ-th mask variable:

$$p^{\ell}_{0|t}(\mathbf{x}_0^{\ell}|\mathbf{x}_t) = \textstyle\sum_{\mathbf{n}_t^{\ell}} p^{\ell}_{0|t}(\mathbf{x}_0^{\ell}|\mathbf{x}_t, \mathbf{n}_t^{\ell})\, \nu_t^{\ell}(\mathbf{n}_t^{\ell}|\mathbf{x}_t) \tag{44}$$

where

$$\nu_t^\ell(\mathbf{n}_t^\ell|\mathbf{x}_t) \coloneqq \textstyle\sum_{\mathbf{n}_t^{-\ell}} \nu_t(\mathbf{n}_t|\mathbf{x}_t) \;, \quad \mathbf{n}_t^\ell \in \{0,1\}$$

and, whenever $\nu_t^\ell(\mathbf{n}_t^\ell|\mathbf{x}_t) > 0$,

$$\nu_t^{-\ell}(\mathbf{n}_t^{-\ell}|\mathbf{x}_t, \mathbf{n}_t^\ell) \coloneqq \nu_t(\mathbf{n}_t|\mathbf{x}_t) / \nu_t^\ell(\mathbf{n}_t^\ell|\mathbf{x}_t) \;.$$

and finally the token-wise mask conditional posterior in (44) is

$$p_{0|t}^\ell(\mathbf{x}_0^\ell|\mathbf{x}_t, \mathbf{n}_t^\ell) \coloneqq \textstyle\sum_{\mathbf{n}_t^{-\ell}} \sum_{\mathbf{x}_0^{-\ell}} p_{0|t}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{n}_t)\, \nu_t^{-\ell}(\mathbf{n}_t^{-\ell}|\mathbf{x}_t, \mathbf{n}_t^\ell) \;.$$

It follows from the carry-over property above that

$$p_{0|t}^\ell(\mathbf{x}_0|\mathbf{x}_t, \mathbf{n}_t^\ell = 0) = \mathrm{Cat}(\mathbf{x}_0; \mathbf{x}_t^\ell) \;.$$

Thus the factorized denoiser has the two-component form

$$p_{0|t}^\ell(\mathbf{x}_0|\mathbf{x}_t) = \mathrm{Cat}(\mathbf{x}_0^\ell; (1 - \boldsymbol{a}(\mathbf{x}_t, t)^\ell)\mathbf{x}_t^\ell + \boldsymbol{a}(\mathbf{x}_t, t)^\ell \boldsymbol{d}(\mathbf{x}_t, t)^\ell) \;, \tag{45}$$

where the exact *planner* is

$$\boldsymbol{a}(\mathbf{x}_t, t)^\ell \coloneqq \nu_t^\ell(\mathbf{n}_t^\ell = 1|\mathbf{x}_t) \;,$$

and the corresponding one-position denoiser is the probability vector such that

$$\mathrm{Cat}(\mathbf{x}_0^\ell; \boldsymbol{d}(\mathbf{x}_t, t)^\ell) = p_{0|t}^\ell(\mathbf{x}_0^\ell|\mathbf{x}_t, \mathbf{n}_t^\ell = 1) \;.$$

The posterior conditioned on the full mask is time-independent, but the practical factorized denoiser is not: both the planner and the one-position denoiser are functions of $\mathbf{x}_t$ and t, because they marginalize over the posterior law of the unobserved mask variables. Thus planned denoising parameterizes

$$\hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t)^\ell \coloneqq \big(1 - \boldsymbol{a}_\theta(\mathbf{x}_t, t)^\ell\big)\mathbf{x}_t^\ell + \boldsymbol{a}_\theta(\mathbf{x}_t, t)^\ell \boldsymbol{d}_\theta(\mathbf{x}_t, t)^\ell \;, \tag{46}$$

where $\boldsymbol{a}_\theta(\mathbf{x}_t, t)^\ell$ approximates $\boldsymbol{a}(\mathbf{x}_t, t)^\ell$ and $\boldsymbol{d}_\theta(\mathbf{x}_t, t)^\ell$ approximates $\boldsymbol{d}(\mathbf{x}_t, t)^\ell$. The continuous-time sampler associated with this parameterization is the maximal-coupling CTMC of Appendix G.3, not the standard UDM CTMC of Appendix F. By Proposition 9, the exact reverse generator is

$$\bar{R}_t^{\mathrm{mc}}(\tilde{\mathbf{x}}|\mathbf{x}) = \begin{cases} \lambda_t\, p_{0|t}^\ell(\tilde{\mathbf{x}}^\ell|\mathbf{x}) & \text{if } \mathbf{x}^{-\ell} = \tilde{\mathbf{x}}^{-\ell} \text{ and } \mathbf{x}^\ell \neq \tilde{\mathbf{x}}^\ell, \\ 0 & \text{otherwise,} \end{cases} \qquad \lambda_t \coloneqq -\alpha_t'/(1 - \alpha_t) \;.$$

Replacing the exact denoiser by $\hat{\mathbf{x}}_0^\theta(\mathbf{x}, t)$ in (68) gives the planned-denoising rate

$$\bar{R}_t^\theta(\tilde{\mathbf{x}}|\mathbf{x}) = \begin{cases} \lambda_t\, \boldsymbol{a}_\theta(\mathbf{x}, t)^\ell \langle \tilde{\mathbf{x}}^\ell, \boldsymbol{d}_\theta(\mathbf{x}, t)^\ell \rangle & \text{if } \mathbf{x}^{-\ell} = \tilde{\mathbf{x}}^{-\ell} \text{ and } \mathbf{x}^\ell \neq \tilde{\mathbf{x}}^\ell, \\ 0 & \text{otherwise.} \end{cases} \;, \tag{47}$$

where the carry-over term vanishes because $\tilde{\mathbf{x}}^\ell \neq \mathbf{x}^\ell$. The planner and denoiser are then learned with the cross-entropy losses of Theorem 4.1 in [17], rather than with the continuous-time loss (69).

There are several differences between this construction and AUDM. First, AUDM conditions on the full noise vector U and does not rely on the mask perspective. As a result, AUDM uses a single noise-conditioned denoiser network, whereas planned denoising splits the marginal denoiser into the planner $\boldsymbol{a}_\theta$ and denoiser $\boldsymbol{d}_\theta$. Third, our framework allows us to recover the UDM process using a simple resampling operation and does not require training a planner.

Reparameterized discrete diffusion models [42]. We describe the construction in [42] using our own notation. Their starting point is the exact UDM bridge $q_{s|0,t}^\ell(\cdot|x_0, x_t)$, split according to whether x_t is equal to x_0. This is the same split used in Appendix D; with the constants γ_0, γ_2 of (55), we have

$$q_{s|0,t}^\ell(x_s|x_0, x_t) \coloneqq \begin{cases} (1 - \gamma_0)x_t + \gamma_0 \mathbf{1}/K & \text{if } x_t = x_0 \\ \gamma_2 x_0 + (1 - \gamma_2)\big(\alpha_{t|s} x_t + (1 - \alpha_{t|s})\mathbf{1}/K\big) & \text{if } x_t \neq x_0 \;, \end{cases} \tag{48}$$

Their Eq. 4 augments this bridge with Bernoulli latent random variables. In the notation above, this can be written as a joint p.m.f. over x_s and $\mathbf{r}_s \in \{0,1\}^{L \times 2}$:

$$\tilde{q}_{s|0,t}^\ell(x_s, \mathbf{r}_s^\ell|x_0, x_t) \coloneqq \tilde{q}_{s|0,t}^\ell(x_s|x_0, x_t, \mathbf{r}_s^\ell)\, \mathrm{Bern}(\mathbf{r}_s^{\ell 1}; 1 - \gamma_0)\mathrm{Bern}(\mathbf{r}_s^{\ell 2}; \gamma_2)$$

where

$$\tilde{q}^{\ell}_{s|0,t}(x_s|x_0,x_t,\mathbf{r}^{\ell}_s) \coloneqq \begin{cases} \mathrm{Cat}(x_s;\mathbf{r}^{\ell 1}_s x_t + (1-\mathbf{r}^{\ell 1}_s)\mathbf{1}/K) & \text{if } x_t = x_0 \\ \mathrm{Cat}(x_s;\mathbf{r}^{\ell 2}_s x_0 + (1-\mathbf{r}^{\ell 2}_s)(\alpha_{t|s}x_t + (1-\alpha_{t|s})\mathbf{1}/K)) & \text{if } x_t \neq x_0\ . \end{cases}$$

We emphasize here that marginalizing $\mathbf{r}_s$ recovers the original bridge above and that using (Marginalization) with (48) yields the same generative model as in Appendix D. Indeed, define the transition:

$$\tilde{p}^{\theta}_{s|t}(\mathbf{x}_s,\mathbf{r}_s|\mathbf{x}_t) \coloneqq \prod_{\ell=1}^{L}\sum_{\mathbf{x}^{\ell}_0} \tilde{q}^{\ell}_{s|0,t}(\mathbf{x}^{\ell}_s,\mathbf{r}^{\ell}_s|\mathbf{x}^{\ell}_0,\mathbf{x}^{\ell}_t)\,\mathrm{Cat}(\mathbf{x}^{\ell}_0;\hat{\mathbf{x}}^{\theta}_0(\mathbf{x}_t,t)^{\ell})$$

and the corresponding joint law

$$\tilde{p}^{\theta}_{0:n}(\mathbf{x}_{t_{0:n}},\mathbf{r}_{t_0:t_{n-1}}) \coloneqq p_{t_n}(\mathbf{x}_{t_n})\prod_{i=1}^{n}\tilde{p}^{\theta}_{s_i|t_i}(\mathbf{x}_{s_i},\mathbf{r}_{s_i}|\mathbf{x}_{t_i})\ .$$

Summing over $\mathbf{r}_{t_0:t_{n-1}}$ gives exactly the reverse-chain law obtained from (Marginalization) and (48), hence the same generative model as in Appendix D. Moreover, under this joint law of $(X_{t_{0:n}}, R_{t_{0:n-1}})$, the random variables R_{s_i} and X_{t_i} are independent.

The RDM model in [42] changes this lifted model by adding a dependence of R_{s_i} on X_{t_i} and a second latent variable $\mathbf{b}_t$ whose purpose is to substitute the indicators $\langle \mathbf{x}^{\ell}_t, \mathbf{x}^{\ell}_0\rangle$ in (48). The considered transitions are instead

$$\begin{aligned}
&\tilde{p}^{\theta,\ell}_{s|t}(x^{\ell}_s,\mathbf{r}_s,\mathbf{b}^{\ell}_s|\mathbf{x}_t,\mathbf{b}_t) \\
&\quad\coloneqq \mathrm{Bern}(\mathbf{r}^{\ell 1}_s;\rho_{s,t,\theta}(\mathbf{x}_t,\mathbf{b}_t)^{\ell 1})\mathrm{Bern}(\mathbf{r}^{\ell 2}_s;\rho_{s,t,\theta}(\mathbf{x}_t,\mathbf{b}_t)^{\ell 2})\mathbb{1}\{\mathbf{b}^{\ell}_s = (\mathbf{b}^{\ell}_t \wedge \mathbf{r}^{\ell 1}_s)\vee \mathbf{r}^{\ell 2}_s\} \\
&\qquad\times \begin{cases} \mathrm{Cat}(x^{\ell}_s;\mathbf{r}^{\ell 1}_s\cdot \mathbf{x}^{\ell}_t + (1-\mathbf{r}^{\ell 1}_s)\mathbf{1}/K) & \text{if } \mathbf{b}^{\ell}_t = 1 \\ \mathrm{Cat}(x^{\ell}_s;\mathbf{r}^{\ell 2}_s\cdot \hat{\mathbf{x}}^{\theta}_0(\mathbf{x}_t,t)^{\ell} + (1-\mathbf{r}^{\ell 2}_s)(\alpha_{t|s}\mathbf{x}^{\ell}_t + (1-\alpha_{t|s})\mathbf{1}/K)) & \text{if } \mathbf{b}^{\ell}_t = 0\ , \end{cases}
\end{aligned}$$

and $\rho_{s,t,\theta}(\mathbf{x}_t,\mathbf{b}_t) \in [0,1]^{L\times 2}$ are probability vectors computed with a given rule provided in [42, Section 4.3].

These transitions are best viewed as a type of confidence-sampling for UDMs. Despite the latent variables, AUDM and ReAUDM are substantially different, since our data augmentation is part of an exact lifted model: U is sampled from the uniform corruption law, the reverse chain is defined conditionally on this variable, and marginalizing U recovers the UDM reverse-chain law by Proposition 3. By contrast, once the route probabilities are made confidence-dependent, the variables $(\mathbf{r}_s,\mathbf{b}_s)$ in RDM no longer describe a marginally exact augmentation of the UDM reverse process. They are auxiliary decoding variables used to decide which positions should be treated as already denoised, and thus serve a completely different purpose than our data augmentation.

Discrete non-Markov diffusion models [5]. We now describe the process used in [5] through the bridge it induces. In the UDM case, their construction first samples transition times $\boldsymbol{\tau} \in \mathbb{R}^{L}_{\geq 0}$ with independent coordinates satisfying $\mathbb{P}(\boldsymbol{\tau}^{\ell} < t) = 1-\alpha_t$. Conditionally on $(X_0,\boldsymbol{\tau})$, the process evolves in reverse time with $X_1 \sim \mathrm{Uniform}(\llbracket 1,K\rrbracket)^{\otimes L}$, and its bridge for $0<s<t<1$ is deterministic at every position:

$$q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0,\mathbf{x}_t,\boldsymbol{\tau}) = \textstyle\prod_{\ell=1}^{L}\begin{cases} \mathrm{Cat}(\mathbf{x}^{\ell}_s;\mathbf{x}^{\ell}_0) & \text{if } s<\boldsymbol{\tau}^{\ell}\leq t, \\ \mathrm{Cat}(\mathbf{x}^{\ell}_s;\mathbf{x}^{\ell}_t) & \text{otherwise.} \end{cases}\ .$$

The corresponding conditional law at time t is

$$q_{t|0}(\mathbf{x}_t|\mathbf{x}_0,\boldsymbol{\tau}) = \textstyle\prod_{\ell=1}^{L}\mathrm{Cat}\big(\mathbf{x}^{\ell}_t;\mathbb{1}_{\boldsymbol{\tau}^{\ell}>t}\mathbf{x}^{\ell}_0 + \mathbb{1}_{\boldsymbol{\tau}^{\ell}\leq t}\mathbf{1}/K\big)\ ,$$

and marginalizing over $\boldsymbol{\tau}$ recovers the UDM forward transition. Equivalently, with $j(\boldsymbol{\tau}) \coloneqq \prod_{\ell=1}^{L}(-\alpha'_{\boldsymbol{\tau}^{\ell}})$, these bridge transitions satisfy

$$\textstyle\int \sum_{\mathbf{x}_t} q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0,\mathbf{x}_t,\boldsymbol{\tau})q_{t|0}(\mathbf{x}_t|\mathbf{x}_0,\boldsymbol{\tau})j(\boldsymbol{\tau})\,\mathrm{d}\boldsymbol{\tau} = q_{s|0}(\mathbf{x}_s|\mathbf{x}_0)\ .$$

Thus this process is marginal preserving.

