SIMPLEX RELAXATION FOR DISCRETE DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Discrete diffusion models for categorical generation are defined by a corruption kernel, which determines the intermediate state space and the associated reverse prediction problem. Within this family, uniform diffusion has been extensively developed, with recent work connecting its categorical corruption process to continuous representations and dynamics. Motivated by this view, we ask whether uniform discrete diffusion can be augmented with an explicit continuous state while leaving its categorical corruption process unchanged. We introduce Simplax, an exact Dirichlet–categorical augmentation that couples each corrupted categorical state with an auxiliary simplex-valued variable while preserving the uniform diffusion process as its categorical marginal. This augmentation yields a tractable Rao–Blackwellized reverse-bridge objective and a stochastic reverse sampler, while retaining the corrupted categorical state as the denoiser input. Empirically, Simplax improves the generative perplexity–entropy tradeoff on unconditional OpenWebText generation. On Sudoku, a model trained exclusively on 30-clue puzzles achieves the highest accuracy among the compared methods across all evaluated clue densities, including the minimum uniquely solvable 17-clue regime, and also achieves the highest validity in unconditional generation.

1 INTRODUCTION

Discrete diffusion models have emerged as a promising framework for generative modeling over categorical data, including text, biological sequences, and other symbolic domains (Hoogeboom et al., 2021; Austin et al., 2021; Campbell et al., 2022; Lou et al., 2024; Zhang et al., 2025). Compared with autoregressive generation, they offer a conceptually different route to parallel prediction by learning to invert a noising process on full categorical states. A central design choice in this framework is the corruption kernel. This choice determines the intermediate state space and the semantics of reverse updates, and shapes the form of the training objective used to approximate the reverse process (Campbell et al., 2022; Lou et al., 2024; Zhao et al., 2025).

Among these constructions, uniform diffusion has recently received increasing attention. Unlike masked diffusion, which corrupts tokens toward a distinguished mask state, uniform diffusion corrupts tokens within the original categorical alphabet without introducing an absorbing symbol (Schiff et al., 2025). Recent work has expanded uniform diffusion along several directions, including generalized corruption processes, repeated token revision, self-correction, few-step generation, improved parameterizations and sampling, and scaling to increasingly large language models (von Rütte et al., 2025; Gourevitch et al., 2026; von Rütte et al., 2026; Sahoo et al., 2026; Google DeepMind, 2026). At the same time, recent work has connected uniform diffusion to continuous representations and dynamics (Sahoo et al., 2025; Deschenaux et al., 2026). More broadly, a growing line of categorical and language modeling work has explored continuous, hybrid, or flow-based formulations over one-hot, simplex, embedding, and latent representations (Jo & Hwang, 2025; Lee et al., 2026; Pynadath et al., 2026; Roos et al., 2026; Chen et al., 2026; Deschenaux & Gulcehre, 2026).

At the same time, discrete diffusion retains a practical advantage over continuous-state formulations: categorical inputs can be embedded by direct token lookup, without the vocabulary-wide dense embedding products required by simplex-valued inputs. This raises a natural question:

Can uniform discrete diffusion incorporate an auxiliary continuous state while preserving both its categorical process and efficient token-based denoising?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

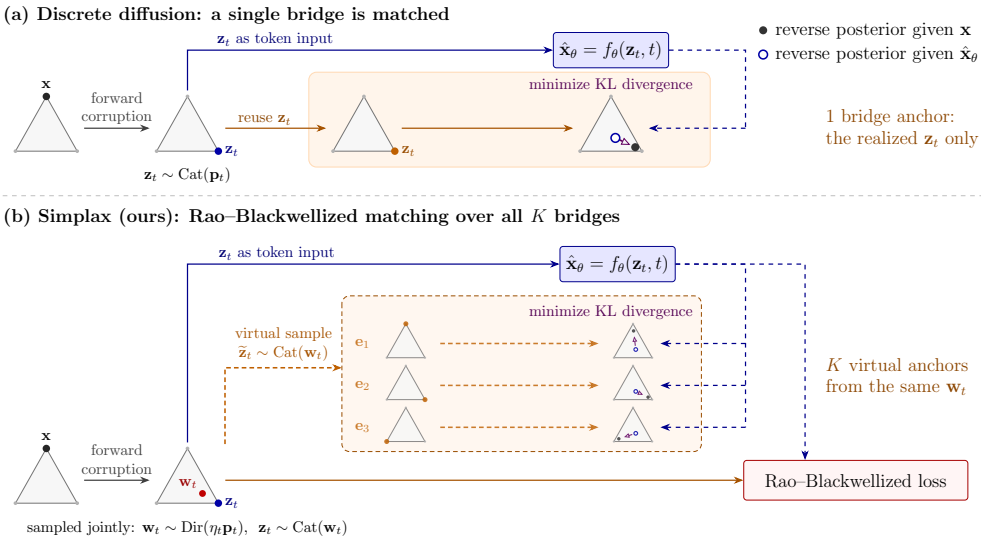

Figure 1: Reverse-bridge matching, illustrated for $K = 3$: one-hot states are vertices of Δ^{K-1} and $\mathbf{w}_t$ is an interior point. (a) The corrupted state $\mathbf{z}_t$ is both the denoiser input and the sole bridge anchor, so a single bridge is matched per sample. (b) Simplax samples $(\mathbf{w}_t, \mathbf{z}_t)$ jointly; $\mathbf{z}_t$ remains the denoiser input, while extra draws $\widetilde{\mathbf{z}}_t \sim \mathrm{Cat}(\mathbf{w}_t)$ anchor all K bridges, which are matched at once and marginalized exactly to give Eq. (16).

We seek an exact augmentation in which the continuous state participates in training and reverse transitions, while marginalizing it recovers the original uniform diffusion.

In this work, we answer this question with Simplax, an exact Dirichlet–categorical augmentation of uniform discrete diffusion. Each corrupted categorical state $\mathbf{z}_t$ is coupled to an auxiliary simplex-valued variable $\mathbf{w}_t$ through a shifted Dirichlet conditional. The hierarchy preserves the uniform diffusion process exactly as its categorical marginal and admits the decoder $q(\mathbf{z}_t \mid \mathbf{w}_t) = \mathrm{Cat}(\mathbf{z}_t; \mathbf{w}_t)$.

The augmented hierarchy can be used at both training and sampling time. For training, $\mathbf{w}_t$ defines a distribution over auxiliary categorical decodes $\widetilde{\mathbf{z}}_t$, each inducing a discrete reverse bridge to be matched (Fig. 1). Averaging over these auxiliary bridge anchors and marginalizing them analytically yields a tractable Rao–Blackwellized objective. For generation, the same simplex state defines a distribution over categorical alternatives that jointly contribute to the reverse update, rather than conditioning the transition only on the currently sampled token.

We evaluate Simplax on unconditional text generation with OpenWebText and constrained categorical generation with Sudoku. On OpenWebText, Simplax improves the generative perplexity–entropy tradeoff across a broad range of inference budgets. On Sudoku, models are trained only on 30-clue puzzles, and Simplax achieves strong in-distribution performance while generalizing to both easier and substantially harder clue densities, including 17-clue puzzles.

Our main contributions are as follows. First, we introduce an exact Dirichlet–categorical augmentation of uniform discrete diffusion that preserves the categorical process as a marginal while introducing an auxiliary simplex-valued state. Second, we derive exact reverse-bridge identities for the augmented process and use them to obtain a tractable Rao–Blackwellized categorical reverse-bridge objective together with an augmented reverse sampler. Third, we empirically demonstrate that the resulting framework improves generation across both open-ended text modeling and constrained categorical generation, including broad inference-budget regimes and distribution shifts in Sudoku.

2 Preliminaries

Notation. Let $\mathcal{V} = \{\mathbf{x} \in \{0,1\}^K : \sum_{i=1}^K x_i = 1\}$ denote the set of one-hot vectors over K categories, and let Δ^{K-1} denote the probability simplex over K categories. We represent scalar

discrete random variables taking K values as one-hot column vectors in $\mathcal{V}$. We write $\mathrm{Cat}(\cdot;\boldsymbol{\pi})$ for the categorical distribution with class probabilities $\boldsymbol{\pi} \in \Delta^{K-1}$. Results involving Dirichlet densities assume $\pi_k > 0$ for every category k; the uniform base distribution used in our experiments satisfies this condition. We write $\mathrm{Dir}(\cdot;\boldsymbol{\alpha})$ for the Dirichlet distribution with concentration vector $\boldsymbol{\alpha} \in \mathbb{R}^K_{>0}$. We use $\mathbf{1} \in \mathbb{R}^K$ for the all-ones vector, $\langle \mathbf{a}, \mathbf{b} \rangle$ for the inner product, $\mathbf{a} \odot \mathbf{b}$ for the Hadamard product, and $\mathbf{a} \oslash \mathbf{b}$ for elementwise division. For sequences of length L, we write $\mathbf{x}^{1:L} \in \mathcal{V}^L$.

2.1 Dirichlet distribution

The Dirichlet distribution is a distribution over the probability simplex (Kotz et al., 2000). Its mean is given by the normalized concentration vector, while the sum of the concentration parameters controls how concentrated the distribution is around its mean. In particular, for $\mathbf{p} \in \Delta^{K-1}$ and $\eta > 0$, $\mathrm{Dir}(\cdot;\eta\mathbf{p})$ denotes the Dirichlet distribution centered at $\mathbf{p}$, with η controlling its concentration.

2.2 Discrete diffusion

We consider a discrete diffusion process on $t \in [0,1]$ with prior $\boldsymbol{\pi} \in \Delta^{K-1}$ and noise schedule $\alpha_t \in [0,1]$, following the categorical diffusion framework of Hoogeboom et al. (2021); Austin et al. (2021) and its formulation with a general categorical stationary distribution (Zhao et al., 2025). Throughout, $0 \le s < t \le 1$, and simplex variables $\mathbf{w}_u$ are defined for $u > 0$.

The noisy categorical state at time t is distributed as

$$q(\mathbf{z}_t \mid \mathbf{x}) = \mathrm{Cat}(\mathbf{z}_t; \mathbf{p}_t(\mathbf{x})), \quad \mathbf{p}_t(\mathbf{x}) \coloneqq \alpha_t \mathbf{x} + (1-\alpha_t)\boldsymbol{\pi}. \tag{1}$$

We use a schedule with $\alpha_0 = 1$, $\alpha_1 = 0$, and $0 < \alpha_t < 1$ for $t \in (0,1)$, which ensures that $\mathbf{p}_t(\mathbf{x})$ is strictly positive for all $t > 0$.

The forward transition can be written as

$$q(\mathbf{z}_t \mid \mathbf{z}_s) = \mathrm{Cat}\left(\mathbf{z}_t; \alpha_{t|s}\mathbf{z}_s + (1-\alpha_{t|s})\boldsymbol{\pi}\right), \quad \alpha_{t|s} \coloneqq \frac{\alpha_t}{\alpha_s}. \tag{2}$$

The corresponding reverse posterior has the usual closed form

$$q(\mathbf{z}_s \mid \mathbf{z}_t, \mathbf{x}) = \mathrm{Cat}\left(\mathbf{z}_s; \mathbf{r}_{s|t}(\mathbf{x}, \mathbf{z}_t)\right), \tag{3}$$

where

$$\mathbf{r}_{s|t}(\mathbf{x}, \mathbf{z}_t) \coloneqq \frac{\left[\alpha_{t|s}\mathbf{z}_t + (1-\alpha_{t|s})\,\langle \mathbf{z}_t,\, \boldsymbol{\pi}\rangle\,\mathbf{1}\right] \odot \mathbf{p}_s(\mathbf{x})}{\langle \mathbf{z}_t,\, \mathbf{p}_t(\mathbf{x})\rangle}. \tag{4}$$

Standard discrete diffusion training minimizes the categorical reverse KL divergence

$$\mathcal{L}_{z_s|z_t}(\mathbf{z}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}) = D_{\mathrm{KL}}[q(\mathbf{z}_s \mid \mathbf{z}_t, \mathbf{x}) \,\|\, q(\mathbf{z}_s \mid \mathbf{z}_t, \hat{\mathbf{x}}_\theta)]. \tag{5}$$

where $\hat{\mathbf{x}}_\theta = f_\theta(\mathbf{z}_t, t) \in \Delta^{K-1}$ is the model prediction of the clean-token distribution. We use the shorthand $\mathbf{p}_t \coloneqq \mathbf{p}_t(\mathbf{x})$ and $\hat{\mathbf{p}}_t \coloneqq \mathbf{p}_t(\hat{\mathbf{x}}_\theta)$.

3 Method

We construct Simplax by augmenting uniform discrete diffusion with an auxiliary simplex-valued variable while leaving categorical corruption unchanged. This defines an exact Dirichlet–categorical hierarchy over each corrupted state, whose reverse bridge identities yield a tractable Rao–Blackwellized training objective and a sampler from the same bridge structure.