The transition of the Markovian projection relevant for reverse sampling conditioned on $\boldsymbol{\tau}$ is given by

$$p_{s|t}(\mathbf{x}_s|\mathbf{x}_t, \boldsymbol{\tau}) = \sum_{\mathbf{x}_0} q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{x}_t, \boldsymbol{\tau})\, p_{0|t}(\mathbf{x}_0|\mathbf{x}_t, \boldsymbol{\tau}) \ .$$

The posterior appearing in this reverse transition is

$$p_{0|t}(\mathbf{x}_0|\mathbf{x}_t, \boldsymbol{\tau}) = \frac{p_0(\mathbf{x}_0) \prod_{\ell:\boldsymbol{\tau}^\ell > t} \mathbb{1}_{\mathbf{x}_0^\ell = \mathbf{x}_t^\ell}}{\sum_{\tilde{\mathbf{x}}_0} p_0(\tilde{\mathbf{x}}_0) \prod_{\ell:\boldsymbol{\tau}^\ell > t} \mathbb{1}_{\tilde{\mathbf{x}}_0^\ell = \mathbf{x}_t^\ell}} \ ,$$

Writing $\mathbf{n}_t^\ell \coloneqq \mathbb{1}_{\boldsymbol{\tau}^\ell \leq t}$, this posterior is exactly the full-mask posterior of (43):

$$p_{0|t}(\mathbf{x}_0|\mathbf{x}_t, \boldsymbol{\tau}) = p_{0|t}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{n}_t) \ .$$

This makes the link with planned denoising explicit. The transition times define a deterministic planner mask between s and t: positions satisfying $s < \boldsymbol{\tau}^\ell \leq t$ are resampled from the denoising posterior through the bridge, while all other positions are carried over from $\mathbf{x}_t$. Thus the exact denoising object suggested by the Markovian projection is the $\boldsymbol{\tau}$-conditioned posterior $p_{0|t}(\cdot|\mathbf{x}_t, \boldsymbol{\tau})$. In [5], however, the implemented transition plugs in a standard denoiser $p_{0|t}(\cdot|\mathbf{x}_t)$ which is not conditioned on $\boldsymbol{\tau}$.

C Masked Uniform Diffusion

This section develops a transition-time lifting complementary to the absorbing-state construction of Section 4. It shows that the UDM denoising problem can be exactly reduced to an MDM denoising problem and it is therefore possible to reuse a denoiser trained on MDM while reproducing the full UDM joint law. Unless explicitly stated otherwise, all forward and reverse quantities without an additional argument are those associated with UDM.

C.1 Transition-time construction

The absorbing-state lifting in Section 4 gives a noise-conditioned analogue of masked diffusion. We now give a complementary lifting in which the connection to masked diffusion is exact: conditioned on latent transition times, the UDM denoising problem is precisely an MDM denoising problem. Following [5], we introduce the transition times $\boldsymbol{\tau} \in \mathbb{R}_{\geq 0}^L$ with c.d.f $\mathbb{P}(\boldsymbol{\tau}^\ell < t) = 1 - \alpha_t$. Conditionally on $(X_0, \boldsymbol{\tau})$, we consider the process $(X_t)_{t\in(0,1]}$ evolving in reverse time with $X_1 \sim \mathrm{Uniform}(\llbracket 1, K \rrbracket)^{\otimes L}$ and with transitions for $0 < s < t < 1$ given by

$$q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{x}_t, \boldsymbol{\tau}) = \prod_{\ell=1}^{L} \begin{cases} \mathrm{Cat}(\mathbf{x}_s^\ell; \mathbf{x}_0^\ell) & \text{if } s < \boldsymbol{\tau}^\ell \leq t, \\ \mathrm{Cat}(\mathbf{x}_s^\ell; \mathbf{x}_t^\ell) & \text{otherwise,} \end{cases} \ .$$

Essentially X_s^ℓ equals X_0^ℓ exactly when the jump occurs between s and t; otherwise no jump occurs on $(s, t]$, so $X_s^\ell = X_t^\ell$. Furthermore, the conditional distribution of X_t is

$$q_{t|0}(\mathbf{x}_t|\mathbf{x}_0, \boldsymbol{\tau}) = \textstyle\prod_{\ell=1}^{L} \mathrm{Cat}\left(\mathbf{x}_t^\ell; \mathbb{1}_{\boldsymbol{\tau}^\ell > t}\mathbf{x}_0^\ell + \mathbb{1}_{\boldsymbol{\tau}^\ell \leq t}\mathbf{1}/K\right) \ ,$$

and thus, marginalizing over $\boldsymbol{\tau}$ recovers the UDM forward transition $q_{t|0}(\mathbf{x}_t|\mathbf{x}_0)$.

Define $\tilde{\mathbf{x}}_t(\boldsymbol{\tau})^\ell \coloneqq \mathbf{x}_t^\ell$ when $\boldsymbol{\tau}^\ell > t$ and $\tilde{\mathbf{x}}_t(\boldsymbol{\tau})^\ell \coloneqq \mathsf{m}$ when $\boldsymbol{\tau}^\ell \leq t$. We show in Appendix C.3 that the reverse Markov chain conditioned only on $\boldsymbol{\tau}$ admits the following one step transition

$$p_{s|t}(\mathbf{x}_s|\mathbf{x}_t, \boldsymbol{\tau}) = \sum_{\mathbf{x}_0} q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{x}_t, \boldsymbol{\tau})\, p_{0|t}^{\mathrm{mask}}(\mathbf{x}_0|\tilde{\mathbf{x}}_t(\boldsymbol{\tau})) \ ,$$

where $p_{0|t}^{\mathrm{mask}}$ is the denoiser of an MDM. This shows that we can reuse an MDM denoiser to define a discrete diffusion model with UDM marginals.

C.2 Recovering the UDM joint distribution

We can go one step further and recover the full UDM joint law (19). We apply the same logic as in Section 4.2 and Algorithm 1 and resample the transition times after each reverse step according to $j_{s|0}(\boldsymbol{\tau}_s|\mathbf{x}_0, \mathbf{x}_s) \coloneqq \prod_{\ell=1}^{L} j_{s|0}^\ell(\boldsymbol{\tau}_s^\ell|\mathbf{x}_0^\ell, \mathbf{x}_s^\ell)$ with

$$j_{s|0}(\boldsymbol{\tau}_s^\ell|\mathbf{x}_0^\ell, \mathbf{x}_s^\ell) = \begin{cases} \frac{-\alpha'_{\boldsymbol{\tau}_s^\ell}}{1-\alpha_s} \mathbb{1}_{\boldsymbol{\tau}_s^\ell \leq s} & \text{if } \mathbf{x}_s^\ell \neq \mathbf{x}_0^\ell, \\ \frac{-\alpha'_{\boldsymbol{\tau}_s^\ell}}{1+(K-1)\alpha_s} \left(\mathbb{1}_{\boldsymbol{\tau}_s^\ell \leq s} + K\mathbb{1}_{\boldsymbol{\tau}_s^\ell > s}\right) & \text{if } \mathbf{x}_s^\ell = \mathbf{x}_0^\ell. \end{cases}$$

The corresponding lifted reverse transition is then

$$\bar{p}_{s|t}(\mathbf{x}_s, \boldsymbol{\tau}_s | \mathbf{x}_t, \boldsymbol{\tau}_t) \coloneqq \sum_{\mathbf{x}_0} j_{s|0}(\boldsymbol{\tau}_s | \mathbf{x}_0, \mathbf{x}_s)\, q_{s|0,t}(\mathbf{x}_s | \mathbf{x}_0, \mathbf{x}_t)\, p_{0|t}^{\text{mask}}(\mathbf{x}_0 | \tilde{\mathbf{x}}_t(\boldsymbol{\tau}_t)) \ .$$

This has a simple remasking interpretation. If $\mathbf{x}_s^\ell \neq \mathbf{x}_0^\ell$, then the transition must already have occurred before time s, so $\boldsymbol{\tau}_s^\ell \leq s$. If $\mathbf{x}_s^\ell = \mathbf{x}_0^\ell$, there are two possible explanations: either the position has not transitioned yet, so $\boldsymbol{\tau}_s^\ell > s$, or it has already transitioned before time s and the uniform resampling happened to return $\mathbf{x}_0^\ell$. The law $j_{s|0}$ reweights these two cases appropriately.

The resulting algorithm analogous to Algorithm 1 is provided in Algorithm 4. We then have the following result:

Proposition 7. *The trajectory generated in Algorithm 4 satisfies Law*$(X_{t_0}, \dots, X_{t_n}) = p_{0:n}$*, where* $p_{0:n}$ *is the uniform UDM reverse-chain law defined in* (19)*.*

This formalizes the idea that UDM is naturally associated with a remasking mechanism absent from standard MDM, and clarifies the relation between these two constructions. Appendix C.3 gives the full derivation and algorithm.

C.3 Derivations and sampling algorithm

We write $j(\boldsymbol{\tau}) \coloneqq \prod_{\ell=1}^{L} (-\alpha'_{\boldsymbol{\tau}^\ell})$ for the density of the transition times.

Lemma 2. *For every* $t \in [0, 1]$,

$$q_{t|0}(\mathbf{x}_t | \mathbf{x}_0, \boldsymbol{\tau}) = \prod_{\ell=1}^{L} \mathrm{Cat}\big(\mathbf{x}_t^\ell; \mathbb{1}_{\boldsymbol{\tau}^\ell > t} \mathbf{x}_0^\ell + \mathbb{1}_{\boldsymbol{\tau}^\ell \leq t} \mathbf{1}/K\big) \ . \tag{49}$$

Proof. Fix a grid $0 = t_0 < \cdots < t_n = 1$. Starting from $q_{t_0|0}(\mathbf{x}_{t_0} | \mathbf{x}_0, \boldsymbol{\tau}) = \mathrm{Cat}(\mathbf{x}_{t_0}; \mathbf{x}_0)$, define recursively

$$q_{t_i|0}(\mathbf{x}_{t_i} | \mathbf{x}_0, \boldsymbol{\tau}) \coloneqq \sum_{\mathbf{x}_{t_{i+1}}} q_{t_i|0,t_{i+1}}(\mathbf{x}_{t_i} | \mathbf{x}_0, \mathbf{x}_{t_{i+1}}, \boldsymbol{\tau})\, q_{t_{i+1}|0}(\mathbf{x}_{t_{i+1}} | \mathbf{x}_0, \boldsymbol{\tau}) \ .$$

We prove by backward induction on the grid that (49) holds at every time t_i. Assume it holds at time t_{i+1}. Since both the bridge and the induction hypothesis factorize over positions, it is enough to compute one position ℓ:

$$\begin{aligned}
&\sum_{\mathbf{x}_{t_{i+1}}^\ell} q_{t_i|0,t_{i+1}}^\ell(\mathbf{x}_{t_i}^\ell | \mathbf{x}_0^\ell, \mathbf{x}_{t_{i+1}}^\ell, \boldsymbol{\tau}^\ell)\, \mathrm{Cat}\Big(\mathbf{x}_{t_{i+1}}^\ell; \mathbb{1}_{\boldsymbol{\tau}^\ell > t_{i+1}} \mathbf{x}_0^\ell + \mathbb{1}_{\boldsymbol{\tau}^\ell \leq t_{i+1}} \mathbf{1}/K\Big) \\
&\quad = \mathbb{1}_{\boldsymbol{\tau}^\ell \leq t_i} \sum_{\mathbf{x}_{t_{i+1}}^\ell} \mathrm{Cat}(\mathbf{x}_{t_i}^\ell; \mathbf{x}_{t_{i+1}}^\ell)\, \mathrm{Cat}(\mathbf{x}_{t_{i+1}}^\ell; \mathbf{1}/K) \\
&\qquad + \mathbb{1}_{t_i < \boldsymbol{\tau}^\ell \leq t_{i+1}} \sum_{\mathbf{x}_{t_{i+1}}^\ell} \mathrm{Cat}(\mathbf{x}_{t_i}^\ell; \mathbf{x}_0^\ell)\, \mathrm{Cat}(\mathbf{x}_{t_{i+1}}^\ell; \mathbf{1}/K) \\
&\qquad + \mathbb{1}_{\boldsymbol{\tau}^\ell > t_{i+1}} \sum_{\mathbf{x}_{t_{i+1}}^\ell} \mathrm{Cat}(\mathbf{x}_{t_i}^\ell; \mathbf{x}_{t_{i+1}}^\ell)\, \mathrm{Cat}(\mathbf{x}_{t_{i+1}}^\ell; \mathbf{x}_0^\ell) \\
&\quad = \mathbb{1}_{\boldsymbol{\tau}^\ell \leq t_i} \mathrm{Cat}(\mathbf{x}_{t_i}^\ell; \mathbf{1}/K) + \mathbb{1}_{t_i < \boldsymbol{\tau}^\ell \leq t_{i+1}} \mathrm{Cat}(\mathbf{x}_{t_i}^\ell; \mathbf{x}_0^\ell) + \mathbb{1}_{\boldsymbol{\tau}^\ell > t_{i+1}} \mathrm{Cat}(\mathbf{x}_{t_i}^\ell; \mathbf{x}_0^\ell) \\
&\quad = \mathrm{Cat}\big(\mathbf{x}_{t_i}^\ell; \mathbb{1}_{\boldsymbol{\tau}^\ell > t_i} \mathbf{x}_0^\ell + \mathbb{1}_{\boldsymbol{\tau}^\ell \leq t_i} \mathbf{1}/K\big) \ .
\end{aligned}$$

This proves the induction step. Hence (49) holds for every time of the grid, and therefore for every t. □

Lemma 3. *If*

$$\tilde{\mathbf{x}}_t(\boldsymbol{\tau})^\ell \coloneqq \begin{cases} \mathbf{x}_t^\ell & \text{if } \boldsymbol{\tau}^\ell > t, \\ \mathsf{m} & \text{if } \boldsymbol{\tau}^\ell \leq t, \end{cases}$$

then

$$p_{0|t}(\mathbf{x}_0 | \mathbf{x}_t, \boldsymbol{\tau}) = p_{0|t}^{\text{mask}}(\mathbf{x}_0 | \tilde{\mathbf{x}}_t(\boldsymbol{\tau})) \ .$$

Proof. By Bayes' rule and (49),

$$p_{0|t}(\mathbf{x}_0|\mathbf{x}_t,\boldsymbol{\tau}) = \frac{p_0(\mathbf{x}_0)q_{t|0}(\mathbf{x}_t|\mathbf{x}_0,\boldsymbol{\tau})}{\sum_{\tilde{\mathbf{x}}_0} p_0(\tilde{\mathbf{x}}_0)q_{t|0}(\mathbf{x}_t|\tilde{\mathbf{x}}_0,\boldsymbol{\tau})} = \frac{p_0(\mathbf{x}_0)\prod_{\ell:\boldsymbol{\tau}^\ell>t}\mathbb{1}_{\mathbf{x}_0^\ell=\mathbf{x}_t^\ell}}{\sum_{\tilde{\mathbf{x}}_0} p_0(\tilde{\mathbf{x}}_0)\prod_{\ell:\boldsymbol{\tau}^\ell>t}\mathbb{1}_{\tilde{\mathbf{x}}_0^\ell=\mathbf{x}_t^\ell}} .$$

Since $\tilde{\mathbf{x}}_t(\boldsymbol{\tau})^\ell \neq \mathsf{m}$ if and only if $\boldsymbol{\tau}^\ell > t$, and then necessarily $\tilde{\mathbf{x}}_t(\boldsymbol{\tau})^\ell = \mathbf{x}_t^\ell$, we have

$$\prod_{\ell:\boldsymbol{\tau}^\ell>t}\mathbb{1}_{\mathbf{x}_0^\ell=\mathbf{x}_t^\ell} = \prod_{\ell:\tilde{\mathbf{x}}_t(\boldsymbol{\tau})^\ell\neq\mathsf{m}}\mathbb{1}_{\mathbf{x}_0^\ell=\tilde{\mathbf{x}}_t(\boldsymbol{\tau})^\ell} ,$$

and similarly in the denominator. Therefore

$$p_{0|t}(\mathbf{x}_0|\mathbf{x}_t,\boldsymbol{\tau}) = \frac{p_0(\mathbf{x}_0)\prod_{\ell:\tilde{\mathbf{x}}_t(\boldsymbol{\tau})^\ell\neq\mathsf{m}}\mathbb{1}_{\mathbf{x}_0^\ell=\tilde{\mathbf{x}}_t(\boldsymbol{\tau})^\ell}}{\sum_{\tilde{\mathbf{x}}_0} p_0(\tilde{\mathbf{x}}_0)\prod_{\ell:\tilde{\mathbf{x}}_t(\boldsymbol{\tau})^\ell\neq\mathsf{m}}\mathbb{1}_{\tilde{\mathbf{x}}_0^\ell=\tilde{\mathbf{x}}_t(\boldsymbol{\tau})^\ell}} = p_{0|t}^{\mathrm{mask}}(\mathbf{x}_0|\tilde{\mathbf{x}}_t(\boldsymbol{\tau})) .$$

□

Using the $\boldsymbol{\tau}$-conditioned bridge and reverse transition introduced in Appendix C.1, we denote by $j_t(\boldsymbol{\tau}|\mathbf{x}_t)$ the posterior density of $\boldsymbol{\tau}$ given $X_t = \mathbf{x}_t$, namely

$$j_t(\boldsymbol{\tau}|\mathbf{x}_t) \coloneqq \sum_{\mathbf{x}_0} \frac{p_0(\mathbf{x}_0)q_{t|0}(\mathbf{x}_t|\mathbf{x}_0,\boldsymbol{\tau})}{p_t(\mathbf{x}_t)} j(\boldsymbol{\tau}) ,$$

then the exact UDM denoising posterior decomposes as

$$p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) = \int p_{0|t}(\mathbf{x}_0|\mathbf{x}_t,\boldsymbol{\tau})\, j_t(\boldsymbol{\tau}|\mathbf{x}_t) . \tag{50}$$

This is the exact $\boldsymbol{\tau}$-mixture identity behind the masked-diffusion view of UDM. The difficulty is that $j_t(\cdot|\mathbf{x}_t)$ is not tractable, so it cannot be used directly in a sampler.

To obtain an exact lifted chain with a tractable resampling step, we instead condition on (X_0, X_s) and define

$$j_{s|0}(\boldsymbol{\tau}_s|\mathbf{x}_0,\mathbf{x}_s) \coloneqq \frac{j(\boldsymbol{\tau}_s)q_{s|0}(\mathbf{x}_s|\mathbf{x}_0,\boldsymbol{\tau}_s)}{q_{s|0}(\mathbf{x}_s|\mathbf{x}_0)} = \prod_{\ell=1}^{L}\begin{cases} \frac{-\alpha'_{\boldsymbol{\tau}_s^\ell}}{1-\alpha_s}\mathbb{1}_{\boldsymbol{\tau}_s^\ell\le s} & \mathbf{x}_s^\ell \neq \mathbf{x}_0^\ell, \\ \frac{-\alpha'_{\boldsymbol{\tau}_s^\ell}}{1+(K-1)\alpha_s}\left(\mathbb{1}_{\boldsymbol{\tau}_s^\ell\le s} + K\mathbb{1}_{\boldsymbol{\tau}_s^\ell>s}\right) & \mathbf{x}_s^\ell = \mathbf{x}_0^\ell. \end{cases} \tag{51}$$

If $\mathbf{x}_s^\ell \neq \mathbf{x}_0^\ell$, then the transition must already have occurred before time s, so $\boldsymbol{\tau}_s^\ell$ is sampled from the prior restricted to $[0, s]$. If $\mathbf{x}_s^\ell = \mathbf{x}_0^\ell$, then either the position has not transitioned yet, or it has already transitioned and the uniform resampling happened to return $\mathbf{x}_0^\ell$. The law $j_{s|0}$ exactly reweights these two possibilities.

The lifted reverse transition is then

$$\bar{p}_{s|t}(\mathbf{x}_s,\boldsymbol{\tau}_s|\mathbf{x}_t,\boldsymbol{\tau}_t) \coloneqq \sum_{\mathbf{x}_0} j_{s|0}(\boldsymbol{\tau}_s|\mathbf{x}_0,\mathbf{x}_s)\, q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0,\mathbf{x}_t)\, p_{0|t}(\mathbf{x}_0|\mathbf{x}_t,\boldsymbol{\tau}_t) . \tag{52}$$

For a grid $0 = t_0 < \cdots < t_n = 1$, let $\bar{p}_{0:n}$ denote the corresponding path law, with initialization $\bar{p}_{t_n}(\mathbf{x}_{t_n},\boldsymbol{\tau}_{t_n}) \coloneqq p_{t_n}(\mathbf{x}_{t_n})j_{t_n}(\boldsymbol{\tau}_{t_n}|\mathbf{x}_{t_n})$. If $\alpha_{t_n} = 0$, this reduces to $X_{t_n} \sim \boldsymbol{\upsilon}^{\otimes L}$ and $\boldsymbol{\tau}_{t_n} \sim j$.

Proof of Proposition 7. Similar to the proof of Proposition 3, it is enough to prove the one-step identity:

$$\int \bar{p}_{s|t}(\mathbf{x}_s,\boldsymbol{\tau}_s|\mathbf{x}_t,\boldsymbol{\tau}_t)\, j_t(\boldsymbol{\tau}_t|\mathbf{x}_t) = p_{s|t}(\mathbf{x}_s|\mathbf{x}_t)\, j_s(\boldsymbol{\tau}_s|\mathbf{x}_s) .$$

Using (52), we obtain

$$\begin{aligned}&\int \bar{p}_{s|t}(\mathbf{x}_s,\boldsymbol{\tau}_s|\mathbf{x}_t,\boldsymbol{\tau}_t)\, j_t(\boldsymbol{\tau}_t|\mathbf{x}_t) \\ &\quad= \sum_{\mathbf{x}_0} j_{s|0}(\boldsymbol{\tau}_s|\mathbf{x}_0,\mathbf{x}_s)\, q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0,\mathbf{x}_t) \int p_{0|t}(\mathbf{x}_0|\mathbf{x}_t,\boldsymbol{\tau}_t)\, j_t(\boldsymbol{\tau}_t|\mathbf{x}_t) .\end{aligned}$$

Algorithm 4 Resampled MUDM

Require: grid $0 = t_0 < \cdots < t_n = 1$
  Draw $X_{t_n} \sim p_{t_n}(\cdot)$ and $\boldsymbol{\tau}_{t_n} \sim j_{t_n}(\cdot|X_{t_n})$
  for $i = n{-}1, \dots, 0$ **do**
    Draw $X_0 \sim p^{\theta,\mathrm{mask}}_{0|t_{i+1}}(\cdot|\tilde{X}_{t_{i+1}}(\boldsymbol{\tau}_{t_{i+1}}))$
    Draw $X_{t_i} \sim q_{t_i|0,t_{i+1}}(\cdot|X_0, X_{t_{i+1}})$
    for $\ell = 1, \dots, L$ **do**
      if $X^\ell_{t_i} \neq X^\ell_0$ **then**
        Draw $\tau^\ell_{t_i}$ with density $\frac{-\alpha'_\tau}{1-\alpha_{t_i}} \mathbb{1}_{\tau \le t_i}$
      else
        Draw $\tau^\ell_{t_i}$ with density $\frac{-\alpha'_\tau}{1+(K-1)\alpha_{t_i}} \big(\mathbb{1}_{\tau \le t_i} + K \mathbb{1}_{\tau > t_i}\big)$
      end if
    end for
  end for
  return X_{t_0}

By (50), the last integral is $p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)$. Therefore

$$\int \bar{p}_{s|t}(\mathbf{x}_s, \boldsymbol{\tau}_s|\mathbf{x}_t, \boldsymbol{\tau}_t)\, j_t(\boldsymbol{\tau}_t|\mathbf{x}_t) = \sum_{\mathbf{x}_0} j_{s|0}(\boldsymbol{\tau}_s|\mathbf{x}_0, \mathbf{x}_s)\, q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{x}_t) p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)\ .$$

Now use $p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) = p_0(\mathbf{x}_0) q_{t|0}(\mathbf{x}_t|\mathbf{x}_0) \big/ p_t(\mathbf{x}_t)$ and the UDM bridge identity

$$q_{s|0,t}(\mathbf{x}_s|\mathbf{x}_0, \mathbf{x}_t)\, q_{t|0}(\mathbf{x}_t|\mathbf{x}_0) = q_{s|0}(\mathbf{x}_s|\mathbf{x}_0)\, q_{t|s}(\mathbf{x}_t|\mathbf{x}_s)\ .$$

Substituting (51) then gives

$$\int \bar{p}_{s|t}(\mathbf{x}_s, \boldsymbol{\tau}_s|\mathbf{x}_t, \boldsymbol{\tau}_t)\, j_t(\boldsymbol{\tau}_t|\mathbf{x}_t) = \frac{q_{t|s}(\mathbf{x}_t|\mathbf{x}_s)}{p_t(\mathbf{x}_t)} \sum_{\mathbf{x}_0} p_0(\mathbf{x}_0)\, j(\boldsymbol{\tau}_s)\, q_{s|0}(\mathbf{x}_s|\mathbf{x}_0, \boldsymbol{\tau}_s)\ .$$

The sum on the right-hand side is exactly the joint forward law of $(X_s, \boldsymbol{\tau}_s)$, hence it is equal to $p_s(\mathbf{x}_s) j_s(\boldsymbol{\tau}_s|\mathbf{x}_s)$. We conclude that

$$\begin{aligned}
&\int \bar{p}_{s|t}(\mathbf{x}_s, \boldsymbol{\tau}_s|\mathbf{x}_t, \boldsymbol{\tau}_t)\, j_t(\boldsymbol{\tau}_t|\mathbf{x}_t) \\
&\quad = \frac{p_s(\mathbf{x}_s) q_{t|s}(\mathbf{x}_t|\mathbf{x}_s)}{p_t(\mathbf{x}_t)} j_s(\boldsymbol{\tau}_s|\mathbf{x}_s) \\
&\quad = p_{s|t}(\mathbf{x}_s|\mathbf{x}_t)\, j_s(\boldsymbol{\tau}_s|\mathbf{x}_s)\ .
\end{aligned}$$

This proves the one-step identity. Iterating along the grid yields the claimed equality of the X-trajectory law with the UDM reverse-chain law. □

D On the role of the bridge extension in the plug-in parameterization

The plug-in parameterization Bridge Plug-in is only well defined once one specifies how the token-wise bridge $q^\ell_{s|0,t}(x_s|\cdot,\cdot)$, originally defined for pairs of one-hot vectors, is extended to a first argument in the simplex Δ_K. As a consequence, different extensions coinciding on one-hot inputs define the same exact bridge, while leading to different plug-in parameterizations once the network output is interpreted as a probability vector.

The following remark specifies the bridge conventions used in Section 3.1.

Remark 2 (Bridge extensions)**.** *There are two extension choices required for our analysis. First, the bridge is determined by the Bayes' formula only on pairs (x_0, x_t) such that $q^\ell_{t|0}(x_t|x_0) > 0$. Outside this support, its value is fixed by convention. For UDM no such completion is needed, while for MDM we use the completion (and keep the same notation) of [31] which agrees with (6) on supported pairs:*

$$q^\ell_{s|0,t}(x_s|x_0, x_t) = \begin{cases} \mathrm{Cat}\,(x_s; x_t) & \text{if } x_t \neq \mathsf{m} \\ \mathrm{Cat}\left(x_s; \frac{\alpha_s - \alpha_t}{1-\alpha_t} x_0 + \frac{1-\alpha_s}{1-\alpha_t} \mathsf{m}\right) & \text{otherwise.} \end{cases} \tag{53}$$

*Second, the plug-in parameterization evaluates the first bridge argument at the simplex-valued prediction $\hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t)^\ell$, which is not necessarily a one-hot vector. The extensions we consider next satisfy $q^\ell_{s|0,t}(x_s|\nu, x_t) = q^\ell_{t|s}(x_t|x_s)q^\ell_{s|0}(x_s|\nu)/q^\ell_{t|0}(x_t|\nu)$ for $\nu \in \Delta_K$. We refer to this property as Assumption (**A1**) in Appendix A. It holds by construction for the UDM bridge (7), and (53) also satisfies it, as shown in Appendix A.*

The extension used in the main text is the "canonical" one obtained by keeping the Bayes formula (3) for $\mu \in \Delta_K$, when it is possible. In the UDM case, this yields

$$q^\ell_{s|0,t}(\cdot|\mu, x_t) = \mathrm{Cat}\left(\cdot; \frac{K\alpha_t\langle x_t, \mu\rangle x_t + (\alpha_{t|s} - \alpha_t)x_t + (\alpha_s - \alpha_t)\mu + D_{s,t}\mathbf{1}/K}{K\alpha_t\langle x_t, \mu\rangle + 1 - \alpha_t}\right),$$

which is nonlinear in μ because of the denominator. This is precisely the extension for which Proposition 1 shows that the minimizer of the expected NELBO is the leave-one-out posterior.

Another admissible choice is the barycentric extension obtained by averaging the exact bridge over a clean token drawn from μ. Since every $\mu \in \Delta_K$ admits the decomposition $\mu = \sum_{x_0 \in \mathsf{V}} \langle x_0, \mu\rangle x_0$, this extension is uniquely characterized by

$$\bar{q}^\ell_{s|0,t}(\cdot|\mu, x_t) \coloneqq \sum_{x_0 \in \mathsf{V}} \langle x_0, \mu\rangle q^\ell_{s|0,t}(\cdot|x_0, x_t). \tag{54}$$

By construction, (54) coincides with the exact bridge on the vertices of the simplex and is affine in μ. In MDM, because the bridge is already affine, this extension coincides with the canonical one. In UDM, the two differ.

For UDM, (54) admits a simple closed form. Define

$$\gamma_0 \coloneqq \frac{D_{s,t}}{1 + (K-1)\alpha_t}, \quad \gamma_1 \coloneqq \frac{\alpha_{t|s} - \alpha_t}{1 - \alpha_t}, \quad \gamma_2 \coloneqq \frac{\alpha_s - \alpha_t}{1 - \alpha_t}, \quad \gamma_3 \coloneqq \frac{D_{s,t}}{1 - \alpha_t},$$

and write $x_t = e_k$ and $\mu_k \coloneqq \langle e_k, \mu\rangle$. Then

$$\bar{q}^\ell_{s|0,t}(\cdot|\mu, e_k) = \mathrm{Cat}\left(\cdot; \gamma_2\mu + (\gamma_1 + (\gamma_3 - \gamma_0)\mu_k)\, e_k + (\gamma_3 + (\gamma_0 - \gamma_3)\mu_k)\, \frac{\mathbf{1}}{K}\right). \tag{55}$$

Indeed, when $x_t = x_0$, the exact UDM bridge is

$$q^\ell_{s|0,t}(\cdot|x_t, x_t) = \mathrm{Cat}\left(\cdot; \left(1 - \frac{D_{s,t}}{1 + (K-1)\alpha_t}\right) x_t + \frac{D_{s,t}}{1 + (K-1)\alpha_t}\frac{\mathbf{1}}{K}\right),$$

whereas for $x_0 \neq x_t$ it is

$$q^\ell_{s|0,t}(\cdot|x_0, x_t) = \mathrm{Cat}\left(\cdot; \frac{\alpha_s - \alpha_t}{1 - \alpha_t}x_0 + \frac{\alpha_{t|s} - \alpha_t}{1 - \alpha_t}x_t + \frac{D_{s,t}}{1 - \alpha_t}\frac{\mathbf{1}}{K}\right),$$

and averaging these two cases against μ gives (55).