3.1 Simplex relaxation

Building upon Eq. (2), we consider the following joint factorization:

$$q(\mathbf{x}, \mathbf{z}_s, \mathbf{z}_t, \mathbf{w}_s, \mathbf{w}_t) = q(\mathbf{x})\, q(\mathbf{z}_s \mid \mathbf{x})\, q(\mathbf{w}_s \mid \mathbf{z}_s, \mathbf{x})\, q(\mathbf{z}_t \mid \mathbf{z}_s)\, q(\mathbf{w}_t \mid \mathbf{z}_t, \mathbf{x}). \tag{6}$$

Fig. 2 illustrates the graphical model implied by this factorization.

We introduce a simplex-valued variable $\mathbf{w}_t \in \Delta^{K-1}$ through

$$q(\mathbf{w}_t \mid \mathbf{z}_t, \mathbf{x}) = \mathrm{Dir}(\mathbf{w}_t; \eta_t \mathbf{p}_t + \mathbf{z}_t)\,, \tag{7}$$

where $\eta_t > 0$ is a concentration parameter. Its conditional mean interpolates between the diffusion-time categorical marginal $\mathbf{p}_t$ and the sampled discrete token $\mathbf{z}_t$, with the additive one-hot count anchoring the relaxed state to $\mathbf{z}_t$.

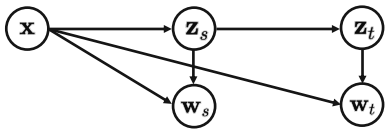

Figure 2: Graphical model implied by the factorization in Eq. (6).

This construction yields a Dirichlet–categorical hierarchy. The relaxed and discrete states satisfy

$$q(\mathbf{w}_t \mid \mathbf{x}) = \mathrm{Dir}(\mathbf{w}_t; \eta_t \mathbf{p}_t)\,, \quad q(\mathbf{z}_t \mid \mathbf{w}_t, \mathbf{x}) = q(\mathbf{z}_t \mid \mathbf{w}_t) = \mathrm{Cat}(\mathbf{z}_t; \mathbf{w}_t)\,. \tag{8}$$

Moreover, the reverse conditional posterior of $\mathbf{z}_s$ given $\mathbf{w}_t$ and $\mathbf{x}$ is categorical:

$$q(\mathbf{z}_s \mid \mathbf{w}_t, \mathbf{x}) = \mathrm{Cat}\left(\mathbf{z}_s; \boldsymbol{\rho}_{s|t}(\mathbf{x}, \mathbf{w}_t)\right), \tag{9}$$

where

$$\boldsymbol{\rho}_{s|t}(\mathbf{x}, \mathbf{w}_t) \coloneqq \mathbf{p}_s \odot \left[\alpha_{t|s}\left(\mathbf{w}_t \oslash \mathbf{p}_t\right) + (1 - \alpha_{t|s})\, \langle \mathbf{w}_t,\, \boldsymbol{\pi} \oslash \mathbf{p}_t \rangle\, \mathbf{1}\right]. \tag{10}$$

In turn, the reverse conditional posterior of $\mathbf{w}_s$ given $\mathbf{w}_t$ and $\mathbf{x}$ is a Dirichlet mixture:

$$q(\mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x}) = \sum_{k=1}^{K} \rho_{s|t,k}(\mathbf{x}, \mathbf{w}_t)\, \mathrm{Dir}(\mathbf{w}_s; \eta_s \mathbf{p}_s + \mathbf{e}_k)\,. \tag{11}$$

See Sec. D for the proof. These identities show that $\mathbf{w}_t$ is not an ad hoc surrogate. It is an exact auxiliary variable whose marginal remains native to the simplex and whose decoder back to $\mathbf{z}_t$ is simply categorical sampling from $\mathbf{w}_t$.

3.2 Training Objective

A natural starting point is to match the simplex bridge directly:

$$\mathcal{L}_{w_s|w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; s, t) = D_{\mathrm{KL}}[q(\mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x}) \,\|\, q(\mathbf{w}_s \mid \mathbf{w}_t, \hat{\mathbf{x}}_\theta)]\,. \tag{12}$$

However, this objective is generally intractable because $q(\mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x})$ is a Dirichlet mixture, as shown in Eq. (11). We therefore move back to the categorical space and match discrete reverse bridges induced by categorical decodes from $\mathbf{w}_t$.

To do so while retaining a categorical denoiser input, we first sample the augmented noisy state as

$$\mathbf{w}_t \sim q(\mathbf{w}_t \mid \mathbf{x}), \qquad \mathbf{z}_t \sim q(\mathbf{z}_t \mid \mathbf{w}_t) = \mathrm{Cat}(\mathbf{z}_t; \mathbf{w}_t), \tag{13}$$

and predict the clean-token distribution $\hat{\mathbf{x}}_\theta = f_\theta(\mathbf{z}_t, t)$.

We then introduce an auxiliary categorical decode $\widetilde{\mathbf{z}}_t$ satisfying

$$\widetilde{\mathbf{z}}_t \sim q(\widetilde{\mathbf{z}}_t \mid \mathbf{w}_t) = \mathrm{Cat}(\widetilde{\mathbf{z}}_t; \mathbf{w}_t), \tag{14}$$

conditionally independently of the network input $\mathbf{z}_t$ given $\mathbf{w}_t$. We optimize

$$\bar{\mathcal{L}}_{z_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; s, t) \coloneqq \mathbb{E}_{q(\widetilde{\mathbf{z}}_t|\mathbf{w}_t)}\Big[D_{\mathrm{KL}}[q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{x}) \,\|\, q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t, \hat{\mathbf{x}}_\theta)]\Big]. \tag{15}$$

This objective averages the standard discrete reverse KL divergence over an auxiliary decoder sample from $\mathbf{w}_t$, while retaining $\mathbf{z}_t$ as the denoiser input (Fig. 1).

Rao–Blackwellized form. The expectation over the auxiliary decoder sample $\widetilde{\mathbf{z}}_t$ in Eq. (15) can be marginalized exactly:

$$\bar{\mathcal{L}}_{z_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; s, t) = \langle \mathbf{w}_t,\, \log \hat{\mathbf{p}}_t - \log \mathbf{p}_t \rangle + \left\langle \boldsymbol{\rho}_{s|t}(\mathbf{x}, \mathbf{w}_t),\, \log \mathbf{p}_s - \log \hat{\mathbf{p}}_s \right\rangle. \tag{16}$$

This is the Rao–Blackwellized form of the Monte Carlo estimator that samples $\widetilde{\mathbf{z}}_t \sim q(\widetilde{\mathbf{z}}_t \mid \mathbf{w}_t)$ before evaluating the discrete reverse KL. The independently sampled $\mathbf{z}_t$ remains the denoiser input and is not marginalized. The proof is given in Sec. E. The closed form eliminates sampling noise from $\widetilde{\mathbf{z}}_t$ and depends on $\mathbf{w}_t$ only through $\langle \mathbf{w}_t, \log \hat{\mathbf{p}}_t \rangle$ and $\boldsymbol{\rho}_{s|t}(\mathbf{x}, \mathbf{w}_t)$. Thus, a single prediction from the realized input $\mathbf{z}_t$ suffices to evaluate the bridge contributions of all auxiliary decodes.

Continuous-time limit. The closed form in Eq. (16) yields a non-degenerate infinitesimal limit. Let $s = t - \Delta$ with $\Delta \downarrow 0$. Up to θ-independent additive terms, the relaxed discrete bridge objective satisfies

$$\bar{\mathcal{L}}_{z_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; t-\Delta, t) = \Delta\, \ell_{\mathrm{ct}}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}, t) + o(\Delta), \tag{17}$$

where $\lambda(t) := -\frac{\mathrm{d}}{\mathrm{d}t} \log \alpha(t)$ and

$$\ell_{\mathrm{ct}}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}, t) = \lambda(t) \Big[\langle \mathbf{w}_t,\, \boldsymbol{\pi} \oslash \hat{\mathbf{p}}_t \rangle - \langle \mathbf{w}_t,\, \boldsymbol{\pi} \oslash \mathbf{p}_t \rangle\ \langle \mathbf{p}_t,\, \log \hat{\mathbf{p}}_t \rangle + \langle \boldsymbol{\pi} \odot (\mathbf{w}_t \oslash \mathbf{p}_t),\, \log \hat{\mathbf{p}}_t \rangle \Big]. \tag{18}$$

The corresponding continuous-time objective is

$$\mathcal{L}_{\mathrm{ct}} = \int_0^1 \mathbb{E}_{q(\mathbf{x})} \Big[\mathbb{E}_{q(\mathbf{w}_t|\mathbf{x})} \Big[\mathbb{E}_{q(\mathbf{z}_t|\mathbf{w}_t)} \Big[\ell_{\mathrm{ct}}\big(\mathbf{w}_t, f_\theta(\mathbf{z}_t, t), \mathbf{x}, t\big) \Big] \Big] \Big]\, \mathrm{d}t. \tag{19}$$

The proof is deferred to Sec. F.1. Eq. (19) is therefore the continuous-time counterpart of Eq. (15).

The structure of Eqs. (15) and (19) is closely related to UDLM (Schiff et al., 2025). UDLM derives a continuous-time reverse-KL objective directly from the discrete corrupted state Eq. (5). Our construction introduces the exact auxiliary variable $\mathbf{w}_t$, averages the same reverse-KL bridge over an auxiliary draw $\widetilde{\mathbf{z}}_t \sim q(\widetilde{\mathbf{z}}_t \mid \mathbf{w}_t) = \mathrm{Cat}(\mathbf{w}_t)$, and then takes the infinitesimal limit. In this sense, Eq. (19) can be understood as a simplex-relaxed continuous-time analogue of the UDLM objective.

Alternative bridge objectives. One might instead directly match the marginal categorical bridge $q(\mathbf{z}_s \mid \mathbf{w}_t, \mathbf{x})$ or the full augmented bridge $q(\mathbf{z}_s, \mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x})$. More generally, different choices of the conditioning variables and the matched bridge lead to distinct objectives. For $s = t - \Delta$, direct marginal matching is $O(\Delta^2)$ and therefore yields a degenerate continuous-time limit, whereas full augmented matching contains an $O(1)$ Dirichlet-matching term and diverges under continuous-time scaling. In contrast, our objective is $O(\Delta)$ and admits a finite, nonzero continuous-time limit. We summarize these objectives, their inequality relations, and their continuous-time limits in Sec. G.

3.3 Sampling

At inference time, we run the reverse process on a grid $1 = t_N > t_{N-1} > \cdots > t_0 = 0$. For a denoiser conditioned on $\mathbf{z}_t$, the Dirichlet–categorical hierarchy defines a reverse process over the augmented state $(\mathbf{z}_t, \mathbf{w}_t)$. Since $q(\mathbf{w}_1) = \mathrm{Dir}(\mathbf{w}_1; \eta_1 \boldsymbol{\pi})$ and $q(\mathbf{z}_1 \mid \mathbf{w}_1) = \mathrm{Cat}(\mathbf{z}_1; \mathbf{w}_1)$, generation starts by sampling $\mathbf{w}_{t_N} \sim \mathrm{Dir}(\eta_{t_N} \boldsymbol{\pi})$ followed by $\mathbf{z}_{t_N} \sim \mathrm{Cat}(\mathbf{w}_{t_N})$. Given adjacent times $s = t_{n-1} < t = t_n$ and the current state $(\mathbf{z}_t, \mathbf{w}_t)$, the denoiser predicts $\hat{\mathbf{x}}_\theta = f_\theta(\mathbf{z}_t, t)$, inducing the bridge marginals $\hat{\mathbf{p}}_s$ and $\hat{\mathbf{p}}_t$ and the marginal reverse categorical distribution $\boldsymbol{\rho}_{s|t}(\hat{\mathbf{x}}_\theta, \mathbf{w}_t)$.

Reverse sampler and variants. We sample from this marginal reverse distribution and then update the auxiliary simplex state:

$$\mathbf{z}_s \sim \mathrm{Cat}\big(\boldsymbol{\rho}_{s|t}(\hat{\mathbf{x}}_\theta, \mathbf{w}_t)\big), \qquad \mathbf{w}_s \sim \mathrm{Dir}(\eta_s \hat{\mathbf{p}}_s + \mathbf{z}_s). \tag{20}$$

Here, $\mathbf{z}_s$ serves as the next denoiser input, while $\mathbf{w}_s$ carries the simplex-valued bridge state.

The update contains two stochastic transitions, for $\mathbf{z}_s$ and $\mathbf{w}_s$, yielding four basic variants depending on which transitions are determinized. The *stochastic* sampler draws both as in Eq. (20). The $\mathbf{w}$-*deterministic* variant samples $\mathbf{z}_s$ but replaces the Dirichlet draw for $\mathbf{w}_s$ by its conditional mean, whereas the $\mathbf{z}$-*deterministic* variant replaces the categorical draw for $\mathbf{z}_s$ by its mode while retaining the Dirichlet draw for $\mathbf{w}_s$. The fully *deterministic* variant applies both replacements. Explicit update rules and derivations are given in Sec. H, and the variants are compared in Tab. 4.