This extension makes the denoiser reappear as the plug-in optimum. Indeed, for every $\ell \in \{1, \dots, L\}$ and every $\mathbf{x}_t \in \mathsf{X}$,

$$\begin{aligned} p^\ell_{s|t}(\cdot|\mathbf{x}_t) &= \sum_{x_0 \in \mathsf{V}} \bar{q}^\ell_{s|0,t}(\cdot|x_0, \mathbf{x}_t^\ell) p^\ell_{0|t}(x_0|\mathbf{x}_t) \\ &= \bar{q}^\ell_{s|0,t}(\cdot|\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)^\ell, \mathbf{x}_t^\ell), \end{aligned}$$

where the second line is exactly the definition of the barycentric extension with $\mu = \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)^\ell$. Therefore, if one defines the plug-in reverse kernel by

$$\hat{p}^\theta_{s|t}(\mathbf{x}_s|\mathbf{x}_t) \coloneqq \prod_{\ell=1}^{L} \bar{q}^\ell_{s|0,t}(\mathbf{x}_s^\ell|\hat{\mathbf{x}}_0^\theta(\mathbf{x}_t, t)^\ell, \mathbf{x}_t^\ell),$$

then the argument used for the marginalization parameterization applies, and shows that the denoising posterior $p^\ell_{0|t}(\cdot|\mathbf{x}_t)$ is a minimizer of the expected NELBO, i.e., $\hat{\mathbf{x}}_0^{\theta^\star}(\mathbf{x}_t, t) = \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)$.

This observation clarifies the role of Proposition 1. The difference between the denoiser and the leave-one-out posterior is not caused by the plug-in idea itself, but by the specific extension of the

bridge used to interpret a probability vector as a clean-token argument. The canonical nonlinear extension leads to the leave-one-out target, whereas the barycentric affine extension leads back to the denoiser.

A useful consequence of this observation is that it makes it simple to express the continuous-time ELBO of the marginalization parameterization directly as a plug-in of the denoiser.

We could derive this ELBO by taking $T \to \infty$ in the discrete-time ELBO, but in order to emphasize the role of the extension, we will derive it here using the CTMC point of view (see Appendix F). Again, any extension of the conditional rate matrix that coincides on one-hots gives a similar process. Here the conditional rate of the CTMC, corresponding to the linear-bridge extension, is itself linear and is given by

$$R_t^{\mathrm{lin},\ell}(\tilde{x}|x,\mu) \coloneqq R_t^{\ell}(\tilde{x}|x)\left(1 - \frac{K\alpha_t}{1+(K-1)\alpha_t}\langle x,\mu\rangle + \frac{K\alpha_t}{1-\alpha_t}\langle \tilde{x},\mu\rangle\right).$$

where the UDM forward rate is $R_t^{\ell}(\tilde{x}|x) = \beta_t/K$ with $\beta_t = \frac{-\alpha_t'}{\alpha_t}$. This formula can be derived by linearizing $x_0 \mapsto \frac{q_{t|0}^{\ell}(\tilde{x}|x_0)}{q_{t|0}^{\ell}(x|x_0)}$, or by taking the limit when $s \to t$ in the linear-bridge extension (55).

The corresponding continuous-time marginalization NELBO is (see Appendix F for the general case)

$$\begin{aligned}\mathcal{L}_{\infty}^{\mathrm{marg}}(\theta) \coloneqq{}& \mathbb{E}[\mathrm{KL}(q_{1|0}(\cdot|X_0)\|p_1)] \\ &+ \int_0^1 \frac{\beta_t}{K}\,\mathbb{E}\Bigg[\sum_{\ell=1}^{L}\sum_{\tilde{x}\neq X_t^{\ell}} \Phi\Bigg(1 - \frac{K\alpha_t}{1+(K-1)\alpha_t}\langle X_t^{\ell}, X_0^{\ell}\rangle + \frac{K\alpha_t}{1-\alpha_t}\langle \tilde{x}, X_0^{\ell}\rangle, \\ &1 - \frac{K\alpha_t}{1+(K-1)\alpha_t}\langle X_t^{\ell}, \hat{\mathbf{x}}_0^{\theta}(X_t,t)^{\ell}\rangle + \frac{K\alpha_t}{1-\alpha_t}\langle \tilde{x}, \hat{\mathbf{x}}_0^{\theta}(X_t,t)^{\ell}\rangle\Bigg)\Bigg]\mathrm{d}t.\end{aligned} \tag{56}$$

Here $\Phi(a,b) \coloneqq b - a + a\log(a/b)$, and the expectation is with respect to $X_0 \sim p_0$ and $X_t \sim q_{t|0}(\cdot|X_0)$. Since $R_t^{\mathrm{lin},\ell}(\tilde{x}|x,\mu)$ is affine in μ, the exact rate averaged over the conditional law of X_0^{ℓ} given X_t is obtained by replacing X_0^{ℓ} with the denoising posterior $p_{0|t}^{\ell}(\cdot|X_t)$, which is why (56) targets the denoiser.

In order to understand the duality between linearization of the considered bridge extension and parameterization (marginalization or plug-in), it is worth noting that the same expression can be obtained from the canonical plug-in parameterization if the network is understood to parameterize a denoiser and converted to the leave-one-out representation before it is plugged into the bridge. More precisely, let $\hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)$ be obtained from $\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)$ by the inverse conversion (25). In UDM, substituting this $\hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)$ into the canonical plug-in CTMC rate $R_t^{\ell}(\tilde{x}|\mathbf{x}_t^{\ell})q_{t|0}^{\ell}(\tilde{x}|\hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)^{\ell})/q_{t|0}^{\ell}(\mathbf{x}_t^{\ell}|\hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)^{\ell})$ gives exactly $R_t^{\mathrm{lin},\ell}(\tilde{x}|x,\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)^{\ell})$. Thus the linear-bridge marginalization ELBO and the $T \to \infty$ limit of the converted plug-in ELBO are the same generalized-KL objective, written in denoiser coordinates.

E Leave-one-out and predictor-corrector

Gibbs samplers. Let π be a probability distribution on a product space $\mathcal{X}_1 \times \cdots \times \mathcal{X}_L$. For each coordinate $\ell \in \{1,\dots,L\}$, denote by $\pi(x^{\ell} \mid x^{-\ell})$ the conditional distribution of the ℓ-th coordinate given all the others. The Gibbs sampler associated with π is the Markov chain obtained by repeatedly choosing a coordinate ℓ and replacing only x^{ℓ} by a fresh draw from $\pi(\cdot \mid x^{-\ell})$, while keeping $x^{-\ell}$ fixed. Equivalently, the one-coordinate Gibbs kernel is

$$K_{\ell}(x,\mathrm{d}y) = \delta_{x^{-\ell}}(\mathrm{d}y^{-\ell})\pi(\mathrm{d}y^{\ell} \mid x^{-\ell}).$$

The invariance of π under K_ℓ follows immediately from the definition of conditional distributions. Indeed, for any bounded measurable test function φ,

$$\begin{aligned}\int \pi(\mathrm{d}x) \int K_\ell(x, \mathrm{d}y)\varphi(y) &= \int \pi(\mathrm{d}x) \int \pi(\mathrm{d}y^\ell \mid x^{-\ell})\varphi(y^\ell, x^{-\ell}) \\ &= \int \pi(\mathrm{d}x^{-\ell}) \int \pi(\mathrm{d}x^\ell \mid x^{-\ell}) \int \pi(\mathrm{d}y^\ell \mid x^{-\ell})\varphi(y^\ell, x^{-\ell}) \\ &= \int \pi(\mathrm{d}x^{-\ell}) \int \pi(\mathrm{d}y^\ell \mid x^{-\ell})\varphi(y^\ell, x^{-\ell}) \\ &= \int \pi(\mathrm{d}y)\varphi(y).\end{aligned}$$

Hence $\pi K_\ell = \pi$. It follows that deterministic scan, which composes the kernels $K_1, \dots, K_L$, and random scan, which averages them, also leave π invariant.

At a fixed time t, we apply this construction with $\pi = p_t$. Identity (35) gives the corresponding one-coordinate conditional explicitly. Starting from a current state $\mathbf{x}_t$, one chooses a coordinate ℓ and resamples only this coordinate according to

$$\mathbf{x}_t^\ell \sim p_t^\ell(\cdot|\mathbf{x}_t^{-\ell}) = q_{t|0}^\ell(\cdot|\hat{\mathbf{x}}_{0|t}^{\mathrm{loo}}(\mathbf{x}_t)^\ell),$$

while keeping $\mathbf{x}_t^{-\ell}$ unchanged. This yields a Gibbs sampler with invariant distribution p_t.

Predictor-corrector samplers. Predictor-corrector schemes were introduced in the continuous setting in [33]. They alternate a predictor step, which moves along the reverse dynamics, and a corrector step, which applies an MCMC kernel preserving the current marginal. They make use of the fact that during training, we learn the approximate score $s_\theta(t, \mathbf{x}_t) \approx \nabla \log p_t(\mathbf{x}_t)$ to construct a Langevin dynamic $dX_t = s_\theta(t, \mathbf{x}_t)dt + \sqrt{2}dW_t$ which approximately leaves p_t invariant. In the discrete setting, [2] proposed an analogue based on the reverse CTMC for the predictor and on a forward-backward CTMC for the corrector. The role of the corrector is to improve the sample quality at a fixed time before continuing the reverse trajectory. The identity (35) provides another such corrector: instead of using an uninformed transition that preserves p_t, one may apply Gibbs updates based on the exact one-coordinate conditionals of p_t.

Informed Gibbs corrector. This observation is particularly useful in UDM. Indeed, Proposition 1 shows that, under the Bridge Plug-in parameterization, the network already learns the leave-one-out object needed to evaluate the Gibbs conditional. Therefore, unlike [40], no auxiliary leave-one-out model is required. Among exact Gibbs schemes, deterministic scan and random scan are the canonical choices. In practice, however, it is often more efficient to concentrate the corrector effort on the positions where the predictor is least confident. Following [40], we therefore use the margin score

$$c_\ell^{\mathrm{margin}}(\mathbf{x}_t) \coloneqq \log p_t^\ell(\mathbf{x}_t^\ell|\mathbf{x}_t^{-\ell}) - \max_{x \neq \mathbf{x}_t^\ell} \log p_t^\ell(x|\mathbf{x}_t^{-\ell}),$$

which measures the gap between the current token and the most likely alternative. Low values of $c_\ell^{\mathrm{margin}}(\mathbf{x}_t)$ correspond to 'unlikely' positions, and the corrector updates the coordinates with the smallest margins first. This is the practical scheme used in our experiments. As in [40], one may also update several coordinates in parallel, in the spirit of Hogwild Gibbs sampling [14]. This parallel version should be viewed as a heuristic acceleration of the exact single-coordinate Gibbs corrector.

Figure 3 shows that the Gibbs corrector significantly improves sample quality at a comparable computational cost. As already emphasized in [2], predictor-corrector schemes are often particularly effective in the discrete setting. The reason is that a corrector step may modify the value of a token altogether, rather than merely perturbing the state locally. As a consequence, correcting a few uncertain positions can substantially alter the subsequent reverse trajectory.

This effect is typically stronger than in continuous domains, where corrector steps usually act as local refinements of an already smooth state. In discrete domains, by contrast, the uncertainty is concentrated on a finite set of alternatives, and resolving it can produce qualitatively different samples. This explains why corrector-based methods have become a central tool in discrete diffusion models; see for instance [2, 37, 40].

Algorithm 5 Predictor-corrector sampler with margin-based Gibbs corrector

Require: time grid $0 = t_0 < t_1 < \cdots < t_n = 1$
Require: integers $(m_i)_{i=1}^n$, number of parallel updates k, and a model providing $p^\theta_{s_i|t_i}$ and $\hat{\mathbf{x}}^{\mathrm{loo}}_{0|s_i}(\cdot)$
Ensure: a sample approximately distributed according to p_0

Draw $\mathbf{x}_{t_n} \sim p_1$
for $i = n, n-1, \dots, 1$ **do**
  Draw predictor state $\mathbf{x}_{s_i} \sim p^\theta_{s_i|t_i}(\cdot|\mathbf{x}_{t_i})$
  for $r = 1, \dots, m_i$ **do**
    for $\ell = 1, \dots, L$ **do** ▷ in parallel over ℓ
      Define $p^\ell_{s_i}(\cdot|\mathbf{x}^{-\ell}_{s_i}) \leftarrow q^\ell_{s_i|0}(\cdot|\hat{\mathbf{x}}^{\mathrm{loo}}_{0|s_i}(\mathbf{x}_{s_i})^\ell)$
      Compute $c_\ell \leftarrow \log p^\ell_{s_i}(\mathbf{x}^\ell_{s_i}|\mathbf{x}^{-\ell}_{s_i}) - \max_{x \neq \mathbf{x}^\ell_{s_i}} \log p^\ell_{s_i}(x|\mathbf{x}^{-\ell}_{s_i})$
    end for
    Let $\ell_{r,1}, \dots, \ell_{r,k}$ be the coordinates with smallest scores c_ℓ
    for $j = 1, \dots, k$ **do**
      Resample $\mathbf{x}^{\ell_{r,j}}_{s_i} \sim p^{\ell_{r,j}}_{s_i}(\cdot|\mathbf{x}^{-\ell_{r,j}}_{s_i})$ in parallel
    end for
  end for
  Set $\mathbf{x}_{t_{i-1}} := \mathbf{x}_{s_i}$
end for
return $\mathbf{x}_0$

E.1 On the distinction between LOO and denoiser in [8]

In [8], the authors introduce, for each position ℓ, a family of superposition posteriors which, in the notation of the present paper, can be written as

$$\Psi^\ell_{s|0,t}(\cdot|\mathbf{x}^\ell_0, \mathbf{x}^\ell_t) := \kappa_t q^\ell_{s|0,t}(\cdot|\mathbf{x}^\ell_0, \mathbf{x}^\ell_t) + (1-\kappa_t) q^\ell_{s|0}(\cdot|\mathbf{x}^\ell_0) \,. \tag{57}$$

The first term is the usual reverse bridge and the second term is a forward noising step. After averaging (57) against the clean-token posterior, one obtains a reverse kernel with the same marginals as the original diffusion. In the masked case, different choices of κ_t recover ReMDM and related predictor-corrector samplers [2, 37]. The reverse transition is given by

$$\Psi_{s|t}(\mathbf{x}_s|\mathbf{x}_t) = \textstyle\sum_{\mathbf{x}^\ell_0} \Psi^\ell_{s|0,t}(\mathbf{x}_s|\mathbf{x}^\ell_0, \mathbf{x}^\ell_t)\, p^\ell_{0|t}(\mathbf{x}^\ell_0|\mathbf{x}_t) \,,$$

Following Proposition 1, since DUO [8, 29] trains a LOO denoiser similarily to [30], the useful rewriting of the previous transition is

$$\Psi_{s|t}(\mathbf{x}_s|\mathbf{x}_t) = \kappa_t q_{s|0,t}(\mathbf{x}_s|\hat{\mathbf{x}}^{\mathrm{loo}}_0(\mathbf{x}_t, t), \mathbf{x}_t) + (1-\kappa_t) q_{s|0}(\mathbf{x}_s|\hat{\mathbf{x}}_0(\mathbf{x}_t, t)) \tag{58}$$

and $\hat{\mathbf{x}}_0(\mathbf{x}_t, t)$ is the plain denoiser. On the other hand [8, Eqn 12], plugs inside the forward transition $q_{s|0}$ the quantity $q_{0|0,t}(\cdot|\hat{\mathbf{x}}^{\mathrm{loo}}_0(\mathbf{x}_t, t), \mathbf{x}_t)$, *i.e.* (3) with $s = 0$. It turns out that this is equivalent to the LOO denoiser to denoiser conversion formula (26) and so [8] does indeed implement the transition (58). This is however valid only if we train the network to approximate a LOO denoiser. If instead we train a plain denoiser, then we need to first convert it into a LOO denoiser using (27) in order to plug it in the bridge as in (58).

F Continuous time perspective

Here we consider a CTMC $(X_t)_{t\in[0,1]}$ on X with infinitesimal generator R_t defined, for any $\mathbf{x}, \tilde{\mathbf{x}} \in \mathsf{X}$ as

$$R_t(\tilde{\mathbf{x}}|\mathbf{x}) := \begin{cases} R^\ell_t(\tilde{\mathbf{x}}^\ell|\mathbf{x}^\ell) & \text{if } \mathbf{x}^{-\ell} = \tilde{\mathbf{x}}^{-\ell} \text{ and } \mathbf{x}^\ell \neq \tilde{\mathbf{x}}^\ell \\ 0 & \text{otherwise,} \end{cases} \tag{59}$$

and for all $\ell \in [1:L]$, R^ℓ_t is the generator of the CTMC $(X^\ell_t)_{t\in[0,1]}$ on V thus satisfying $R^\ell_t(\tilde{x}|x) \geq 0$ if $x \neq \tilde{x}$ and $\sum_{\tilde{x}\in\mathsf{V}} R^\ell_t(\tilde{x}|x) = 0$. Writing $\beta_t := -\alpha'_t/\alpha_t \geq 0$, we consider the following form of the token-wise generator

$$R^\ell_t(\tilde{x}|x) = \beta_t \langle \tilde{x}, \pi \rangle \quad \text{for } x \neq \tilde{x}, \qquad R^\ell_t(x|x) = -\beta_t(1 - \langle x, \pi \rangle) \,.$$

Then, since $\exp(-\int_s^t \beta_u \mathrm{d}u) = \alpha_{t|s}$ and using the Kolmogorov forward equation we find that the resulting conditional law of the sequence-level CTMC satisfies $X_t \mid X_s = \mathbf{x}_s \sim q_{t|s}(\cdot|\mathbf{x}_s)$ for all $0 \le s < t \le 1$. With $R_t^\ell(\tilde{x}|x) = \beta_t/K$, for all $\tilde{x} \neq x$, we recover the UDM transition. For the MDM transitions, the generator is $R_t^\ell(e_K|x) = \beta_t$ for $x \neq e_K$ and all other off-diagonal rates vanish. The reverse CTMC starts from p_1 at time 1 and evolves backward to reach p_0 at time 0. Its backward transition kernels are $p_{s|t}(\mathbf{x}_s|\mathbf{x}_t)$, and the off-diagonal elements of its generator are given by $\bar{R}_t(\tilde{\mathbf{x}}|\mathbf{x}) \coloneqq R_t(\mathbf{x}|\tilde{\mathbf{x}})\, p_t(\tilde{\mathbf{x}})/p_t(\mathbf{x})$. Next, the generator of the chain $(X_t)_{t\in[0,1]}$ running backwards and conditioned on $X_0 = \mathbf{x}_0$ is given by

$$\bar{R}_t(\tilde{\mathbf{x}}|\mathbf{x}_0, \mathbf{x}) \coloneqq \begin{cases} \beta_t \langle \mathbf{x}^\ell, \pi\rangle \cdot \langle \tilde{\mathbf{x}}^\ell, s_t(\mathbf{x}^\ell|\mathbf{x}_0^\ell)\rangle & \text{if } \mathbf{x}^{-\ell} = \tilde{\mathbf{x}}^{-\ell} \text{ and } \mathbf{x}^\ell \neq \tilde{\mathbf{x}}^\ell \\ 0 & \text{otherwise,} \end{cases}$$

where we define the conditional score

$$s_t(x|x_0) \coloneqq (\alpha_t x_0 + (1-\alpha_t)\pi)/q_{t|0}(x|x_0) \tag{60}$$

if $q_{t|0}(x|x_0) > 0$ and 0 otherwise. Following [2], we have the following identity

$$\bar{R}_t(\tilde{\mathbf{x}}|\mathbf{x}) \coloneqq \begin{cases} \beta_t \langle \mathbf{x}^\ell, \pi\rangle \cdot \langle \tilde{\mathbf{x}}^\ell, \mathbf{s}_t(\mathbf{x})^\ell\rangle & \text{if } \mathbf{x}^{-\ell} = \tilde{\mathbf{x}}^{-\ell} \text{ and } \mathbf{x}^\ell \neq \tilde{\mathbf{x}}^\ell \\ 0 & \text{otherwise,} \end{cases} \tag{61}$$

where $\mathbf{s}_t(\mathbf{x}) \in \mathbb{R}_{\ge 0}^{L\times K}$ and its ℓ-th row is

$$\mathbf{s}_t(\mathbf{x})^\ell \coloneqq \sum_{\mathbf{x}_0^\ell} s_t(\mathbf{x}^\ell|\mathbf{x}_0^\ell) p_{0|t}^\ell(\mathbf{x}_0^\ell|\mathbf{x}). \tag{62}$$

Following SEDD, we consider next the parameterized reverse generator $\bar{R}_t^\theta$ defined similarly to (61) but with the intractable score $\mathbf{s}_t$ replaced with a model $\mathbf{s}_t^\theta$. Following [2, 18], the continuous-time ELBO specialized to the generator (61) is

$$\begin{aligned} \mathcal{L}_\infty(\theta) \coloneqq\ & \mathbb{E}[\mathrm{KL}(q_{1|0}(\cdot|X_0)\|p_1)] \\ & + \int_0^1 \beta_t\, \mathbb{E}\Bigg[\sum_{\ell=1}^L \langle X_t^\ell, \pi^\ell\rangle \sum_{\tilde{x}\neq X_t^\ell} \Phi\Big(\langle \tilde{\mathbf{x}}^\ell, s_t(X_t^\ell|X_0^\ell)\rangle, \langle \tilde{\mathbf{x}}^\ell, \mathbf{s}_t^\theta(X_t)^\ell\rangle \Big) \Bigg] \mathrm{d}t\ . \end{aligned} \tag{63}$$

where $\Phi(a,b) \coloneqq b - a + a\log(a/b)$ for $a, b > 0$, with the usual convention $0 \cdot \log 0 = 0$, and both expectations are w.r.t. the joint law $X_0 \sim p_0$ and $X_t \sim q_{t|0}(\cdot|X_0)$.

Sampling from a CTMC Sampling from the reverse CTMC can be done exactly with the Gillespie algorithm, but this updates only one position at a time. In practice one introduces a decreasing grid $1 = t_N > \cdots > t_0 = 0$, writes $\Delta_i \coloneqq t_i - t_{i-1}$, and approximates the reverse dynamics over each interval $[t_{i-1}, t_i]$. Given the current state $\mathbf{x}_i$ at time t_i, we freeze the model reverse rates over the interval and define

$$\hat{\lambda}_i^\ell(\mathbf{x}_i) \coloneqq \sum_{\tilde{\mathbf{x}}:\, \tilde{\mathbf{x}}^{-\ell} = \mathbf{x}_i^{-\ell},\, \tilde{\mathbf{x}}^\ell \neq \mathbf{x}_i^\ell} \bar{R}_{t_i}^\theta(\tilde{\mathbf{x}}|\mathbf{x}_i), \qquad \hat{\lambda}_i(\mathbf{x}_i) \coloneqq \sum_{\ell=1}^L \hat{\lambda}_i^\ell(\mathbf{x}_i)\ . \tag{64}$$

The simplest discretization is the explicit Euler scheme. It keeps at most one jump over the whole interval and samples

$$p_{i-1|i}^{\mathrm{euler}}(\tilde{\mathbf{x}} \mid \mathbf{x}_i) \coloneqq \mathbb{1}_{\tilde{\mathbf{x}}=\mathbf{x}_i}\big(1 - \Delta_i \hat{\lambda}_i(\mathbf{x}_i)\big) + \Delta_i \bar{R}_{t_i}^\theta(\tilde{\mathbf{x}}|\mathbf{x}_i)\mathbb{1}_{\tilde{\mathbf{x}}\neq\mathbf{x}_i} \tag{Euler}$$

provided $\Delta_i \hat{\lambda}_i(\mathbf{x}_i) \le 1$. Equivalently, one keeps $\mathbf{x}_i$ with probability $1 - \Delta_i\hat{\lambda}_i(\mathbf{x}_i)$ and otherwise performs a single jump drawn from the normalized off-diagonal rates of $\bar{R}_{t_i}^\theta$.

Tau-leaping [2] keeps the same frozen rates but allows several positions to jump in parallel. Concretely, for every ℓ and every $\tilde{x} \neq \mathbf{x}_i^\ell$, one draws independent Poisson variables

$$N_{i,\ell,\tilde{x}} \sim \mathrm{Poisson}\big(\Delta_i \bar{R}_{t_i}^\theta((\mathbf{x}_i^{-\ell}, \tilde{x})|\mathbf{x}_i)\big)$$

and sets

$$\mathbf{x}_{i-1}^\ell \coloneqq \begin{cases} \mathbf{x}_i^\ell & \text{if } \sum_{\tilde{x}\neq\mathbf{x}_i^\ell} N_{i,\ell,\tilde{x}} \neq 1, \\ \sum_{\tilde{x}\neq\mathbf{x}_i^\ell} N_{i,\ell,\tilde{x}}\tilde{x} & \text{if } \sum_{\tilde{x}\neq\mathbf{x}_i^\ell} N_{i,\ell,\tilde{x}} = 1, \end{cases} \tag{τ-leaping}$$

independently over $\ell \in [\![1, L]\!]$. Thus, if exactly one jump occurs at a position, the token is replaced by its destination; if no jump or several jumps occur, the update is rejected at that position.

G Uniform Diffusion and Maximal Coupling

G.1 Maximal couplings

Let P and Q be two probability distributions on a finite state space $\mathcal{X}$. A coupling of (P, Q) is a probability distribution γ on $\mathcal{X}^2$ such that

$$\sum_{y\in\mathcal{X}} \gamma(x, y) = P(x), \qquad \sum_{x\in\mathcal{X}} \gamma(x, y) = Q(y).$$

If $(X, Y) \sim \gamma$, a maximal coupling is, by definition, a coupling that minimizes the mismatch probability $\mathbb{P}(X \neq Y)$ over all couplings of (P, Q), namely $\inf_{\gamma\in\Gamma(P,Q)} \mathbb{P}(X \neq Y)$, where $\Gamma(P, Q)$ denotes the set of couplings of (P, Q). It is standard that

$$\inf_{\gamma\in\Gamma(P,Q)} \mathbb{P}(X \neq Y) = \|P - Q\|_{\mathrm{TV}}.$$

Equivalently, since $\mathbb{P}(X \neq Y) = 1 - \mathbb{P}(X = Y)$, maximal couplings are exactly those that maximize the coincidence probability. In the discrete setting this is equivalent to

$$\sum_{x\in\mathcal{X}} \gamma(x, x) = \sum_{x\in\mathcal{X}} \min\{P(x), Q(x)\} = 1 - \|P - Q\|_{\mathrm{TV}}.$$

G.2 Maximal couplings and masked diffusion

Masked diffusion provides a simple example of a maximal coupling at the token level. Fix $\ell \in [\![1, L]\!]$, $0 \leq s < t \leq 1$, and $x_0 \neq \mathsf{m}$. The MDM forward transition is

$$q^\ell_{t|0}(x_t|x_0; \mathsf{m}) = \mathrm{Cat}(x; \alpha_t x_0 + (1 - \alpha_t)\mathsf{m}) = \alpha_t \mathbb{1}_{x=x_0} + (1 - \alpha_t)\mathbb{1}_{x=\mathsf{m}} .$$

Consider the joint distribution with p.m.f.

$$q^\ell_{s,t|0}(x_s, x_t|x_0; \mathsf{m}) \coloneqq q^\ell_{s|0,t}(x_s|x_0, x_t; \mathsf{m})\, q^\ell_{t|0}(x_t|x_0; \mathsf{m}), \tag{65}$$

which is explicitly

$$q^\ell_{s,t|0}(x_s, x_t|x_0; \mathsf{m}) = \alpha_t \mathbb{1}_{x_s=x_0}\mathbb{1}_{x_t=x_0} + (\alpha_s - \alpha_t)\mathbb{1}_{x_s=x_0}\mathbb{1}_{x_t=\mathsf{m}} + (1 - \alpha_s)\mathbb{1}_{x_s=\mathsf{m}}\mathbb{1}_{x_t=\mathsf{m}} .$$

Therefore its coincidence probability is

$$\textstyle\sum_{x\in\mathsf{V}} q^\ell_{s,t|0}(x, x|x_0; \mathsf{m}) = \alpha_t + 1 - \alpha_s.$$

Since $s < t$ implies $\alpha_s \geq \alpha_t$, the overlap of the two MDM marginals is

$$\textstyle\sum_{x\in\mathsf{V}} \min\big\{q^\ell_{s|0}(x|x_0; \mathsf{m}), q^\ell_{t|0}(x|x_0; \mathsf{m})\big\} = \alpha_t + 1 - \alpha_s.$$

Thus (65) is a maximal coupling between $q^\ell_{s|0}(\cdot|x_0; \mathsf{m})$ and $q^\ell_{t|0}(\cdot|x_0; \mathsf{m})$.

G.3 Maximum coupling bridge and ELBO for uniform diffusion

We now introduce a token-wise bridge for uniform diffusion that is a maximal coupling between the forward marginals while preserving the same one-time marginals as the original forward process. We also derive the corresponding continuous-time reverse rate matrix and the associated CTMC ELBO. Interestingly, in their Appendix A, [32] derive a bridge that can modulate the stochasticity of the reverse process and that recovers the same maximal coupling bridge in the limit where one sets the coefficient σ_t to $\frac{1-\alpha_{t-1}}{1-\alpha_t}$.

Proposition 8. *Fix $\ell \in [\![1, L]\!]$, $x_0 \in \mathsf{V}$ and $0 \leq s < t \leq 1$. Consider the bridge*

$$q^\ell_{s|0,t}(x_s|x_0, x_t) = \mathrm{Cat}\left(x_s; \frac{\alpha_s - \alpha_t}{1 - \alpha_t}x_0 + \frac{1 - \alpha_s}{1 - \alpha_t}x_t\right) . \tag{66}$$

The joint distribution with p.m.f.

$$q^\ell_{s,t|0}(x_s, x_t|x_0) \coloneqq q^\ell_{s|0,t}(x_s|x_0, x_t)q^\ell_{t|0}(x_t|x_0)$$

satisfies

$$q^{\ell}_{s,t|0}(x_s, x_t|x_0) = \alpha_t \mathbb{1}_{x_s=x_0}\mathbb{1}_{x_t=x_0} + \frac{\alpha_s - \alpha_t}{K}\mathbb{1}_{x_s=x_0} + \frac{1-\alpha_s}{K}\mathbb{1}_{x_t=x_s}. \tag{67}$$

Moreover, it preserves the forward marginals,

$$\sum_{x_s\in\mathsf{V}} q^{\ell}_{s,t|0}(x_s, x_t|x_0) = q^{\ell}_{t|0}(x_t|x_0), \qquad \sum_{x_t\in\mathsf{V}} q^{\ell}_{s,t|0}(x_s, x_t|x_0) = q^{\ell}_{s|0}(x_s|x_0),$$

and is a maximal coupling between $q^{\ell}_{s|0}(\cdot|x_0)$ and $q^{\ell}_{t|0}(\cdot|x_0)$.