4 Experiments

We evaluate Simplax on unconditional OpenWebText generation (Gokaslan & Cohen, 2019) and constrained categorical Sudoku generation (Lee et al., 2026; Deschenaux & Gulcehre, 2026).

4.1 Experimental setup

For the main Simplax results, we use $\mathbf{z}_t$ as the denoiser input, constant concentration $\eta_t \equiv 0.01$, and the $\mathbf{w}$-deterministic reverse sampler. These choices are examined in Sec. 4.4.

Table 1: OpenWebText unconditional generation at selected NFE values. Gen. ENT is generative unigram entropy and should be compared with the data entropy 5.44. Gen. PPL is evaluated by the indicated external model. Time is median seconds per batch of 16 sequences of length 1024. The best and second-best values among diffusion models are shown in bold and underlined, respectively.

	NFE = 16				NFE = 128				NFE = 1024			
Method	ENT	GPT-2 L	Llama-2	Time (s)	ENT	GPT-2 L	Llama-2	Time (s)	ENT	GPT-2 L	Llama-2	Time (s)
AR	-	-	-	-	-	-	-	-	5.43	22.8	16.0	8.46
CANDI (Pynadath et al., 2026)	5.43	97.2	56.0	0.648	5.45	67.2	36.8	4.97	5.46	74.3	39.2	39.7
UDLM (Schiff et al., 2025)	5.53	186.0	79.0	0.677	5.49	62.9	34.6	5.53	5.46	59.2	32.2	44.5
MDLM (Sahoo et al., 2024)	5.47	117.9	63.8	**0.548**	5.46	65.4	37.3	**4.38**	5.40	55.1	33.9	**35.4**
Duo (Sahoo et al., 2025)	5.45	164.0	86.0	0.677	5.45	94.4	52.6	5.53	5.43	88.2	49.5	44.5
FLM (Lee et al., 2026)	5.58	259.5	139.3	0.896	5.42	112.0	66.5	7.12	5.45	125.6	74.4	57.0
LangFlow (Chen et al., 2026)	5.42	115.1	59.9	1.15	5.42	60.2	**30.0**	9.38	5.41	68.3	28.2	75.3
S-FLM (Deschenaux & Gulcehre, 2026)	5.45	124.6	62.9	1.18	5.43	103.4	52.4	9.98	5.46	108.5	53.7	80.5
Simplax	5.46	98.7	54.2	0.814	5.46	63.4	35.0	6.09	5.46	52.6	29.8	48.4
Simplax (UDLM init)	5.42	**81.5**	**45.2**	0.814	5.45	**56.5**	31.1	6.09	5.42	**42.8**	**24.4**	48.4

OpenWebText. We tokenize OpenWebText with the GPT-2 BPE tokenizer (Radford et al., 2019), giving $|\mathcal{V}| = 50{,}257$, and use sequence length $L = 1024$. All methods use the same 179M-parameter diffusion transformer backbone of Sahoo et al. (2024) and are trained for a total budget of 1M optimization steps. Preprocessing, model architecture, optimization, checkpoint selection, sampling, and computational-resource details are provided in Sec. A.

Sudoku. We build on the Sudoku benchmark of Deschenaux & Gulcehre (2026) and evaluate cross-clue generalization. All models are trained only on puzzles with 30 clues and are evaluated using the same checkpoint on puzzles with 40, 35, 30, 25, 20, and 17 clues. Thus, the evaluation measures transfer to both more heavily and more sparsely conditioned inputs than those seen during training. The 17-clue setting is particularly challenging, as 17 is the minimum number of clues for which a standard 9×9 Sudoku puzzle can admit a unique solution (McGuire et al., 2014; Lin et al., 2013). We additionally evaluate unconditional Sudoku generation using an all-blank puzzle prefix. Dataset construction, sequence representation, model architectures, optimization, and computational-resource details are provided in Sec. B.

Metrics. For OpenWebText, we draw 1024 sequences and report generative unigram entropy and generative perplexity under GPT-2 Large and Llama-2 7B (Radford et al., 2019; Touvron et al., 2023); GPT-2 XL results are reported in Sec. A.6. The OpenWebText unigram entropy is 5.44 nats. For all OpenWebText evaluations, we sweep 15 temperatures from 0.84 to 1.12 in increments of 0.02 for each configuration and NFE budget, and select the temperature whose generated entropy is closest to 5.44 nats. We also report sampling time at selected NFE budgets; runtime and peak GPU memory are in Sec. A.4. For conditional Sudoku, we report solving accuracy. For unconditional Sudoku, we report validity, the fraction of generated boards satisfying all Sudoku constraints.

4.2 Unconditional generation on OpenWebText

We compare Simplax with AR, MDLM (Sahoo et al., 2024), UDLM (Schiff et al., 2025), Duo (Sahoo et al., 2025), CANDI (Pynadath et al., 2026), FLM (Lee et al., 2026), LangFlow (Chen et al., 2026), and S-FLM (Deschenaux & Gulcehre, 2026). We also evaluate a staged variant that trains UDLM for 800k steps and Simplax for 200k steps under the same 1M-step budget; other allocations are reported in Sec. C. We compare with ELF (Hu et al., 2026) separately in Sec. A.5, as its sweep does not reach the OpenWebText data entropy required for the matched-entropy comparison.

At the entropy-matched operating points, Simplax trained from scratch remains competitive across inference budgets, while UDLM initialization further reduces generative perplexity (Tab. 1). The UDLM-initialized variant gives the lowest PPL among diffusion methods at low and high NFE under all evaluators and remains close to LangFlow at the intermediate budget. The full temperature sweeps in Fig. 4 show the same pattern beyond these operating points, with Simplax on or near the lower PPL–entropy envelope across temperatures and NFE values. The benefit of UDLM initialization further suggests that staged allocation of a fixed training budget can be beneficial: categorical diffusion pretraining followed by Simplax training performs better than either objective used throughout. This is similar in spirit to the staged training strategy of Duo.

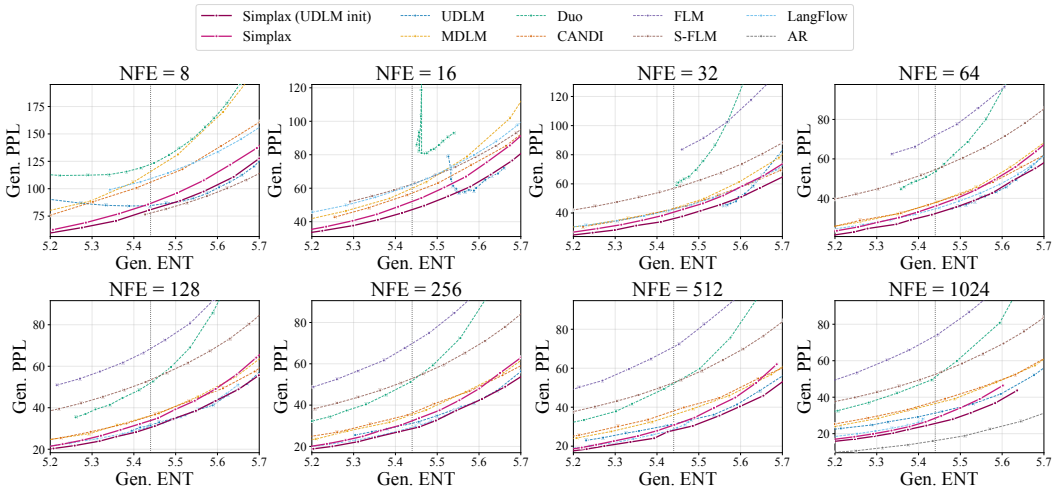

Figure 4: Llama-2 7B generative frontiers on OpenWebText for NFE $\in \{8, 16, 32, 64, 128, 256, 512, 1024\}$. Each panel shows the temperature-swept Gen. PPL–Gen. ENT tradeoff. Lower Gen. PPL is better, and the reference data entropy is 5.44.

Table 2: Conditional Sudoku solving accuracy and unconditional Sudoku validity in percent. All models are trained with 30 clues. We report the mean ± sample standard deviation over five sampling seeds using a fixed checkpoint and the same 2,000 examples per seed.

	Conditional accuracy (%)						Uncond. validity (%)
Method	40 clues	35 clues	30 clues	25 clues	20 clues	17 clues	0 clues
AR	16.44 ± 0.38	4.08 ± 0.39	0.67 ± 0.04	0.08 ± 0.03	0.00 ± 0.00	0.00 ± 0.00	8.48 ± 0.67
UDLM	97.02 ± 0.27	84.24 ± 0.36	48.69 ± 0.64	16.78 ± 0.40	5.44 ± 0.53	0.58 ± 0.17	80.71 ± 1.07
FLM	96.86 ± 0.27	83.96 ± 0.12	49.27 ± 0.74	15.06 ± 0.96	3.74 ± 0.17	0.34 ± 0.21	33.48 ± 0.44
MDLM	98.10 ± 0.27	85.76 ± 0.84	47.12 ± 0.68	11.87 ± 0.53	2.49 ± 0.72	0.23 ± 0.07	22.02 ± 0.75
S-FLM	77.28 ± 0.62	42.87 ± 0.37	9.96 ± 0.77	0.96 ± 0.16	0.04 ± 0.02	0.00 ± 0.00	1.52 ± 0.20
S-FLM (trunc.)	94.99 ± 0.32	79.00 ± 0.32	41.34 ± 0.38	11.53 ± 0.26	3.08 ± 0.45	0.29 ± 0.11	4.58 ± 0.84
Simplax	$\mathbf{98.37 \pm 0.26}$	$\mathbf{90.53 \pm 0.45}$	$\mathbf{61.70 \pm 1.08}$	$\mathbf{24.69 \pm 1.04}$	$\mathbf{8.38 \pm 0.87}$	$\mathbf{0.82 \pm 0.13}$	$\mathbf{95.30 \pm 0.37}$

Simplax also has moderate sampling cost (Tab. 1). It is faster than FLM, LangFlow, and S-FLM, whose sampler updates require dense matrix products involving the vocabulary embedding matrix. It is slower than MDLM, UDLM, Duo, and CANDI, consistent with the cost of maintaining dense $\boldsymbol{\rho}$ during Simplax sampling. Thus, Simplax lies between categorical diffusion and continuous-state or flow-based methods in sampling runtime while providing the improved PPL–entropy tradeoff above.

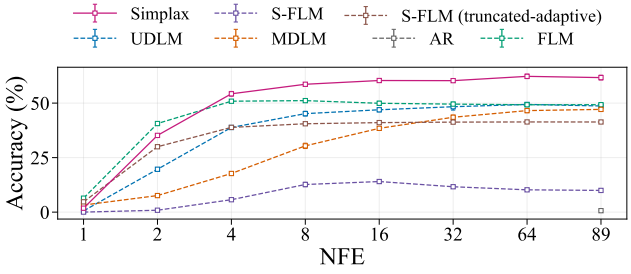

Figure 3: Few-step generation with 30 clues for both training and inference.

Additional analyses are provided in the appendix. Bootstrap uncertainty estimates and matched-entropy GPT-2 XL results are reported in Sec. A.6, with GPT-2 Large and GPT-2 XL PPL–entropy frontiers in Figs. 6 and 7. Detailed runtime and memory-usage procedures and results appear in Sec. A.4 and Tab. 8. Representative generations at NFE = 16, 128, and 1024 appear in Sec. A.7.

4.3 Constrained Categorical Generation on Sudoku

We compare Simplax with AR, UDLM (Schiff et al., 2025), FLM (Lee et al., 2026), MDLM (Sahoo et al., 2024), S-FLM (Deschenaux & Gulcehre, 2026), and truncated-adaptive S-FLM.

Table 3: **Input.** Simplax 50k, $\eta = 0.01$, with stochastic sampling; each input is used in training and inference.

	NFE = 16	NFE = 128	NFE = 1024
Input	PPL / ENT	PPL / ENT	PPL / ENT
$\mathbf{w}_t$	113.1/5.43	87.6/5.46	94.3/5.46
$\mathbf{z}_t$	105.1/5.43	73.6/5.46	56.2/5.43

Table 4: **Reverse-sampler quality and runtime.** Simplax 50k, $\eta = 0.01$, and $\mathbf{z}_t$ input. Runtime is measured at NFE = 1024.

Sampler	NFE = 16	NFE = 128	NFE = 1024	Time (s)
	PPL/ENT	PPL/ENT	PPL/ENT	
$\mathbf{w}$-det.	106.0/5.44	75.1/5.45	60.7/5.46	48.4
$\mathbf{z}$-det.	20.8/1.07†	445.9/2.92†	694.8/4.79†	85.5
Stochastic	105.1/5.43	73.6/5.46	56.2/5.43	103.3
Deterministic	14.7/0.91†	14.8/0.91†	14.8/0.91†	38.8

Table 5: **Concentration.** Simplax 50k, stochastic sampling, $\mathbf{z}_t$ input.