Proof. If $x_t = x_0$, the bridge in (66) is deterministic and therefore

$$q^{\ell}_{s,t|0}(x_s, x_t|x_0) = \left(\alpha_t + \frac{1-\alpha_t}{K}\right)\mathbb{1}_{x_s=x_0}\mathbb{1}_{x_t=x_0}.$$

If $x_t \neq x_0$, then

$$q^{\ell}_{t|0}(x_t|x_0) = \frac{1-\alpha_t}{K},$$

so multiplying the second line of (66) by $q^{\ell}_{t|0}(x_t|x_0)$ gives

$$q^{\ell}_{s,t|0}(x_s, x_t|x_0) = \frac{\alpha_s - \alpha_t}{K}\mathbb{1}_{x_s=x_0} + \frac{1-\alpha_s}{K}\mathbb{1}_{x_s=x_t}.$$

Combining the two cases yields (67). We next verify the marginals. Summing (67) over x_s gives

$$\begin{aligned}\sum_{x_s\in\mathsf{V}} q^{\ell}_{s,t|0}(x_s, x_t|x_0) &= \alpha_t\mathbb{1}_{x_t=x_0} + \frac{\alpha_s-\alpha_t}{K} + \frac{1-\alpha_s}{K} \\ &= \alpha_t\mathbb{1}_{x_t=x_0} + \frac{1-\alpha_t}{K} = q^{\ell}_{t|0}(x_t|x_0).\end{aligned}$$

Summing instead over x_t gives

$$\begin{aligned}\sum_{x_t\in\mathsf{V}} q^{\ell}_{s,t|0}(x_s, x_t|x_0) &= \alpha_t\mathbb{1}_{x_s=x_0} + \frac{K(\alpha_s-\alpha_t)}{K}\mathbb{1}_{x_s=x_0} + \frac{1-\alpha_s}{K} \\ &= \alpha_s\mathbb{1}_{x_s=x_0} + \frac{1-\alpha_s}{K} = q^{\ell}_{s|0}(x_s|x_0).\end{aligned}$$

It remains to prove maximality. Under the coupling above,

$$\sum_{x\in\mathsf{V}} q^{\ell}_{s,t|0}(x, x|x_0) = \alpha_t + (1-\alpha_s) + \frac{\alpha_s-\alpha_t}{K}.$$

On the other hand, for the two uniform-diffusion marginals we have

$$q^{\ell}_{s|0}(x|x_0) = \alpha_s\mathbb{1}_{x=x_0} + \frac{1-\alpha_s}{K}, \qquad q^{\ell}_{t|0}(x|x_0) = \alpha_t\mathbb{1}_{x=x_0} + \frac{1-\alpha_t}{K},$$

so

$$\begin{aligned}\sum_{x\in\mathsf{V}} \min\{q^{\ell}_{s|0}(x|x_0), q^{\ell}_{t|0}(x|x_0)\} &= \alpha_t + \frac{1-\alpha_t}{K} + (K-1)\frac{1-\alpha_s}{K} \\ &= \alpha_t + (1-\alpha_s) + \frac{\alpha_s-\alpha_t}{K}.\end{aligned}$$

Hence the probability of equality reaches the maximal-coupling upper bound. □

Proposition 9. *Let $\beta_t := -\alpha_t'/\alpha_t$. The token-wise backward generator conditioned on $X^{\ell}_0 = x_0$ and associated with the bridge (66) is, for $\tilde{x} \neq x$,*

$$\bar{R}^{\mathrm{mc}}_t(\tilde{x}|x, x_0) = \frac{\alpha_t\beta_t}{1-\alpha_t}\mathbb{1}_{x\neq x_0}\mathbb{1}_{\tilde{x}=x_0}.$$

The diagonal term is given by row-sum zero. Consequently, the exact reverse generator obtained after averaging against the denoising posterior is

$$\bar{R}^{\mathrm{mc}}_t(\tilde{\mathbf{x}}|\mathbf{x}) = \begin{cases} \frac{\alpha_t\beta_t}{1-\alpha_t} p^{\ell}_{0|t}(\tilde{\mathbf{x}}^{\ell}|\mathbf{x}) & \text{if } \mathbf{x}^{-\ell} = \tilde{\mathbf{x}}^{-\ell} \text{ and } \mathbf{x}^{\ell} \neq \tilde{\mathbf{x}}^{\ell}, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. Fix x_0 and let $h > 0$ be such that $t - h \geq 0$. If $x = x_0$, then the first line of (66) gives

$$q^{\ell}_{t-h|0,t}(\tilde{x}|x_0, x) = \mathbb{1}_{\tilde{x}=x},$$

so there is no off-diagonal jump. If $x \neq x_0$, the bridge gives

$$q^{\ell}_{t-h|0,t}(\tilde{x}|x_0, x) = \frac{\alpha_{t-h} - \alpha_t}{1 - \alpha_t}\mathbb{1}_{\tilde{x}=x_0} + \frac{1 - \alpha_{t-h}}{1 - \alpha_t}\mathbb{1}_{\tilde{x}=x}.$$

Since $\alpha_{t-h} = \alpha_t + h\beta_t\alpha_t + o(h)$, for every $\tilde{x} \neq x$ we obtain

$$\lim_{h\downarrow 0} \frac{q^{\ell}_{t-h|0,t}(\tilde{x}|x_0, x)}{h} = \frac{\alpha_t\beta_t}{1 - \alpha_t}\mathbb{1}_{\tilde{x}=x_0}.$$

This proves the conditioned rate formula.

For the exact reverse generator, let $\mathbf{x}, \tilde{\mathbf{x}} \in \mathsf{X}$ differ only at position ℓ, with $\mathbf{x}^{\ell} \neq \tilde{\mathbf{x}}^{\ell}$. By construction,

$$p_{t-h|t}(\tilde{\mathbf{x}}|\mathbf{x}) = \sum_{\mathbf{x}_0} q_{t-h|0,t}(\tilde{\mathbf{x}}|\mathbf{x}_0, \mathbf{x}) p_{0|t}(\mathbf{x}_0|\mathbf{x}).$$

Because the bridge factorizes across positions and $\tilde{\mathbf{x}}$ differs from $\mathbf{x}$ only at ℓ, the only nonzero off-diagonal contribution comes from clean tokens such that $\mathbf{x}_0^{\ell} = \tilde{\mathbf{x}}^{\ell}$. Therefore

$$\lim_{h\downarrow 0} \frac{p_{t-h|t}(\tilde{\mathbf{x}}|\mathbf{x})}{h} = \frac{\alpha_t\beta_t}{1 - \alpha_t} \sum_{\mathbf{x}_0 :\, \mathbf{x}_0^{\ell} = \tilde{\mathbf{x}}^{\ell}} p_{0|t}(\mathbf{x}_0|\mathbf{x}) = \frac{\alpha_t\beta_t}{1 - \alpha_t} p^{\ell}_{0|t}(\tilde{\mathbf{x}}^{\ell}|\mathbf{x}),$$

which is the claimed formula. □

Interestingly, this backward rate matrix coincides with the one obtained in [3]. Even though it arises from a different perspective, in both cases, the effect is to reduce the stochasticity of the reverse process by making the token-wise coupling maximal while preserving the correct forward marginals.

We now derive the corresponding continuous-time ELBO. We parameterize the reverse generator by a denoiser model $\hat{\mathbf{x}}_0^{\theta}(\mathbf{x}, t) \in (\Delta_K)^L$ and define

$$\bar{R}^{\theta}_t(\tilde{\mathbf{x}}|\mathbf{x}) \coloneqq \begin{cases} \frac{\alpha_t\beta_t}{1-\alpha_t}\langle \tilde{\mathbf{x}}^{\ell}, \hat{\mathbf{x}}_0^{\theta}(\mathbf{x}, t)^{\ell}\rangle & \text{if } \mathbf{x}^{-\ell} = \tilde{\mathbf{x}}^{-\ell} \text{ and } \mathbf{x}^{\ell} \neq \tilde{\mathbf{x}}^{\ell}, \\ 0 & \text{otherwise.} \end{cases} \tag{68}$$

Proposition 10. *Up to a θ-independent constant, the continuous-time ELBO associated with the maximal-coupling reverse generator (68) is*

$$\mathcal{L}^{\mathrm{MCUD}}_{\infty}(\theta) \coloneqq -\int_0^1 \tfrac{\alpha_t'}{1-\alpha_t}\, \mathbb{E}\big[\textstyle\sum_{\ell=1}^{L} \big\{1 - \langle X_t^{\ell}, \hat{\mathbf{x}}_0^{\theta}(X_t, t)^{\ell}\rangle \\ - \mathbb{1}_{X_t^{\ell} \neq X_0^{\ell}}\big(1 + \log\langle X_0^{\ell}, \hat{\mathbf{x}}_0^{\theta}(X_t, t)^{\ell}\rangle\big)\big\}\big]\,\mathrm{d}t, \tag{69}$$

where the expectation is w.r.t. $X_0 \sim p_0$ and $X_t \sim q_{t|0}(\cdot|X_0)$.

Proof. By Proposition 9, conditioned on (X_0^{ℓ}, X_t^{ℓ}), the token-wise off-diagonal reverse rate is

$$\frac{\alpha_t\beta_t}{1 - \alpha_t}\mathbb{1}_{X_t^{\ell} \neq X_0^{\ell}}\mathbb{1}_{\tilde{x}=X_0^{\ell}}$$

for every $\tilde{x} \neq X_t^{\ell}$. Therefore, exactly as in the derivation of (63), the contribution of position ℓ at time t to the continuous-time ELBO is

$$\frac{\alpha_t\beta_t}{1 - \alpha_t} \sum_{\tilde{x} \neq X_t^{\ell}} \Phi\Big(\mathbb{1}_{X_t^{\ell} \neq X_0^{\ell}}\mathbb{1}_{\tilde{x}=X_0^{\ell}}, \langle \tilde{x}, \hat{\mathbf{x}}_0^{\theta}(X_t, t)^{\ell}\rangle\Big).$$

If $X_t^{\ell} = X_0^{\ell}$, the first argument of Φ is zero for every $\tilde{x} \neq X_t^{\ell}$, hence

$$\sum_{\tilde{x} \neq X_t^{\ell}} \Phi\big(0, \langle \tilde{x}, \hat{\mathbf{x}}_0^{\theta}(X_t, t)^{\ell}\rangle\big) = \sum_{\tilde{x} \neq X_t^{\ell}} \langle \tilde{x}, \hat{\mathbf{x}}_0^{\theta}(X_t, t)^{\ell}\rangle \\ = 1 - \langle X_t^{\ell}, \hat{\mathbf{x}}_0^{\theta}(X_t, t)^{\ell}\rangle.$$

If $X_t^\ell \neq X_0^\ell$, only the term $\tilde{x} = X_0^\ell$ has a nonzero first argument, and therefore

$$\begin{aligned}
&\sum_{\tilde{x}\neq X_t^\ell} \Phi\Big(\mathbb{1}_{\tilde{x}=X_0^\ell}, \langle \tilde{x}, \hat{\mathbf{x}}_0^\theta(X_t,t)^\ell\rangle\Big) \\
&\qquad = \sum_{\tilde{x}\neq X_t^\ell} \langle \tilde{x}, \hat{\mathbf{x}}_0^\theta(X_t,t)^\ell\rangle - 1 - \log\langle X_0^\ell, \hat{\mathbf{x}}_0^\theta(X_t,t)^\ell\rangle \\
&\qquad = -\langle X_t^\ell, \hat{\mathbf{x}}_0^\theta(X_t,t)^\ell\rangle - \log\langle X_0^\ell, \hat{\mathbf{x}}_0^\theta(X_t,t)^\ell\rangle.
\end{aligned}$$

Combining the two cases gives

$$1 - \langle X_t^\ell, \hat{\mathbf{x}}_0^\theta(X_t,t)^\ell\rangle - \mathbb{1}_{X_t^\ell \neq X_0^\ell}\big(1 + \log\langle X_0^\ell, \hat{\mathbf{x}}_0^\theta(X_t,t)^\ell\rangle\big).$$

Multiplying by $\alpha_t\beta_t/(1-\alpha_t) = -\alpha_t'/(1-\alpha_t)$ and summing over ℓ yields exactly (69). □

H Related Works

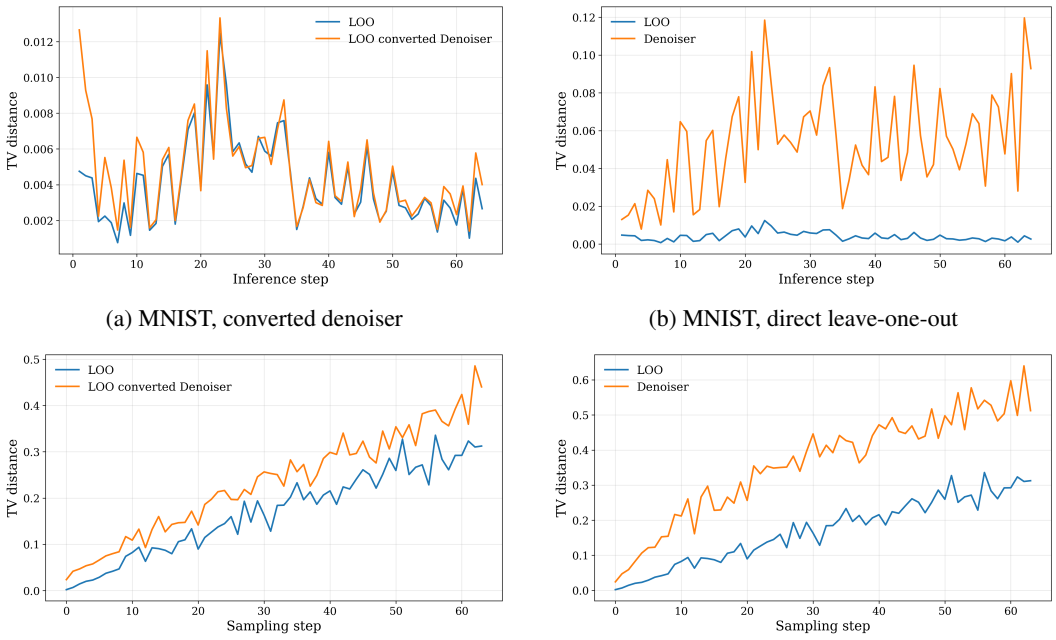
Leave-one-out denoiser. *Categorical ratio matching* [34] trains discrete diffusion models by learning the concrete score $\mathbf{s}_t$; by (11), learning the conditional distribution in (14), which is also leave-one-out, is enough to recover the score and has motivated the use of the Hollow Transformer architecture. As recalled in Appendix A.4, using the Hollow Transformer is essential for their loss function to recover the correct conditional distribution; otherwise, the model can simply learn the identity function. Here we show instead that standard UDMs trained with the bridge parameterization and ELBO already target the same leave-one-out object, and we propose a simple cross-entropy objective whose *unique* minimizer is the leave-one-out conditional distribution, without imposing architectural constraints.

Predictor-corrector samplers. Predictor-corrector samplers alternate a reverse transition step with a corrector kernel that preserves the marginal p_t. For discrete diffusion, [2] propose an uninformed birth-death corrector whose rates combine the forward and reverse generators, and ReMDM [37] introduces remasking in masked diffusion, which can be read as predictor steps interleaved with a single corrector step. Closer to our setting, [40] train an auxiliary leave-one-out model to build an informed Gibbs corrector. For UDMs, our key observation is that, leveraging the leave-one-out representation, this correction does not require a separately trained model. Finally, in the recent work [8] the authors propose the Ψ-samplers framework, which recovers specific instances of known predictor-corrector samplers. Interestingly, when specialized to UDMs, their sampler assumes that the trained network has learned a LOO denoiser and thus does not work when the network has learned a denoiser instead. We provide in Appendix E.1 the general form that applies in both cases.

UDMs with augmented states. Several recent works also augment the state space of UDMs, but for different purposes. [42] introduces routing variables that decide, in a confidence-dependent way, which positions should be treated as already clean during sampling. [17] conditions on a binary corruption mask and learns a separate planner that chooses which positions to denoise as well as a mask-conditioned denoiser, leading to a planner-denoiser factorization. [5] instead uses latent transition times, which reveals an exact connection between UDMs and masked denoising, although the implemented sampler ultimately plugs in an ordinary denoiser that is not conditioned on these times. Our absorbing-state construction is different in spirit: the auxiliary variable is part of an exact lifted model, and the resampling step is chosen so that marginalizing the augmentation recovers the UDM reverse-chain law rather than only providing a decoding heuristic. We give the detailed comparison in Appendix B.3.

I Leave-one-out and Hollow Transformers

As noted in Section 3.2, the ℓ-th component of the leave-one-out posterior is independent of the local observation $\mathbf{x}_t^\ell$. This is a necessary structural property of the optimum identified in Proposition 1. It therefore has two practical consequences. First, it provides a direct diagnostic for suboptimality: after converting a trained model to the leave-one-out representation, any residual dependence of the



(a) MNIST, converted denoiser

(b) MNIST, direct leave-one-out

(c) OWT, converted denoiser

(d) OWT, direct leave-one-out

Figure 5: Sensitivity of the leave-one-out prediction to the local observation $\mathbf{x}_t^\ell$. On MNIST, the converted denoiser and the directly trained leave-one-out model are both nearly insensitive to $\mathbf{x}_t^\ell$. On OWT, the sensitivity remains visible for both models, but it is lower for the directly trained leave-one-out predictor than for the converted denoiser. This indicates that the leave-one-out invariance is not automatically recovered in the high-dimensional regime, and that proximity to this invariance is informative about the quality of the attained optimum.

ℓ-th output on $\mathbf{x}_t^\ell$ certifies that the model has not reached the leave-one-out optimum. Second, it suggests an architectural design: instead of hoping that optimization learns this invariance, one may try to enforce it by construction.

LOO sensitivity diagnostic. We check the sensitivity of the leave-one-out posterior to the local observation $\mathbf{x}_t^\ell$ on some of the models that we have trained. More precisely, we vary $\mathbf{x}_t^\ell$ while keeping $\mathbf{x}_t^{-\ell}$ fixed and measure the induced change in the ℓ-th leave-one-out prediction. The results are reported in Figure 5.

On low-dimensional examples such as MNIST, the leave-one-out posterior is indeed nearly insensitive to the local observation. Moreover, the denoiser converted into the leave-one-out representation and the leave-one-out predictor trained directly produce almost identical sensitivity curves. This strongly suggests that both models are already close to their respective optima, in which case the conversion formula of Proposition 4 is effectively exact.

On larger sequence models, such as the transformer trained on OWT, the sensitivity remains clearly nonzero. This shows that in realistic high-dimensional settings the leave-one-out invariance is not automatically recovered by finite-capacity models trained with standard objectives. More generally, the sensitivity diagnostic appears to be well aligned with downstream performance: models that are closer to the leave-one-out optimum (using the local sensitivity as proxy) also tend to be the ones that perform better generatively.

This observation naturally suggests enforcing the invariance directly in the architecture. For transformers, however, doing so while preserving access to the rest of the context is not entirely straightforward.

Hollow transformer. A proposal for such an architecture is the hollow transformer introduced by [34] and later used by [40]. The idea is to combine two autoregressive streams, one left-to-right and

Table 2: Training setup for the language-model experiments.

Setting	LM1B / OWT
Backbone	DDiT
Hidden size	768
Conditioning dimension	128
Number of blocks	12
Number of heads	12
Dropout	0.1
Training steps	10^6
Global batch size	512
Optimizer	Adam
Learning rate	3×10^{-4}
$\beta_1, \beta_2, \epsilon$	$0.9, 0.999, 10^{-8}$
Weight decay	0
Learning-rate schedule	constant with 2500 warmup steps
Precision	bfloat16
Gradient clipping	1.0
EMA decay	0.9999
Antithetic time sampling	yes
Sampling floor	10^{-3}
Noise schedule	linear schedule

one right-to-left, and to offset the representations so that the output at position ℓ never attends to the input token at the same position, while still depending on all the other positions. Equivalently, in the continuous relaxation used to describe the architecture, the ℓ-th output is constrained to satisfy $\partial f(\mathbf{x}_t, t)^\ell / \partial \mathbf{x}_t^\ell = 0$. This makes the architecture structurally compatible with the leave-one-out target.

In [40], this construction is used in the absorbing setting to learn the additional leave-one-out information needed by the informed corrector without modifying the ELBO objective. The same motivation applies here: if the target at position ℓ should ignore $\mathbf{x}_t^\ell$, it is appealing to remove that dependence by design rather than by optimization.

This inductive bias, however, is not free. Besides requiring two autoregressive streams, the hollow transformer also restricts the information flow relative to a standard transformer, since the output at position ℓ cannot aggregate the full context in a single attention step. More precisely, when computing the ℓ-th output, the tokens to the left and to the right of ℓ interact within their respective groups, but cross-group interactions can only occur through deeper layers. This weakens the directness of the attention mechanism and makes the prediction problem harder. In our own experiments on LM1B, this translated into worse optimization and worse final performance than the corresponding standard DiT models, in line with the observations of [40]. We therefore do not use the hollow transformer in the main experiments. Our view is that it remains a conceptually clean way to enforce the leave-one-out structure, but in large-scale language modeling this structural benefit can be outweighed by the optimization cost.

J Experimental Details

J.1 Setup and hyperparameters

We report here the details relevant to the language-model experiments. We use the same setting as [18, 28]. All models are trained with the same DDiT backbone and the same optimization scheme, and the only dataset-specific architectural change is the sequence length. Both LM1B and OWT are trained from scratch for 10^6 steps with global batch size 512, Adam with learning rate 3×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, no weight decay, gradient clipping at 1.0, and a constant learning-rate schedule with 2500 warmup steps. Training uses bfloat16, exponential moving average with decay 0.9999, antithetic time sampling, and sampling floor $\epsilon = 10^{-3}$. Unless stated otherwise, all experiments use seed 1. For generative evaluation we report GPT-2 Large generative perplexity. The sampling uses higher precision float64. These hyperparameters are summarized in Table 2.

Table 3: Sampling runtime in seconds per sample for the predictor-only temperature and nucleus frontiers, and the predictor-corrector runs. Nucleus is split into filtered top-p sampling ($p < 1$) and the unfiltered case ($p = 1$).

Method	NFE 128	NFE 256	NFE 512	NFE 1024
Temperature	8.512	15.895	30.602	60.316
Nucleus ($p = 1$)	8.410	15.883	30.402	59.492
Nucleus ($p < 1$)	27.141	52.969	104.504	207.531
Predictor-corrector	7.621	13.945	26.543	53.199

Table 4: Modeling choices for considered methods.

Method family	Training loss	Prediction target
UDM	cross-entropy or ELBO	denoiser or leave-one-out
Max-coupling UDM	ELBO	denoiser or leave-one-out
MDM	cross-entropy or ELBO	denoiser
AUDM	ELBO	denoiser
ReAUDM	ELBO	denoiser

Table 5: Sudoku experiment setup and sweep ranges.

Quantity	Value
Task	9×9 Sudoku completion
Training split	50k puzzles
Model	DiT, about 6M parameters
Learning-rate sweep	$\{10^{-4}, 3{\times}10^{-4}, 5{\times}10^{-4}, 7{\times}10^{-4}, 10^{-3}\}$
NFE sweep	$\{16, 32, 64, 128\}$
Evaluation	100 repeats with 1000 validation puzzles
Metric	solve rate, reported as Sudoku accuracy

LM1B is trained and evaluated on the same dataset family with the BERT tokenizer, with sequence length $L = 128$. OWT is trained on the OpenWebText train split and validated on the held-out OpenWebText validation split with the GPT-2 tokenizer, with sequence length $L = 1024$. The zero-shot perplexity evaluation of Table 1 uses the OWT checkpoints and evaluates them on OWT together with AG News [39], PubMed [7], Lambada [24], WikiText [20], PTB [19], and LM1B retokenized with GPT-2, all at sequence length 1024. We use sentence packing for all datasets.

In the absorbing-uniform variants, the noised data and the uniform noise variable (x_t, u_t) are first embedded with the same embedding table with hidden size d, concatenated together and with $\mathbb{1}_{x_t=u_t}$, forming a vector of dimension $2d + 1$, and then passed through an MLP with hidden dimension $4d$ and output dimension d. This adds 7M parameters, around 4% of the total parameter count.

Across all comparisons, we keep the previous training setup fixed and vary only the modeling choices that are central to the paper: the training loss, the prediction target and the associated reverse-process parameterization. The denoiser-based UDM baseline uses the standard denoiser prediction, while the leave-one-out model replaces this with the leave-one-out target. The maximal-coupling models change only the path coupling. The masked baseline uses the standard absorbing-state parameterization of [28] with carry-over unmasking and zero-masking probability. We provide a summary of the available modeling choices in Table 4.

The sampling frontiers are obtained either with temperature sampling or top-p sampling [12] at fixed checkpoint, and plotting generative perplexity against the resulting entropy. For top-p frontiers we use $p \in \{0.80, 0.85, 0.90, 0.92, 0.94, 0.96, 0.98, 1.00\}$ and $\mathrm{NFE} \in \{8, 16, 32, 64, 128, 256, 512, 1024\}$. For temperature frontiers we use $T \in \{0.80, 0.82, \ldots, 1.10\}$ over the same NFE grid. The predictor-corrector experiments are run on OWT with the confidence-based corrector described in the paper, sweeping the number of corrector steps $M \in [\![1, 5]\!]$ and the number of parallel updates $k \in [\![1, 16]\!]$ for $\mathrm{NFE} \in \{64, 128, 256, 512, 1024\}$.

The training time for each OWT run is approximately 1000 H100 hours. For LM1B, it is approximately 250 H100 hours. Each generative frontier evaluations compute time is approximately 1 hour on a H100 GPU.

J.2 Additional results

Temperature and top-p sampling We report the temperature frontier across NFEs $\{16, 128, 512\}$ in Figure 7. The full top-p frontiers across all NFEs are reported in Figure 8. They confirm the same trend as in the main text: for uniform diffusion, the leave-one-out parameterization is consistently preferable in the low-entropy regime. For the maximal-coupling variants the difference between the two parameterizations is smaller but shows a similar trend.

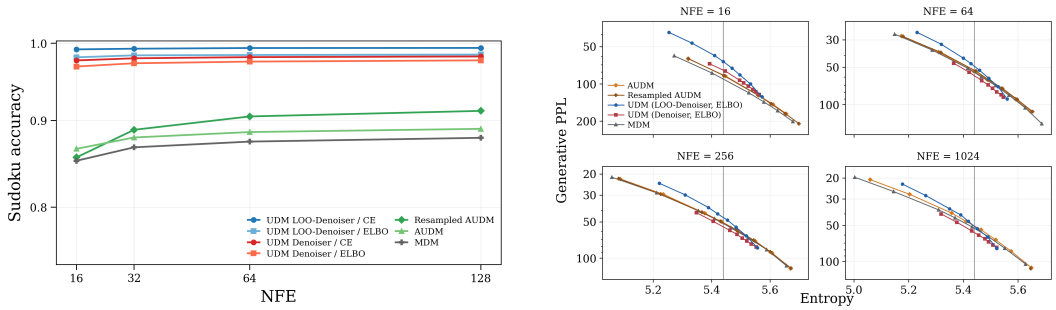


Figure 6: **Left:** Sudoku solve rate as a function of NFE for the best learning rate of each method. **Right:** AUDM, resampled AUDM, UDM, and MDM Gen-PPL frontier. The vertical line corresponds to the dataset entropy. These results are discussed in Section 5.2.

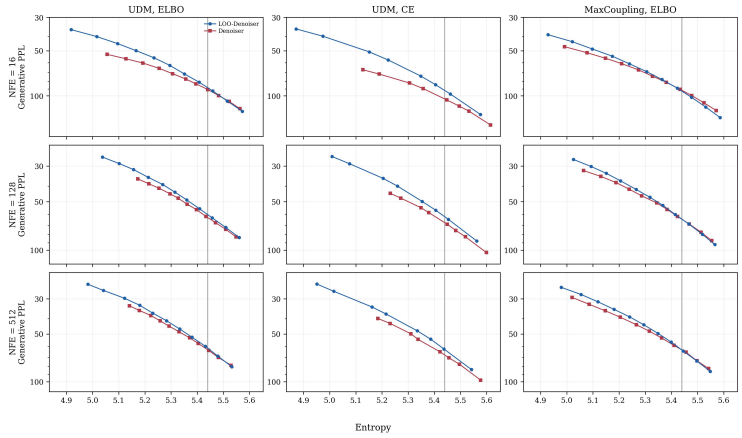


Figure 7: Generative frontiers with varying temperature across NFEs for the denoiser and leave-one-out parameterizations.

Applying nucleus sampling after converting a denoiser prediction to the leave-one-out representation also improves the frontier, as shown in Figure 2. The converted denoiser closes a substantial part of the gap to the model trained directly with the leave-one-out target, while remaining clearly better than applying top-p to the denoiser itself.

The predictor-corrector comparison with filtered top-p sampling is shown in Figure 9. At 128 and 256 NFEs, the two approaches are broadly competitive, while at 512 and 1024 NFEs several predictor-corrector configurations move to a strictly better region of the frontier.

Performance comparison with AUDM Finally, Figure 10 reports the full frontier comparison for AUDM, resampled AUDM, UDM, and MDM. The two AUDM variants track each other closely and remain competitive throughout the NFE range. They outperform MDM and the denoiser parameterization variants, which is consistent with the main-text conclusion that the absorbing-uniform construction preserves strong sampling behavior while improving likelihood.

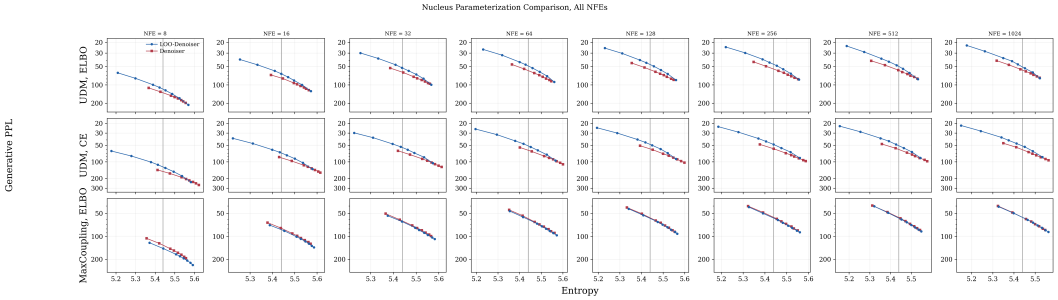


Figure 8: Top-p frontiers across all NFEs for the denoiser and leave-one-out parameterizations.

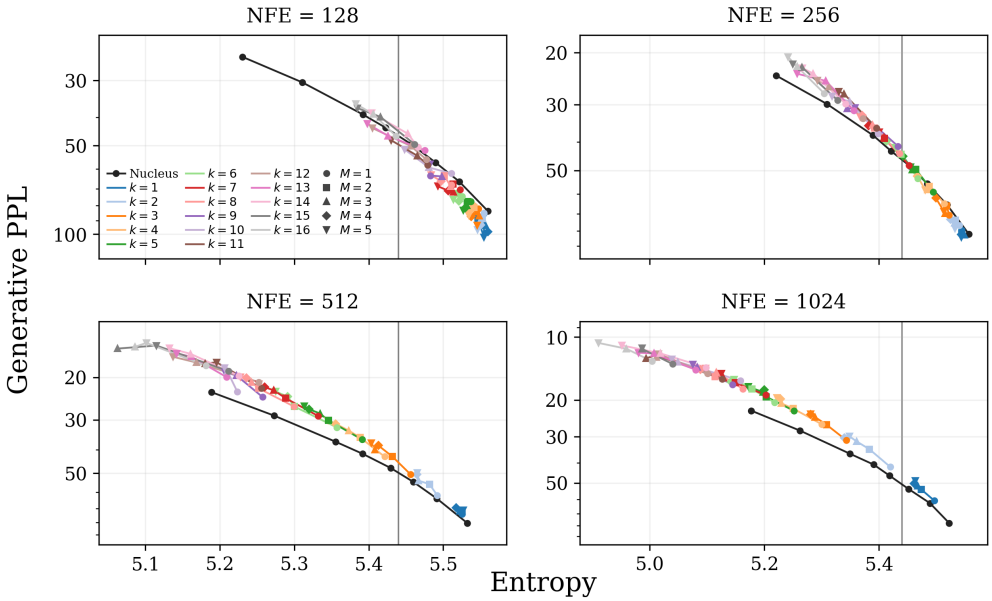


Figure 9: Comparison between predictor-corrector sampling and top-p sampling.