	NFE = 16	NFE = 128	NFE = 1024
η	PPL / ENT	PPL / ENT	PPL / ENT
0	103.9/5.44	81.8/5.47	78.7/5.45
10^{-3}	103.6/5.43	80.2/5.45	75.4/5.43
10^{-2}	105.1/5.43	73.6/5.46	56.2/5.43
10^{-1}	132.4/5.43	69.1/5.45	32.3/5.15†

Table 6: **Training–sampling decomposition.** UDLM vs Simplax, $\eta = 0.01$, $\mathbf{z}_t$ input, $\mathbf{w}$-det. sampler.

Training	Sampler	NFE = 16	NFE = 128	NFE = 1024
		PPL / ENT	PPL / ENT	PPL / ENT
UDLM	UDLM	186.0/5.53†	62.9/5.49	59.2/5.46
UDLM	Simplax	82.8/5.42	80.4/5.46	217.8/5.44
Simplax	UDLM	184.3/5.48	54.8/5.43	55.4/5.43
Simplax	Simplax	81.5/5.42	56.5/5.45	42.8/5.42

† marks sweeps that do not reach the data entropy 5.44 ± 0.05; their PPL is not directly comparable at matched entropy. Shaded rows indicate the design choices used in the main experiments.

Simplax achieves the highest conditional accuracy across all clue densities and the highest unconditional validity (Tab. 2). The differences are small when many clues are observed, but become larger as the conditioning information is reduced. The same pattern extends to unconditional generation, where Simplax reaches 95.30% validity compared with 80.71% for the strongest baseline. Together, these results suggest that the improvement extends beyond exploiting the observed clues to modeling the global constraints of valid Sudoku grids.

The few-step results in Fig. 3 show that this advantage also appears under a limited sampling budget. With four or more function evaluations, Simplax achieves the highest accuracy among all evaluated methods and maintains this lead as the sampling budget increases. The separation therefore emerges after only a few refinement steps rather than only in the large-NFE regime, suggesting that Simplax makes more effective use of each function evaluation during generation.

Representative Sudoku instances in Fig. 11 provide visual examples of conditional generation results. Additional few-step results under different inference-time clue counts are provided in Sec. B.4.

4.4 Contribution Analysis of Training and Sampling

Beyond the generation results, we examine which Simplax components drive the improvement and which design choices matter. We study the denoiser input, reverse sampler, bridge concentration, and training–sampling interaction. We report GPT-2 Large Gen. PPL / Gen. ENT at the temperature closest to the data entropy of 5.44 on OpenWebText. Additional details are provided in Sec. C.

Training input. We compare Simplax trained and evaluated with $\mathbf{z}_t$ input against Simplax trained and evaluated with $\mathbf{w}_t$ input. The $\mathbf{z}_t$ input yields a consistently better frontier, with the largest margin at NFE $= 1024$ (Tab. 3). We therefore adopt $\mathbf{z}_t$, which also avoids the vocabulary-sized dense multiplication $\mathbf{w}_t^{\mathsf{T}} E$ required by $\mathbf{w}_t$ input.

Reverse sampler. We compare the *stochastic*, $\mathbf{w}$-*deterministic*, $\mathbf{z}$-*deterministic*, and *deterministic* samplers introduced in Secs. 3.3 and H in terms of entropy-matched PPL and runtime. Runtime evaluation details are provided in Secs. A.4 and C. The *stochastic* and $\mathbf{w}$-*deterministic* frontiers nearly overlap across NFEs, while the latter reduces generation time (Tab. 4). The *deterministic* and $\mathbf{z}$-*deterministic* samplers fail to reach the data entropy. We therefore use $\mathbf{w}$-*deterministic* for its similar quality and lower sampling cost.

Bridge concentration. We compare $\eta \in \{0, 0.001, 0.01, 0.1\}$, with $\eta = 0$ the categorical limit. At $\eta = 0.01$, low-NFE quality is preserved while the frontier improves at higher NFEs, while $\eta = 0.001$

remains similar (Tab. 5). Larger concentration helps at intermediate NFE but is less reliable at high NFE and may fail to reach the target entropy. We therefore use $\eta = 0.01$ across inference budgets.

Training and sampling. Simplax preserves the UDLM marginal over $\mathbf{z}_t$ while augmenting it with auxiliary $\mathbf{w}_t$, allowing either sampler to be used with either checkpoint. Here, UDLM uses the 1M-step checkpoint, while Simplax uses UDLM-initialized variant from Sec. 4.2. We therefore evaluate all four training–sampling combinations (Tab. 6). At low NFE, replacing the UDLM sampler with the Simplax sampler accounts for most of the improvement, even with the UDLM checkpoint. At high NFE, the Simplax sampler instead benefits strongly from Simplax training, while the UDLM sampler largely removes this advantage. Thus, the low-NFE gain is mainly associated with the Simplax reverse process, whereas the high-NFE gain reflects training–sampling interaction.

5 Related Work

Discrete diffusion for categorical data. Discrete diffusion for categorical data was developed through early multinomial formulations and later unified and substantially generalized by D3PM, which introduced structured transition kernels such as uniform and absorbing corruptions and established the standard variational training recipe (Hoogeboom et al., 2021; Austin et al., 2021). This framework was extended to continuous-time formulations and alternative reverse objectives (Campbell et al., 2022; Lou et al., 2024; Zhang et al., 2025), and has since supported a broad line of language-modeling work covering masked, absorbing, and uniform-state diffusion (Sahoo et al., 2024; Shi et al., 2024; Ou et al., 2025; Zheng et al., 2025; Sahoo et al., 2025; Deschenaux et al., 2026; Li et al., 2022). Our method stays within this discrete-diffusion lineage: we keep the original categorical forward process and reverse posterior, and do not replace the primary generative state.

Auxiliary-variable and hybrid formulations. A parallel line of work enriches discrete diffusion through auxiliary variables or structured reverse distributions. Di4C (Hayakawa et al., 2025) uses mixtures of product models to capture dimensional correlations, VADD (Xie et al., 2026) introduces a Gaussian latent into masked denoising, and CoDD couples factorized outputs with probabilistic circuits (Li et al., 2026). Continuous and hybrid constructions include Gaussian-relaxed views in Duo and Duo++ (Sahoo et al., 2025; Deschenaux et al., 2026), Euclidean denoising over one-hot states in FLM (Lee et al., 2026), and discrete–continuous diffusion in CADD and CANDI (Zheng et al., 2026; Pynadath et al., 2026). Simplax instead preserves categorical diffusion and keeps $\mathbf{z}_t$ as the denoiser input, using the simplex variable only to define the reverse bridge.

Diffusion and flow on the simplex. Our method is related to a broader class of diffusion and flow constructions on the simplex for generative modeling, including softmax-transformed continuous diffusion (Floto et al., 2023), categorical SDEs with Cox–Ingersoll–Ross dynamics (Richemond et al., 2023), Dirichlet-based score models such as DDSM (Avdeyev et al., 2023), Dirichlet Flow Matching (Stark et al., 2024), and recent unifying views of discrete, Gaussian, and simplicial diffusion (Chandra et al., 2026). These works are geometrically close to ours through their use or analysis of simplex-valued states, but differ in role: Simplax uses the simplex variable as an exact auxiliary bridge attached to a standard discrete diffusion process, rather than as the primary generative state.

6 Conclusion and Limitations

We introduced Simplax, an exact Dirichlet–categorical augmentation of uniform discrete diffusion. Simplax preserves the categorical forward process while introducing an auxiliary simplex state to derive a Rao–Blackwellized reverse-bridge objective and stochastic ancestral sampler. The denoiser still operates on discrete categorical states, avoiding the dense vocabulary-embedding product associated with continuous simplex inputs while retaining standard embedding lookups. It improves the perplexity–entropy tradeoff on OpenWebText and achieves the highest Sudoku performance among compared methods across different clues from 40 to 17, as well as in unconditional generation.

Limitation. The present formulation is specialized to uniform categorical corruption and introduces an auxiliary simplex-valued state whose computational overhead relative to standard discrete diffusion has not been fully characterized. Moreover, the concentration schedule remains a design choice rather than being determined by theory. Extending the construction to broader categorical corruption kernels and developing more efficient reverse solvers remain important future directions.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the development and refinement of conceptual and mathematical formulations, experimental design, implementation and debugging, and the interpretation of experimental results. We also used generative AI tools for brainstorming, literature search and summarization, and drafting and editing portions of the manuscript. Generative AI tools were also used to assist with mathematical derivations and proofs, including suggesting proof strategies and checking intermediate derivation steps.

All AI-assisted mathematical arguments and claims were independently checked by the authors. References suggested by AI tools were verified against the original sources, and AI-assisted code was reviewed and tested by the authors. All reported numerical results were obtained from actual experimental runs and checked against the underlying outputs. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects or the collection of personal data, and we are not aware of any ethical concerns specific to this work.

REPRODUCIBILITY STATEMENT

We provide complete derivations and additional theoretical details in the appendix. The experimental setup, including datasets and preprocessing, model architectures, training hyperparameters, sampling configurations, evaluation protocols, and computational resources, is documented in the experimental sections and appendix. An anonymized implementation is provided as part of the supplementary material.

REFERENCES

Ali Alp. Sudoku puzzle generator. https://github.com/alicommit-malp/sudoku, 2024.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, 2021.

Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. In *International Conference on Machine Learning*, 2023.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems*, 2022.

Nuria Alina Chandra, Yucen Lily Li, Alan Nawzad Amin, Alex Ali, Joshua Rollins, Sebastian W. Ober, Aniruddh Raghu, and Andrew Gordon Wilson. A unification of discrete, gaussian, and simplicial diffusion. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *The Eleventh International Conference on Learning Representations*, 2023.

Yuxin Chen, Chumeng Liang, Hangke Sui, Ruihan Guo, Chaoran Cheng, Jiaxuan You, and Ge Liu. Langflow: Continuous diffusion rivals discrete in language modeling, 2026. URL https://arxiv.org/abs/2604.11748.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2006.

Justin Deschenaux and Caglar Gulcehre. Language modeling with hyperspherical flows, 2026. URL https://arxiv.org/abs/2605.11125.

Justin Deschenaux, Caglar Gulcehre, and Subham Sekhar Sahoo. The diffusion duality, chapter II: Ψ-samplers and efficient curriculum. In *The Fourteenth International Conference on Learning Representations*, 2026.

Griffin Floto, Thorsteinn Jonsson, Mihai Nica, Scott Sanner, and Eric Zhengyu Zhu. Diffusion on the probability simplex. In *ICML 2023 Workshop: Sampling and Optimization in Discrete Space*, 2023.

Aaron Gokaslan and Vanya Cohen. OpenWebText corpus, 2019.

Google DeepMind. Diffusiongemma. https://ai.google.dev/gemma/docs/diffusiongemma/model_card, 2026.

Samson Gourevitch, Yazid Janati, Dario Shariatian, Umut Simsekli, Eric Moulines, Eric P. Xing, and Alain Durmus. Uniform diffusion models revisited: Leave-one-out denoiser and absorbing state reformulation, 2026. URL https://arxiv.org/abs/2605.22765.

Satoshi Hayakawa, Yuhta Takida, Masaaki Imaizumi, Hiromi Wakaki, and Yuki Mitsufuji. Distillation of discrete diffusion through dimensional correlations. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, 2021.

Keya Hu, Linlu Qiu, Yiyang Lu, Hanhong Zhao, Tianhong Li, Yoon Kim, Jacob Andreas, and Kaiming He. Elf: Embedded language flows, 2026. URL https://arxiv.org/abs/2605.10938.

Jaehyeong Jo and Sung Ju Hwang. Continuous diffusion model for language modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Samuel Kotz, Narayanaswamy Balakrishnan, and Norman L. Johnson. *Continuous Multivariate Distributions, Volume 1: Models and Applications*. Wiley, 2 edition, 2000.

Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 1951.

Chanhyuk Lee, Jaehoon Yoo, Manan Agarwal, Sheel Shah, Jerry Huang, Aditi Raghunathan, Seunghoon Hong, Nicholas M. Boffi, and Jinwoo Kim. Flow map language models: One-step language modeling via continuous denoising, 2026. URL https://arxiv.org/abs/2602.16813.

Ian Li, Zilei Shao, Benjie Wang, Rose Yu, Guy Van den Broeck, and Anji Liu. Breaking the factorization barrier in diffusion language models. In *Forty-third International Conference on Machine Learning*, 2026.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems*, 2022.

Hung-Hsuan Lin, I-Chen Wu, and Tinghan Wei. On specific 17-clue sudoku puzzles. *ICGA Journal*, 2013.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2024.

Gary McGuire, Bastian Tugemann, and Gilles Civario. There is no 16-clue sudoku: Solving the sudoku minimum number of clues problem via hitting set enumeration. *Experimental Mathematics*, 2014.

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. In *The Thirteenth International Conference on Learning Representations*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023.

Patrick Pynadath, Jiaxin Shi, and Ruqi Zhang. Candi: Hybrid discrete-continuous diffusion models. In *International Conference on Machine Learning*, 2026.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Technical Report*, 2019.

Pierre Harvey Richemond, Sander Dieleman, and Arnaud Doucet. Categorical SDEs with simplex diffusion. In *ICML 2023 Workshop: Sampling and Optimization in Discrete Space*, 2023.

Daan Roos, Oscar Davis, Floor Eijkelboom, Michael Bronstein, Max Welling, İsmail İlkan Ceylan, Luca Ambrogioni, and Jan-Willem van de Meent. Categorical flow maps, 2026. URL https://arxiv.org/abs/2602.12233.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Forty-second International Conference on Machine Learning*, 2025.

Subham Sekhar Sahoo, Jean-Marie Lemercier, Zhihan Yang, Justin Deschenaux, Jingyu Liu, John Thickstun, and Ante Jukic. Scaling beyond masked diffusion language models. In *The Forty-Third International Conference on Machine Learning*, 2026.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P de Almeida, Alexander M Rush, Thomas PIERROT, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to DNA sequence design. In *Forty-first International Conference on Machine Learning*, 2024.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. In *Forty-second International Conference on Machine Learning*, 2025.

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.

Tianyu Xie, Shuchen Xue, Zijin Feng, Tianyang Hu, Jiacheng Sun, Zhenguo Li, and Cheng Zhang. Variational autoencoding discrete diffusion with enhanced dimensional correlations modeling. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ruixiang Zhang, Shuangfei Zhai, Yizhe Zhang, James Thornton, Zijing Ou, Joshua M. Susskind, and Navdeep Jaitly. Target concrete score matching: A holistic framework for discrete diffusion. In *Forty-second International Conference on Machine Learning*, 2025.

Lingxiao Zhao, Xueying Ding, Lijun Yu, and Leman Akoglu. Unified discrete diffusion for categorical data. *Journal of Machine Learning Research*, 2025.

Huangjie Zheng, Shansan Gong, Ruixiang Zhang, Tianrong Chen, Jiatao Gu, Mingyuan Zhou, Navdeep Jaitly, and Yizhe Zhang. Continuously augmented discrete diffusion model for categorical generative modeling. In *The Fourteenth International Conference on Learning Representations*, 2026.

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. In *The Thirteenth International Conference on Learning Representations*, 2025.

Simplex Relaxation for Discrete Diffusion

Supplementary Material

A OpenWebText Experimental Details

This section provides preprocessing, optimization, checkpoint, sampling, and evaluation details that are omitted from the main text. The dataset, tokenizer, sequence length, backbone, training budget, and entropy-matched evaluation protocol are summarized in Secs. 4.1 and 4.2.

A.1 Training Setup

Data preprocessing. We use the `openwebtext-train` and `openwebtext-valid` splits. Documents are concatenated with an end-of-sequence token inserted between adjacent documents and packed into fixed-length blocks of 1024 GPT-2 tokens.

Model architecture. All methods use the 179M-parameter diffusion transformer of Sahoo et al. (2024), consisting of 12 transformer blocks with rotary position embeddings (Su et al., 2024), AdaLN time conditioning (Peebles & Xie, 2023), and a softmax output head.

Optimization details. Models trained in our common codebase use Adam (Kingma & Ba, 2015) with learning rate 3×10^{-4}, batch size 512, $\beta_1 = 0.9$, $\beta_2 = 0.999$, numerical constant 10^{-8}, and gradient-norm clipping at 1.0. Training uses bfloat16 precision and a linear warmup over the first 2,500 optimizer steps, followed by a constant learning rate.

Simplax training and checkpoint. We evaluate both Simplax trained from scratch for 1M optimizer steps and a UDLM-initialized variant. The latter starts from the 800k-step UDLM checkpoint and is trained with the Simplax objective for the remaining 200k steps, keeping the total optimization budget at 1M steps. In our measured training configuration, a 1M-step run requires approximately ten days on eight NVIDIA H200 GPUs.

The network predicts the clean-token distribution and receives the categorical state $\mathbf{z}_t$ as input, while the relaxed state $\mathbf{w}_t$ remains in the training objective. We use a uniform time schedule, constant Dirichlet concentration $\eta = 0.01$, and constant loss weighting. We do not use an auxiliary self-conditioning input (Chen et al., 2023).

Dirichlet sampling and concentration-dependent computations are performed in float64. Concentration values are restricted to $[10^{-10}, 10^{8}]$ for numerical stability. Following the FLM implementation (Lee et al., 2026), before softmax we softly bound the output logits as $\ell \leftarrow 30 \tanh(\ell/30)$.

A.2 Baseline Checkpoints and Implementations

AR. We use the autoregressive OpenWebText checkpoint released with S-FLM at `https://huggingface.co/jdeschena/s-flm`. The checkpoint is evaluated without additional training.

CANDI. We use an OpenWebText checkpoint provided directly by the CANDI authors after contacting them. We evaluate the checkpoint using the implementation released with the method, without additional training.

UDLM. We could not identify a publicly available OpenWebText checkpoint for UDLM. We therefore train UDLM in our common codebase for 1M optimizer steps. The reported UDLM results use the resulting 1M-step checkpoint.

MDLM. We use the official OpenWebText checkpoint released by the authors at `https://huggingface.co/kuleshov-group/mdlm-owt`. The checkpoint is evaluated without additional training.

Duo. We use the official checkpoint released by the authors at `https://huggingface.co/s-sahoo/duo`. The checkpoint is evaluated without additional training.

Table 7: Sampling and terminal decoding used for OpenWebText and Sudoku experiments.

Method	Sampler	Terminal decoding
AR	Left-to-right autoregressive sampling	Token sampling
MDLM	Absorbing diffusion ancestral sampler	Ancestral sample
DUO / UDLM	Uniform diffusion ancestral sampler	Ancestral sample
FLM	Continuous-state Euler ODE sampler	Categorical Sample
Langflow	Euler–EDM	Temperature-scaled categorical sampling
SFM	Spherical flow sampler	Greedy argmax
Simplax	$\mathbf{w}$-deterministic sampler	Ancestal sample

FLM. We use the official OpenWebText checkpoint released by the authors at https://huggingface.co/david3684/FLM-B-OWT. The checkpoint is evaluated without additional training.

LangFlow. We use the official OpenWebText checkpoint released by the authors at https://huggingface.co/Continuous-Rivals-Discrete/langflow-owt. The checkpoint is evaluated without additional training.

S-FLM. We use the official OpenWebText checkpoint released by the authors at https://huggingface.co/jdeschena/s-flm. The checkpoint is evaluated without additional training.

A.3 Sampling and Evaluation

Tab. 7 summarizes the iterative sampler and terminal operation used in our evaluation.

A.4 Runtime and Memory Usage

Benchmark setup. All methods are benchmarked on the same GPU node equipped with one NVIDIA H200 GPU with 141 GB of HBM, 8 CPU cores, and 96 GB of CPU memory. For all methods, we compile the backbone and the major RNG-independent sampler components with `torch.compile`, using `dynamic=False` and `max_autotune=True`, with Triton CUDA graphs disabled. Random-number generation, including categorical sampling, remains eager. For AR, we enable KV caching and report eager execution, as compiling the backbone and sampling loop was slower in our benchmark. Each method–NFE configuration is evaluated in a fresh Python process with a fresh Inductor cache. We use the cached-integral implementation for CANDI and the $\mathbf{w}$-deterministic transition for Simplax. We preserve the numerical precision of each reference implementation: the backbone is evaluated in bfloat16 by default, while LangFlow uses its FP32 sampler, S-FLM its exact-velocity FP64 sampler.

Runtime. After loading the model and checkpoint, we perform two complete generation runs for warm-up and measure the following five runs. Each measurement is bracketed by CUDA synchronization and includes generation only, excluding model loading, EMA copying, compilation and warm-up, token decoding, perplexity evaluation, and I/O. We report the median wall-clock time $\pm$ sample standard deviation over the five runs, in seconds per batch; each batch contains 16 sequences of length 1024.

Memory. We run full garbage collection before each measured repeat, outside the timed region, and use the same generation-0 garbage-collection schedule within the sampler for all methods. We retain the CUDA allocator cache after warm-up. Immediately before each repeat, we reset the peak-memory statistics and report the maximum `torch.cuda.max_memory_allocated()` value. Thus, the reported memory is peak allocated CUDA tensor memory and includes the resident model and checkpoint. Thus, the reported memory is peak allocated CUDA tensor memory and includes the resident model and checkpoint. Table 8 summarizes the runtime and peak memory measurements.

Table 8: Sampling runtime and peak allocated GPU memory on OpenWebText. Runtime is median seconds per batch of 16 sequences of length 1024 ± sample standard deviation over five runs. Peak memory is the maximum observed across the three NFE settings. The best and second-best values among discrete diffusion methods are shown in bold and underlined, respectively.

Method	NFE = 16	NFE = 128	NFE = 1024	Peak mem. (GiB)
AR	-	-	8.463± 0.027	2.659
CANDI	<u>0.6483±0.0002</u>	<u>4.968±0.005</u>	<u>39.692±0.011</u>	26.52
UDLM	0.6765±0.0011	5.529±0.012	44.475±0.015	26.47
MDLM	**0.5477±0.0009**	**4.382±0.013**	**35.375±0.030**	<u>15.74</u>
Duo	0.6767±0.0008	5.526±0.009	44.490±0.011	26.48
FLM	0.8957±0.0016	7.122±0.006	57.027±0.003	18.81
LangFlow	1.1526±0.0019	9.377±0.016	75.259±0.030	**13.04**
S-FLM	1.1834±0.0036	9.977±0.015	80.489±0.009	20.77
Simplax	0.8141±0.0003	6.093±0.002	48.362±0.002	20.34

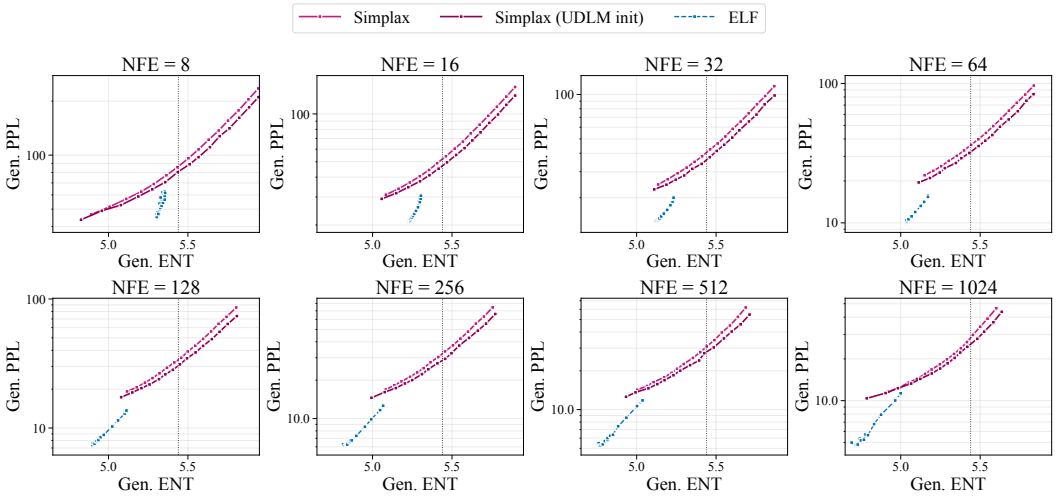

Figure 5: ELF and Simplax generative frontiers on OpenWebText under Llama-2 7B evaluation for NFE $\in \{8, 16, 32, 64, 128, 256, 512, 1024\}$. Each panel shows the Gen. PPL–Gen. ENT tradeoff obtained by sweeping temperature for Simplax and SC-CFG scale for ELF. Lower Gen. PPL is better, and the reference data entropy is 5.44.

A.5 Comparison with ELF

ELF (Hu et al., 2026) is a continuous-state language model based on flow matching in embedding space. Its generation trajectory remains continuous until the final step, where the predicted embedding is mapped to discrete tokens through a shared-weight decoder.

Because the intermediate trajectory does not involve categorical sampling, temperature can naturally be applied only to terminal token decoding and does not control the preceding continuous dynamics. Following the ELF evaluation protocol, we instead sweep self-conditioning classifier-free guidance (SC-CFG) to vary the quality–diversity tradeoff. We use the SDE sampler with a logit-normal time schedule, `sde_gamma`=1.5, and CFG scale 1.0, and sweep 15 SC-CFG scales: $\{0, 0.25, 0.5, 0.75, 1, 1.5, 2, 2.5, 3, 3.5, 4, 5, 6, 8, 10\}$.

As shown in Fig. 5, under Llama-2 7B evaluation, the ELF curves do not reach the OpenWebText data entropy of 5.44 at any evaluated NFE. ELF attains lower Gen. PPL than Simplax in the lower-entropy regime, but a direct matched-entropy comparison is therefore not available.