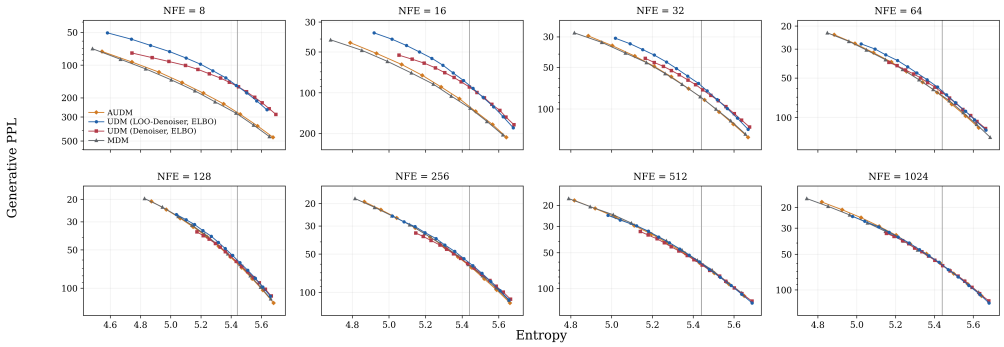


Figure 10: Full frontier comparison for AUDM, resampled AUDM, UDM, and MDM across all NFEs.

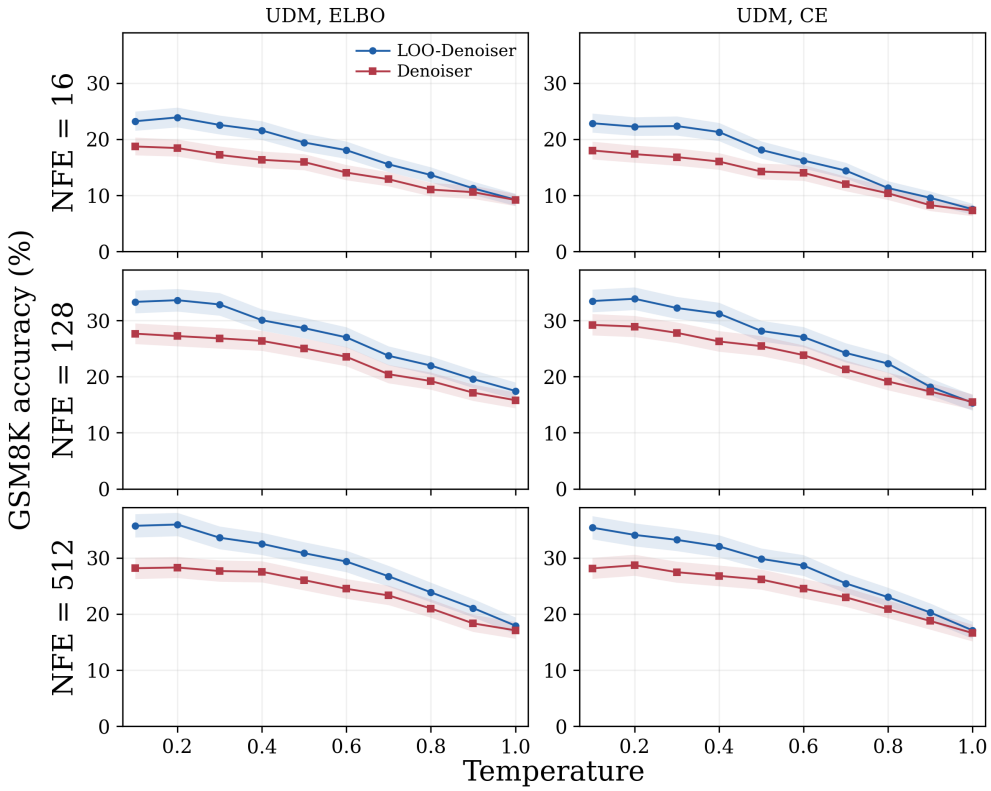


Figure 11: GSM8K accuracy as a function of sampling temperature for UDMs with leave-one-out and denoiser parameterizations. Columns correspond to ELBO and cross-entropy training; rows use 16, 128, and 512 NFEs.

J.3 Mathematical reasoning with TinyGSM

We complement the language-modeling and Sudoku evaluations with TinyGSM [6, 16] experiments. We compare UDMs with denoiser and leave-one-out parameterizations under cross-entropy (CE) and ELBO training.

Figure 11 isolates the effect of the UDM parameterization across temperatures from 0.1 to 1.0 and NFE budgets 16, 128, and 512. The leave-one-out parameterization improves accuracy under both objectives, with the largest gains at lower temperatures and convergence toward the denoiser results at temperature 1.0.

Figure 12 further compares applying temperature directly to a learned denoiser with first converting it to the leave-one-out representation. Conversion improves accuracy over much of the temperature range without additional training, although the gain is slightly unstable with CE training. Direct leave-one-out prediction therefore gives the strongest results across the different settings.

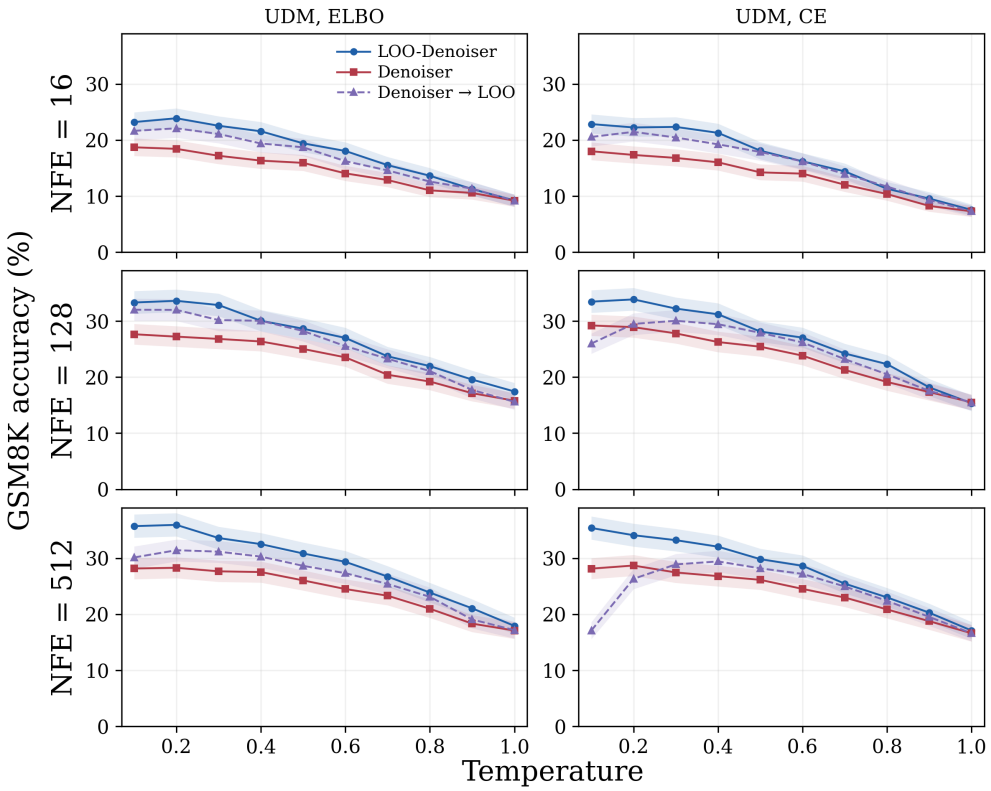


Figure 12: Effect of the representation to which temperature is applied in the TinyGSM experiment. The curves compare a trained leave-one-out model, a trained denoiser, and the same denoiser converted to the leave-one-out representation before applying temperature. Columns correspond to ELBO and cross-entropy training; rows use 16, 128, and 512 NFEs.

J.4 Generated samples

We show representative generated text from the OWT top-p sweep below. For each method, we select the $\mathrm{NFE} = 64$ sample whose empirical entropy is closest to 5.4, matching the data entropy estimate used in the main text. We also compare top-p sampling and predictor-corrector for the UDM ELBO / LOO-Denoiser model at $\mathrm{NFE} = 512$, selecting the closest-entropy sample in each sweep.

Top-p sampling examples ($\mathrm{NFE} = 64$).

UDM CE / LOO-Denoiser Entropy $= 5.418$, Gen-PPL $= 47.0$

because they still play on the same team, rather than each other. Having met the slower contenders, those who think they are the only the fastest have fast. Miguel from the start to start of the race looked good with a decent first finish in the field but struggles in the first 10, as he crosses #13, trailing at ten, and starts off at #13 limping for about 21 seconds.

By 1:20, he is out. He eventually breaks back, however, with a string of misses throughout the race before slamming home on laps 19 to knock out yet another winner, Miguel. In a video provided by Ahmed, Victor Sane of the five-man pack also stated that he has opted not to meet the #13 Chachea after running only 492 with him in lap 2.

UDM CE / Denoiser Entropy $= 5.401$, Gen-PPL $= 54.8$

the current installment of the series, Macrossi and showmaster would give us the two Init-List volumes, with the fundamental decision in place: collecting the six on the first issue and those other five in. Update: in the next time, he no longer acknowledges Nina and his planet, Nina, barely mentioning the source of "contract..." stories.

Those insights from the side that were all straightened for me. TWI's...I think And...yeah yeah probably that means a lot. Now, let's take for a look what we see the comics by Robert Sottarzio and Blake Nagle, and as well as comic book "from the crew" Miles Gerhart and Mike Campbell. Speaking of, those guys: Pat Bubuman and Fluffy Rainplie.

UDM ELBO / Denoiser Entropy $= 5.421$, Gen-PPL $= 54.8$

the current directions of what to expect out of the show. It would get much larger bigger. And then we would have to make sure of it. Things to expect would be forced Alan to begin with puberty. Update: at the next episode, he no longer seems to be finally starting with puberty, but at the next remaining episode of one. What follows is an intriguing glimpse into basic goals that they all had in the past.

The key are those who feel keen that watching these shows will probably reveal what is good for them, such as pimping for its sponsor or profiting from money, which is good for themselves. And than that though, might as well keep the people out of this.

UDM ELBO / LOO-Denoiser Entropy $= 5.400$, Gen-PPL $= 42.2$

the current state of the series as one of my favorites. It would be much easier to see this story evolve to get to 2021, and it shows Pats' POV on 9/11 and the other five timelines. Update: Alan tells it to himself he will "move to his planet with a black cube when the sequence is set to one." Ok, so happy to say this is something that makes all the jokes just disappear.

Yes, I feel bad that the relatively short storytelling has taken too far in the direction of the dust-covering action ending. I love all the tension and action of this show, and, for some reason, might as well watch the hell out of the ending because it's much deeper than that.

AUDM Entropy $= 5.439$, Gen-PPL $= 53.9$

the current condition of the series as to its ratings. Some critics also point to the legacy of Michaels in Michaels hands, and question his decision to leave the still on-air show running "the other way around." Sills, however, have no recent backlash against Michaels. Whether with the network.

Whether the failure of the show, the artistic intent of the show, or their reputation is something that they all share, we should not expect such a backlash. It bears noting that the failure of the Murgons franchise was one of the goals of Michaels earlier. Sports-fors often say "It's good for the teeth," but that's true when key figures in the sports enterprise are replaced. We see it this way: Viewers assume they will have to part ways.

Resampled AUDM Entropy $= 5.446$, Gen-PPL $= 53.6$

Simply put, the author notes and feedback tend to always be anonymous, not multiauthored. The higher your feedback, the less it will be "no reading more" and "how do I have more time," etc. Being anonymous does not play into the reader character and feedback, however. I've spent more time reading what reviews say than watching what I read.

I believe it's not bad, because the text fits into itself, so that's why I can't read all. Get the text card Every web site has a text card: the main page, mailing list, talkback list, the comments, thumbs/crumbs up, reviews etc. I generally do more with 50 cards, no one reads them when you want post reviews, but they're doing that most of the time, so I rely on it.

UDM ELBO / LOO-Denoiser comparison ($\mathrm{NFE} = 512$**).**

Top-p **sampling** Entropy $= 5.392$, Gen-PPL $= 41.5$

average 7.8 assists. That's where he went from getting high to major in the '90s. When it was season's end, though, Westbrook brought Seth Jones in. He didn't buy himself a chance to run with that. "I think we had a lot of matchups, and his matchup looked good," he says. "He also made a lot of adjustments in his playing style. I think Boris (Green) and Kemba Walker are getting better and they both make a lot of plays. But he's also made his game look good. We had guys like David West, but my team is losing guys. I'd be afraid to run into that."

Oh, it's midseason, but Westbrook will have to lead this new roster with Hassan Whiteside, whom the Thunder played with this past year as his Big Sam. Aldridge talks about a three-team rotation based on his Thibs. "They're from San Jose, they play at Portland, San Antonio, it's a rivalry there, I think," he says. "That's the biggest. Their team here is from California. They scored 110 field goals at the 12,000 throw line, they converted on 90. They're learning a new style. That's the big one. Part of it is just being able to incorporate the play and the tendencies and last year, the wins. You want to give the win a building block for a rotation."

Predictor-corrector Entropy $= 5.403$, Gen-PPL $= 37.5$

But the Windows operating system is quickly falling into relative obscurity and/or complete obscurity. In fact it has come to be called WinTV Meets Windows. I was shocked at that in my eyes. And the WinTV Framework? Jonathan (Jackson's nickname) is a developer himself. He created it. He wrote, all the architecture and components along with all of the servers and apps running it on Windows. Moreover,

Jonathan is a senior developer of the C++, so he is responsible for providing the Windows Store API for the software.

4: 1/1% Speedtest list is to help it's team experience the software as it is. Moreover, it ensures that it has everything it needs to build a new system, during development. Even if you want to try different components and projects, the whole code works. 3: Please write tests every time again. As with building the system, a simple, simple user interface, it needs very many tests. If you have more than 1% of such a test (CLS), then the speedtest list is also very useful. It shows how far the code needs to go if you include penetration testing (TPT). So in case you'd like to make a quick test and even then 1-2 in the long run, the new code is only possible.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the paper's three main contributions, namely the leave-one-out characterization of plug-in UDMs, the conversion formulas and predictor-corrector consequence, and the absorbing-state reformulation. These claims are then supported by the theoretical development in Sections 3.2 and 4 and appendix C and by the experiments in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The conclusion includes a dedicated limitations discussion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The main theoretical statements are given in Sections 3.2 and 4 and appendix C, with assumptions stated in the propositions and in the preliminaries, and the complete proofs are provided in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper describes the datasets, model families, losses, training targets, evaluation protocol, and sampling sweeps in Section 5, and the appendix gives the training and evaluation hyperparameters in detail in Appendix J.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper uses existing public datasets and describes the setup needed to reproduce the experiments, but this submission does not include an anonymized code release.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 gives the task setups and the model comparisons, and Appendix J specifies the optimizer, learning rate, batch size, schedule, precision, EMA, data splits, evaluation datasets, and sampling sweeps. These details are sufficient to understand how the reported results were obtained.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Results are reported from single training runs due to compute constraints (a single OWT run requires hundreds of H100-GPU hours).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This is indicated in Appendix J.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and the work consists of theoretical and empirical research on generative modeling using existing datasets, without human-subject experimentation or undisclosed high-risk interventions. The paper also states its scope, limitations, and broader impacts explicitly.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The conclusion discusses the broader impact of our work.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We do not release pre-trained models or scraped datasets.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All the datasets and publicly available models used are properly used and credited.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce or release a new dataset, codebase, checkpoint, or benchmark asset as part of the submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, annotation work commissioned by the authors, or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The work does not involve human subjects, so IRB approval or an equivalent review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not an important, original, or non-standard component of the proposed methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.