Table 9: OpenWebText generative perplexity evaluated by GPT-2 Large. The best and second-best values among discrete diffusion methods are shown in bold and underlined, respectively.

Method	NFE = 8	NFE = 16	NFE = 32	NFE = 64	NFE = 128	NFE = 256	NFE = 512	NFE = 1024
AR	-	-	-	-	-	-	-	22.82 ± 0.26
CANDI	170.78 ± 1.32	97.23 ± 0.85	80.05 ± 0.69	64.47 ± 0.61	67.17 ± 0.64	63.70 ± 0.68	75.83 ± 1.01	74.26 ± 0.98
UDLM	160.31 ± 1.13	186.03 ± 1.91	84.28 ± 0.66	62.23 ± 0.50	$\mathbf{54.97 \pm 0.49}$	57.16 ± 0.55	54.00 ± 0.51	59.24 ± 0.57
MDLM	195.83 ± 1.37	117.86 ± 0.86	72.73 ± 0.55	66.22 ± 0.56	65.37 ± 0.56	66.90 ± 0.63	68.65 ± 0.66	55.13 ± 0.54
Duo	221.00 ± 1.67	166.12 ± 1.81	96.99 ± 0.90	92.48 ± 1.00	93.91 ± 1.12	90.51 ± 1.16	88.63 ± 1.14	88.94 ± 1.10
FLM	569.26 ± 4.09	259.45 ± 1.78	146.13 ± 0.80	122.33 ± 0.69	112.02 ± 0.70	111.57 ± 0.84	122.76 ± 1.10	125.60 ± 1.29
LangFlow	211.22 ± 1.48	115.06 ± 0.91	80.70 ± 0.70	70.19 ± 0.65	60.20 ± 0.62	62.82 ± 0.83	67.35 ± 1.25	68.27 ± 1.75
S-FLM	150.61 ± 1.14	124.63 ± 1.17	116.29 ± 1.26	111.65 ± 1.30	103.39 ± 1.22	111.48 ± 1.37	109.53 ± 1.38	108.53 ± 1.36
Simplax	155.86 ± 1.26	98.72 ± 0.79	77.37 ± 0.61	64.74 ± 0.51	63.37 ± 0.53	60.43 ± 0.49	55.22 ± 0.47	52.55 ± 0.44
Simplax (UDLM init)	$\mathbf{149.13 \pm 1.26}$	$\mathbf{81.53 \pm 0.67}$	$\mathbf{70.00 \pm 0.58}$	$\mathbf{57.89 \pm 0.48}$	56.48 ± 0.51	$\mathbf{54.10 \pm 0.49}$	$\mathbf{48.60 \pm 0.51}$	$\mathbf{42.77 \pm 0.43}$

Table 10: OpenWebText generative perplexity evaluated by GPT-2 XL. The best and second-best values among discrete diffusion methods are shown in bold and underlined, respectively.

Method	NFE = 8	NFE = 16	NFE = 32	NFE = 64	NFE = 128	NFE = 256	NFE = 512	NFE = 1024
AR	-	-	-	-	-	-	-	24.12 ± 0.28
CANDI	174.46 ± 1.34	99.60 ± 0.88	82.43 ± 0.72	66.51 ± 0.63	69.15 ± 0.67	65.76 ± 0.71	78.12 ± 1.06	76.60 ± 1.02
UDLM	163.75 ± 1.16	190.07 ± 1.95	86.85 ± 0.69	64.14 ± 0.52	$\mathbf{56.71 \pm 0.51}$	58.92 ± 0.58	55.79 ± 0.53	61.15 ± 0.60
MDLM	199.60 ± 1.41	120.61 ± 0.88	74.62 ± 0.57	68.11 ± 0.58	67.18 ± 0.59	68.83 ± 0.65	70.52 ± 0.68	56.63 ± 0.56
Duo	223.93 ± 1.69	168.33 ± 1.84	99.03 ± 0.92	94.30 ± 1.02	95.85 ± 1.15	92.40 ± 1.19	90.44 ± 1.17	90.64 ± 1.13
FLM	572.55 ± 4.09	262.56 ± 1.81	148.21 ± 0.82	124.11 ± 0.70	113.72 ± 0.71	113.35 ± 0.86	124.47 ± 1.12	127.16 ± 1.31
LangFlow	212.92 ± 1.49	116.64 ± 0.92	82.33 ± 0.72	71.81 ± 0.67	61.69 ± 0.64	64.43 ± 0.86	69.12 ± 1.30	70.01 ± 1.83
S-FLM	152.51 ± 1.16	126.37 ± 1.20	118.05 ± 1.29	113.55 ± 1.34	105.08 ± 1.25	113.20 ± 1.39	111.19 ± 1.40	110.21 ± 1.39
Simplax	159.23 ± 1.29	101.32 ± 0.81	79.79 ± 0.64	66.75 ± 0.54	65.44 ± 0.55	62.43 ± 0.52	57.09 ± 0.48	54.50 ± 0.46
Simplax (UDLM init)	$\mathbf{152.50 \pm 1.30}$	$\mathbf{83.92 \pm 0.69}$	$\mathbf{72.11 \pm 0.60}$	$\mathbf{59.77 \pm 0.50}$	58.40 ± 0.53	$\mathbf{55.98 \pm 0.52}$	$\mathbf{50.32 \pm 0.53}$	$\mathbf{44.47 \pm 0.44}$

A.6 Additional OpenWebText Evaluation Results

The main text reports the temperature-swept generative frontier under Llama-2 7B evaluation and generative perplexities at NFE $= 16$, 128, and 1024 using GPT-2 Large and Llama-2 7B. Here, we additionally report the corresponding GPT-2 XL perplexities and provide evaluator-specific results with bootstrap standard errors for GPT-2 Large, GPT-2 XL, and Llama-2 7B (Tabs. 9 to 11). For each method and NFE, perplexity is evaluated at the temperature whose generated unigram entropy is closest to the OpenWebText entropy of 5.44.

Bootstrap uncertainty. Evaluating multiple independently generated sets of 1,024 sequences is computationally expensive, particularly at NFE $= 1024$. We therefore estimate sampling uncertainty using a sequence-level bootstrap. For each method and NFE, we fix the selected temperature before bootstrapping and resample the 1,024 generated sequences with replacement for 10,000 bootstrap replicates. For each replicate, generative perplexity is recomputed from the pooled token negative log-likelihood and token count of the resampled sequences. We report the standard deviation across bootstrap replicates as the bootstrap standard error. This uncertainty reflects sampling variability of generative perplexity for the fixed checkpoint and temperature, and does not include uncertainty from temperature selection or independently repeated generation runs. Tables 9 to 11 report PPL $\pm$ bootstrap standard error at the temperature whose generated unigram entropy is closest to 5.44.

We additionally report the full temperature-swept generative frontiers under GPT-2 Large and GPT-2 XL evaluation in Figs. 6 and 7. These evaluations use the same generated samples, temperatures, and generative entropy values as the main OpenWebText experiment; only the external language model used to compute generative perplexity is changed.

A.7 Qualitative Generations

Tabs. 12 to 14 show representative generations at NFE $= 16$, 128, and 1024. For each method and NFE, we select examples from the temperature whose generated unigram entropy is closest to the OpenWebText entropy of 5.44. The generated text is not manually rewritten. The excerpts are truncated at the positions marked by [. . .], and line wrapping is applied only for presentation.

Table 11: OpenWebText generative perplexity evaluated by Llama-2 7B. The best and second-best values among discrete diffusion methods are shown in bold and underlined, respectively.

Method	NFE = 8	NFE = 16	NFE = 32	NFE = 64	NFE = 128	NFE = 256	NFE = 512	NFE = 1024
AR	-	-	-	-	-	-	-	15.95 ± 0.18
CANDI	100.80 ± 0.66	56.03 ± 0.43	44.77 ± 0.33	36.71 ± 0.31	36.79 ± 0.31	35.22 ± 0.34	39.86 ± 0.53	39.17 ± 0.50
UDLM	84.47 ± 0.57	78.98 ± 0.67	44.81 ± 0.33	34.86 ± 0.26	31.51 ± 0.24	31.96 ± 0.27	30.62 ± 0.28	32.18 ± 0.29
MDLM	105.75 ± 0.74	63.84 ± 0.49	41.64 ± 0.29	38.02 ± 0.29	37.27 ± 0.29	37.64 ± 0.31	39.15 ± 0.35	33.88 ± 0.31
Duo	123.74 ± 0.85	87.11 ± 0.81	58.68 ± 0.52	53.98 ± 0.53	53.27 ± 0.60	50.22 ± 0.61	49.25 ± 0.62	50.39 ± 0.60
FLM	277.41 ± 2.55	139.27 ± 1.18	83.61 ± 0.48	71.79 ± 0.39	66.54 ± 0.34	67.62 ± 0.37	72.28 ± 0.43	74.35 ± 0.50
LangFlow	106.23 ± 0.74	59.89 ± 0.44	41.66 ± 0.32	35.17 ± 0.29	$\mathbf{30.03 \pm 0.28}$	29.68 ± 0.35	29.45 ± 0.47	28.16 ± 0.59
S-FLM	$\mathbf{76.46 \pm 0.48}$	62.90 ± 0.46	58.24 ± 0.49	55.88 ± 0.49	52.41 ± 0.47	54.99 ± 0.51	54.10 ± 0.51	53.75 ± 0.51
Simplax	85.66 ± 0.68	54.15 ± 0.44	42.50 ± 0.32	36.27 ± 0.27	34.98 ± 0.28	33.65 ± 0.28	30.93 ± 0.26	29.79 ± 0.27
Simplax (UDLM init)	80.45 ± 0.65	$\mathbf{45.21 \pm 0.37}$	$\mathbf{37.64 \pm 0.31}$	$\mathbf{31.74 \pm 0.24}$	31.12 ± 0.28	$\mathbf{29.40 \pm 0.26}$	$\mathbf{27.40 \pm 0.30}$	$\mathbf{24.43 \pm 0.26}$

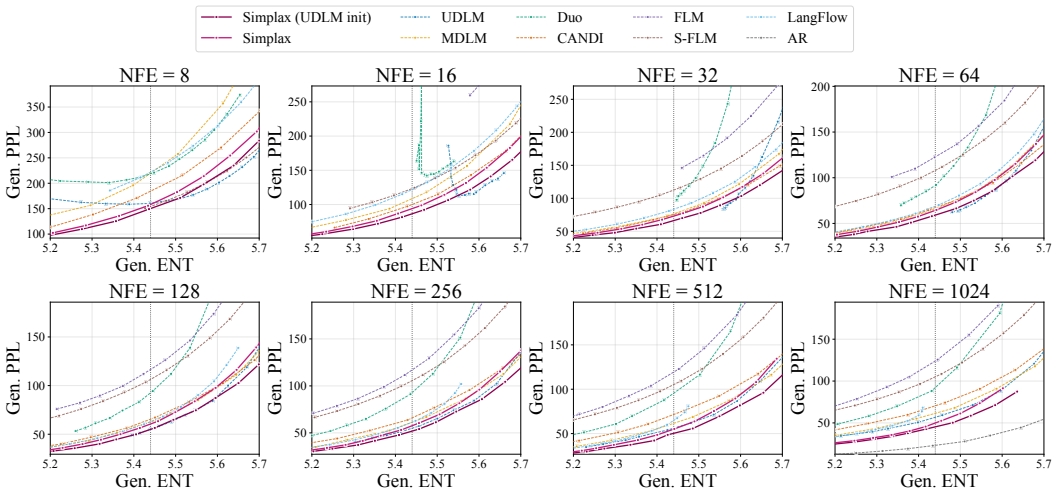

Figure 6: GPT-2 Large generative frontiers on OpenWebText for NFE $\in \{8, 16, 32, 64, 128, 256, 512, 1024\}$. Each panel shows the temperature-swept Gen. PPL–Gen. ENT tradeoff. Lower Gen. PPL is better, and the reference data entropy is 5.44.

B Sudoku Experimental Details

This section provides additional details on the Sudoku experiments, including dataset construction, sequence representation, model architectures, training, sampling, and qualitative generations.

B.1 Dataset and Sequence Representation

Dataset. All Sudoku puzzles used in our experiments have unique solutions. Following Deschenaux & Gulcehre (2026), we use the greedy Sudoku puzzle generator of Alp (2024) to construct the training data and the evaluation sets with 25 or more clues. The training set contains 48,000 puzzles with 30 observed cells, and all models are trained exclusively on this 30-clue training set. For evaluation, we use 2,000 puzzles for each clue count.

For the 20- and 17-clue settings, we instead use uniquely solvable 17-clue Sudoku puzzles studied by Lin et al. (2013). The 17-clue evaluation set uses these puzzles directly, while the 20-clue evaluation set is constructed by augmenting each 17-clue puzzle with three additional clues from its unique solution. Seventeen is the minimum number of clues for which a standard 9×9 Sudoku puzzle can admit a unique solution (McGuire et al., 2014; Lin et al., 2013).

At evaluation time, the same checkpoint trained exclusively on 30-clue puzzles is evaluated with 40, 35, 30, 25, 20, and 17 clues. The 30-clue setting matches the training clue density. The 40- and 35-clue settings provide more conditioning information than observed during training, whereas the 25-, 20-, and 17-clue settings progressively reduce the available conditioning information. For

Table 12: Representative unconditional OpenWebText generations at NFE $= 16$. Entropy is the generative unigram entropy in nats per token.

Method	Ent.	Generated text
CANDI	5.44	July on our records and the song in July, and we're signing people just for the next album, but it didn't couple with laser or anything. I had just a video. I thought they were crazy, because there were a lot of people out there who did it right. They felt like it was completely under my radar. [...]
UDLM	5.53	What with talk of a new bus un the altitude in the first place. In South Bend in November came the focus upon new cyclists – with an being builder. The other duck, Alabama. In total at least 82 bikes were donated to the department. [...]
MDLM	5.43	Select Rally and Comm Rally was a great game with strong designed scenarios, but I did think about no-one should play it as middle of the road. You've already started to love my opinion on Two vs Two. You should check it, and stick and play this once again! We again stack you. This is the final year for the year. [...]
Duo	5.45	I ever thought that it would be torrent be progression to try something new like my team base's football department. While I would have been optimistic about both the br of young players that hockey at the University at which went down in international football over the last of years and the way paths kept me focused so well I was pretty. [...]
FLM	5.58	Dec 16, 2015 Edit: I cant let my reader now clear that this painting looks like a collegecommunication." On Z. Thanks the box. that by the way I am out over to the river to buy your best chance to date for such a horrible year. But at which point after year they release the windows of 3,100 by 11 inches. St. [...]
LangFlow	5.42	God, will thrive. No others, and not at all, will determine our faith. We live in transient despair. But we don't know how to do it tomorrow. We want to live. We are sinners, we do not have God. God. we have identified the freedom of the one Being, is made to the only kind that governs our nation. [...]
S-FLM	5.45	I feel like, how much does that have to work out there? Are you going into a different development company? I think speaking about the timing and the reputation of what I mean as something that have really enjoyed my career doing. I kind of think the company can change things. [...]
Simplax	5.46	It was basically a percentage league, is there going to have to be one a year? They always were because they had them. If they knew that rule, they got to play one for their bodies. Then, when they voted for a "bolt 1" rule and what they got was, they want the fifth best defender to attack you, so they added [...]

Table 13: Representative unconditional OpenWebText generations at NFE $= 128$. Entropy is the generative unigram entropy in nats per token.

Method	Ent.	Generated text
CANDI	5.43	I think they should hear about their jobs more than they think. They know that they to hear about a huge percentage of young adult Americans that are helping create jobs. I realize that what our advisors and architects believe is true. They're in fact engaging in those American jobs, whether or not they've been doing it for several years. [...]
UDLM	5.50	The show has been going on for years, and people who tell the community about the end, and what way it works are already very exciting. But it is difficult to know whether you will disagree, even whether you feel the show expects a lot. So tell other friends and friends who are just around you, the producers and the media what [...]
MDLM	5.45	It's been difficult times with, for me and Brett. It's important to tackle this issue on the long haul. I plan on putting it on hold again once I have my business, but committed to showing up again by the deadline. There's a lot of effort involved in an issue about a loan for $20k, and it was in the middle, I [...]
Duo	5.44	America's social justice system. The ability of America's people to deal with those that are alike and those different is unclear. But when tough and bad, the outcomes are not very different." Congressman Kathy Lee interviewed for this article told me what she called her priority business when serving as lieutenant senator of New York. [...]
FLM	5.43	I think that we would have to bring back aggression by the usual course, with war with the past, on a situation. If it was done, the fascists were inside them or there was nobody stirred up. People were part of the invasion by the Nazis, then they shot, they went on. [...]
LangFlow	5.42	The body bloates with a layer of wood, black air that brings in your skin and exites aging. When reading about your life in the late 90s, what was the puzzle you were plotting in your life? I wanted to tell the secret of being an ideal person, and that everything that relies on that secret still exists. [...]
S-FLM	5.43	"You," asked Reeves, who had a light conversation with her. "You shouldn't sell a media conference. You have been around for weeks and you haven't come out to 83 game," Khanco said, "WHAT?" "Foreby, you have been making a fortune with free agency!" she replied. "I have to wonder what it was your secret. I plan to write everything with this team. [...]
Simplax	5.45	"I thought it was odd," Anna told me. "I wanted to know," Elsa said nervously. It's a lovely thing when you use words the way they mean something. You seem to like it. A lot." That was nice. "It needed to be answered," Anna said. "I just didn't say it since I grew up." I said. [...]

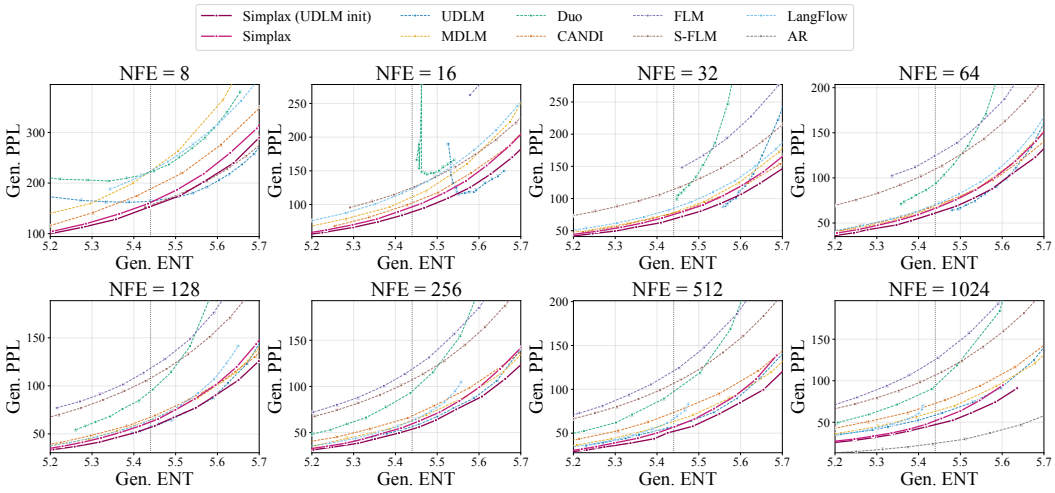

Figure 7: GPT-2 XL generative frontiers on OpenWebText for NFE $\in \{8, 16, 32, 64, 128, 256, 512, 1024\}$. Each panel shows the temperature-swept Gen. PPL–Gen. ENT tradeoff. Lower Gen. PPL is better, and the reference data entropy is 5.44.

Table 14: Representative unconditional OpenWebText generations at NFE $= 1024$. Entropy is the generative unigram entropy in nats per token.

Method	Ent.	Generated text
CANDI	5.46	Tip: Every single mistake you face, you don't pay for it all the time. This is lost confidence. Of course, in this article, you want to be booking your own grooming option, just to pay at your local time schedule. You have to have a monthly with the brand Indigo Tattoo Shop for the monthly fee. [...]
UDLM	5.45	Yeah: That's what I understand... and I'd presumably have come on board. I think it would be foolish to give something to him, but I guess he wouldn't want to write them, because he'd actually want it to remain in front of him for a few years. [...]
MDLM	5.44	No last of the songs have been put together – we really want the whole development going. I really hope it turns out, or the clock falls on its own itself, and if we start putting them together, things will just blow up almost completely. That's pretty scary. This will be our first time back in the studio. [...]
Duo	5.43	If for rats and pies. I've been digging for a hard break. With Ann – she'd have a long, bitter it by trying on. It never got there, I- She was just shy of puberty or a man. I wouldn't even want to tell my grandmother what was going on in her life. Those things went somewhere, baby. [...]
FLM	5.45	There can be something if you don't know you're someones not squad, that's got an entirely different breed but neither do instinctively believe what or everyone born with will do the thing. There are some of that kind of good will. You only do that stuff for nothing but fun. [...]
LangFlow	5.41	50. "My messy experience is everything" he said." Something that has been on for all my life, and I think I'm living with it, but that's tough when I start losing people's attention." He said he had accepted divorce for most of his family's care because he now relies on current degree's rent, healthcare, and social assistance children. [...]
S-FLM	5.45	I can't say no they. She's definitely a stockwoman. She's got a huge attitude. We don't make her look her the way she's someone that's popular. There's just a hint of that." If you would give us more time to the public on our #1 gender department, you can tell us your thoughts in the comments below. [...]
Simplax	5.44	How could I take it to him? Would I have a chat in my life? I didn't freak out. Even still, he did not have enough strength, and I was only millimeters away, and not even touching his chador. Only a rag with lain liquid it around his hair, and he could feel the innocence of a Playboy. [...]

the no-clue evaluation, all 81 cells in the puzzle prefix are replaced with the blank token, yielding unconditional Sudoku generation.

Sequence representation. The vocabulary contains 12 symbols: a blank-cell token, the digits 1–9, a row-separator token, and a `BOS` token. Each example is represented by 180 tokens:

$$[\texttt{BOS}] \;+\; \text{89-token puzzle} \;+\; [\texttt{BOS}] \;+\; \text{89-token solution}.$$

Each 89-token board representation contains 81 cell tokens and eight row separators. The resulting puzzle prefix has length 91, and the generated solution has length 89. Unobserved cells are represented explicitly by the blank token, so the prefix length remains 91 for every clue count. The training objective is evaluated only on the solution portion of the sequence.

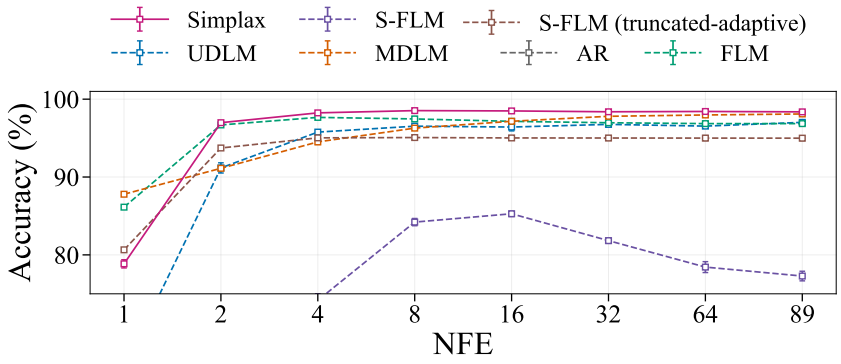

Figure 8: Accuracy versus NFE with 40 clues.

B.2 Model and Training Details

Shared architecture. All methods use Transformer models with hidden dimension 512, eight Transformer blocks, eight attention heads of dimension 64, and dropout probability 0.1. Their parameter counts range from 25.21M to 28.59M, with the principal differences across methods arising from the training objective, time conditioning, and method-specific input and output components. The models use learned token embeddings without embedding–output weight tying. The autoregressive model uses causal attention and no time conditioning. The remaining models use bidirectional attention and AdaLN-based time conditioning with conditioning dimension 128. The methods additionally differ in their inference procedures, as described in the corresponding method definitions.

Optimization. All models are trained for 20k optimization steps with global batch size 256 using bfloat16 precision. We use Adam with learning rate 3×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and gradient-norm clipping at 1.0. The learning rate is linearly warmed up for 2,500 steps and then held constant. We maintain an EMA of the parameters with decay 0.9999 and use the averaged parameters for evaluation. Training for 20k steps requires approximately 30 minutes on one NVIDIA H200 GPU.

B.3 Sampling and Decoding Details

Table 7 summarizes the samplers and terminal decoding rules used for the Sudoku experiments.

B.4 Additional Few-Step Results

Figure 3 reports few-step generation with 30 clues. We additionally report results for the same models, trained exclusively on 30-clue puzzles, under 40-, 35-, and 25-clue evaluation settings in Figs. 8 to 10. Across these clue counts, Simplax maintains strong performance even at low NFE, showing that its few-step generation behavior extends beyond the 30-clue training setting.

B.5 Qualitative Generation Example

To complement the aggregate Sudoku results in the main text, we provide a qualitative comparison of the generated solutions for a representative 25-clue puzzle (Fig. 11). This example illustrates the types of constraint violations that are not visible from solving accuracy alone.

C Contribution Analysis Details and Frontiers

This section provides additional details for the input, reverse-sampler, bridge-concentration, training–sampling, and budget-allocation analyses on OpenWebText. We also report the reverse-sampler runtime protocol and the corresponding temperature-swept Gen. PPL–Gen. ENT curves.

Input. We compare Simplax trained and evaluated with $\mathbf{z}_t$ input against Simplax trained and evaluated with $\mathbf{w}_t$ input. Both models are trained for 50k steps with the Simplax objective and $\eta = 0.01$,

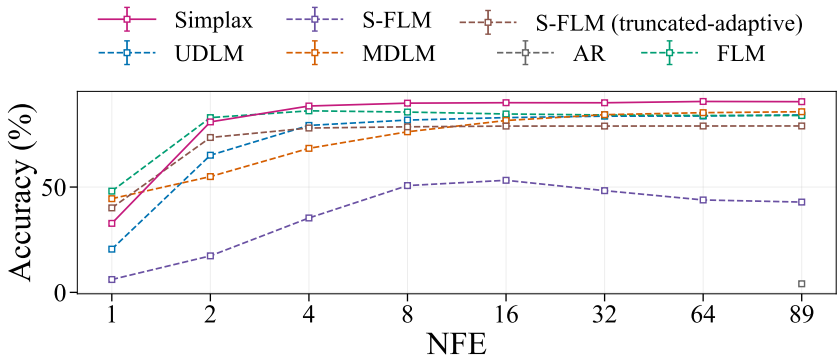

Figure 9: Accuracy versus NFE with 35 clues.

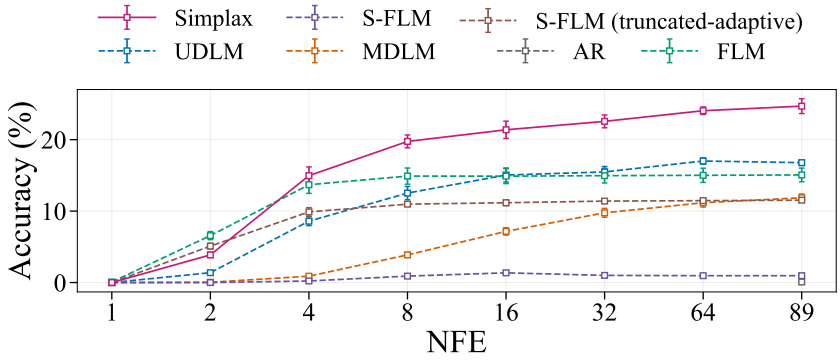

Figure 10: Accuracy versus NFE with 25 clues.

and use stochastic sampling at inference (Fig. 12). We use $\mathbf{z}_t$ in the main experiments because it yields the better frontier, especially at high NFE, while also avoiding the dense vocabulary-sized embedding multiplication required by $\mathbf{w}_t$ input.

Sampler quality and runtime. We compare the four Simplax samplers considered in the main text: *stochastic*, $\mathbf{w}$-*deterministic*, which makes only the $\mathbf{w}$ update deterministic, $\mathbf{z}$-*deterministic*, which makes only the $\mathbf{z}$ update deterministic, and *deterministic*. For the frontier analysis, we additionally evaluate an *expectation* sampler. Their update rules are detailed in Sec. H. All variants use the same Simplax checkpoint trained for 50k steps with $\mathbf{z}_t$ input and $\eta = 0.01$; only the sampler is changed.

Runtime and peak memory for the four main samplers are measured using the protocol described in Sec. A.4 and reported in Tab. 15. The *stochastic* and $\mathbf{w}$-*deterministic* samplers yield similar frontiers (Fig. 13), whereas $\mathbf{z}$-*deterministic* fails to recover the target entropy. The *deterministic* and *expectation* samplers collapse to low entropy and fail to reach the data entropy. The $\mathbf{w}$-*deterministic* sampler avoids the expensive Dirichlet sampling step, which requires FP64 sampler computations, thereby reducing both runtime and peak memory. We therefore use $\mathbf{w}$-*deterministic* sampling in the main experiments, which preserves the *stochastic* sampler's quality while reducing generation time.

Bridge concentration. We compare $\eta \in \{0, 0.001, 0.01, 0.1\}$ using Simplax models trained for 50k steps with $\mathbf{z}_t$ input and evaluated with the stochastic sampler (Fig. 14). Each model is trained with the same concentration used at inference. Small concentrations give similar low-NFE behavior, while $\eta = 0.01$ provides a better high-NFE frontier; $\eta = 0.1$ becomes less reliable and fails to reach the target entropy at high NFE. We therefore use $\eta = 0.01$ in the main experiments.

Training–sampling combinations. We compare the UDLM 1M checkpoint with the UDLM-initialized Simplax checkpoint trained for 800k UDLM steps followed by 200k Simplax steps. Each checkpoint is evaluated with both the native UDLM ancestral sampler and the Simplax $\mathbf{w}$-deterministic sampler (Fig. 15). For Simplax, we use $\eta = 0.01$ and $\mathbf{z}_t$ input.

Input puzzle 25 clues

				6		9		
							3	1
3	2					4	5	8
1	8							9
						3	7	6
		7						
8				2	4			
5								
		2	6		3	1		5

Ground truth reference

4	1	8	3	6	5	9	2	7
9	7	5	8	4	2	6	3	1
3	2	6	9	1	7	4	5	8
1	8	3	2	7	6	5	4	9
2	9	4	1	5	8	3	7	6
6	5	7	4	3	9	8	1	2
8	6	1	5	2	4	7	9	3
5	3	9	7	8	1	2	6	4
7	4	2	6	9	3	1	8	5

AR invalid · 42 errors

7	5	1	4	6	8	9	2	3
4	9	8	5	7	2	6	3	1
3	2	6	9	1	1	4	5	8
1	8	5	3	4	6	7	4	9
2	4	9	1	8	5	3	7	6
6	3	7	2	9	7	5	1	4
8	1	3	7	2	4	5	6	9
5	6	4	8	3	9	2	1	7
9	7	2	6	5	3	1	8	5

MDLM invalid · 8 errors

4	1	8	3	6	5	9	2	7
9	7	5	8	4	2	6	3	1
3	2	6	7	1	9	4	5	8
1	8	3	2	7	6	5	4	9
2	5	4	9	5	8	3	7	6
6	9	7	4	3	5	8	1	2
8	6	1	5	2	4	7	9	3
5	3	9	1	8	7	2	6	4
7	4	2	6	9	3	1	8	5

UDLM invalid · 25 errors

4	1	5	3	6	8	9	2	7
9	7	8	4	5	2	6	3	1
3	2	6	7	9	1	4	5	8
1	8	3	2	7	6	5	4	9
2	5	4	8	1	9	3	7	6
6	9	7	9	4	5	8	1	2
8	9	1	5	2	4	7	6	3
5	6	9	1	3	7	2	8	4
7	4	2	6	8	3	1	9	5

FLM invalid · 26 errors

4	5	1	3	6	8	9	2	7
9	7	8	4	5	2	6	3	1
3	2	6	7	1	9	4	5	8
1	8	7	2	3	6	5	4	9
2	9	5	8	4	1	3	7	6
6	3	7	9	4	5	8	1	2
8	1	9	5	2	4	7	6	3
5	6	3	1	8	7	2	9	4
7	4	2	6	9	3	1	8	5

S-FLM invalid · 25 errors

4	5	1	3	6	8	9	2	7
9	7	8	4	5	2	6	3	1
3	2	6	1	9	7	4	5	8
1	8	3	2	7	6	5	4	9
2	4	5	8	1	9	3	7	6
6	9	7	4	3	5	8	1	2
8	1	9	5	2	4	7	6	3
5	6	3	9	7	1	2	8	4
7	4	2	6	8	3	1	9	5

S-FLM (trunc.) invalid · 27 errors

4	5	1	3	6	8	9	2	7
9	7	8	4	5	2	6	3	1
3	2	6	7	1	9	4	5	8
1	8	4	2	7	6	5	4	9
2	9	5	8	4	1	3	7	6
6	4	7	9	3	5	8	1	2
8	1	9	5	2	4	7	6	3
5	6	4	1	9	7	2	8	4
7	4	2	6	8	3	1	9	5

Simplax valid

4	1	8	3	6	5	9	2	7
9	7	5	8	4	2	6	3	1
3	2	6	9	1	7	4	5	8
1	8	3	2	7	6	5	4	9
2	9	4	1	5	8	3	7	6
6	5	7	4	3	9	8	1	2
8	6	1	5	2	4	7	9	3
5	3	9	7	8	1	2	6	4
7	4	2	6	9	3	1	8	5

Figure 11: Generation results for a representative 25-clue Sudoku puzzle at NFE $= 89$. Blue entries are given clues, black entries agree with the unique solution, and red entries differ from it. All methods receive the same puzzle prefix. Simplax produces the valid solution in this example, whereas the other methods violate at least one Sudoku constraint.

Budget allocation. We compare UDLM 1M and four staged allocations: 600k/400k, 700k/300k, 800k/200k, and 900k/100k UDLM/Simplax steps (Tab. 16 and Fig. 16). All staged variants perform similarly and consistently benefit from UDLM initialization relative to training Simplax from scratch. We use the 800k/200k allocation in the main experiments.

D Auxiliary Identities and Full Dirichlet–Categorical Hierarchy

This section develops the exact probabilistic structure behind the simplex relaxation. The formulas below apply at positive diffusion times, where $\mathbf{p}_t$ is strictly positive under the assumptions in Sec. 2. We define the clean endpoint separately as the categorical state $\mathbf{z}_0 = \mathbf{x}$; the hierarchy does not introduce a Dirichlet variable $\mathbf{w}_0$.

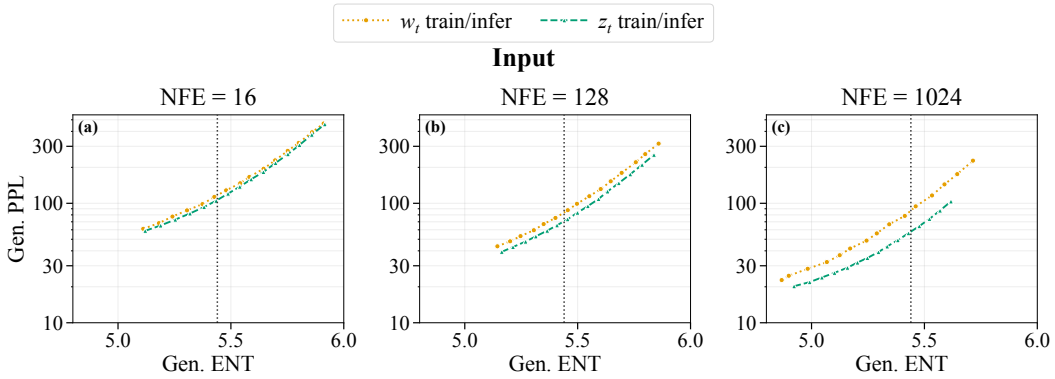

Figure 12: Temperature-swept input ablation on OpenWebText at NFE $=$ 16, 128, and 1024. The vertical dotted line marks the data entropy, 5.44.

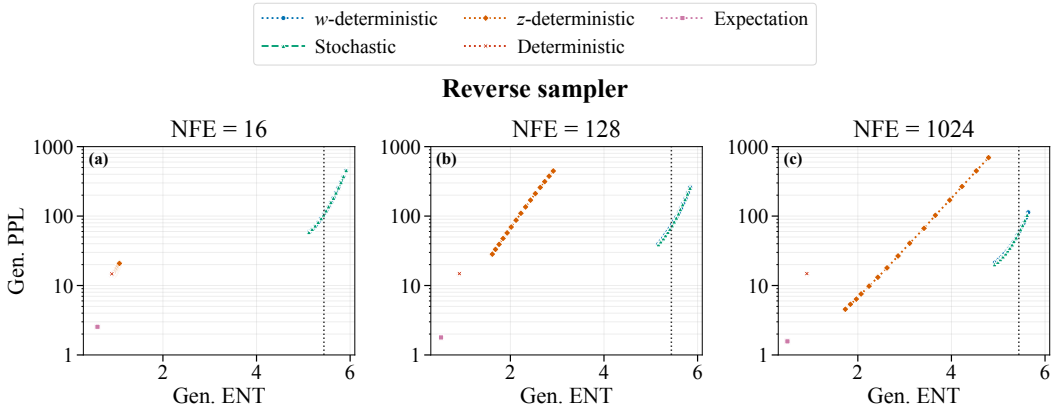

Figure 13: Temperature-swept reverse-sampler comparison on OpenWebText at NFE $=$ 16, 128, and 1024. The vertical dotted line marks the data entropy, 5.44.

The key point is that the auxiliary state $\mathbf{w}_t$ is not introduced as a heuristic soft surrogate. Rather, once we specify the Dirichlet bridge

$$q(\mathbf{w}_t \mid \mathbf{z}_t, \mathbf{x}) = \mathrm{Dir}(\mathbf{w}_t; \eta_t \mathbf{p}_t + \mathbf{z}_t)\,,$$

the resulting joint model admits a closed hierarchy in both directions: the relaxed state has an exact Dirichlet marginal, the discrete state can be decoded exactly from $\mathbf{w}_t$, and the reverse bridge remains tractable after marginalizing either $\mathbf{z}_t$ or $\mathbf{z}_s$. We begin with two elementary identities that make these cancellations possible.

D.1 Useful identities

The first identity is the basic shift formula for Dirichlet densities. It shows that adding a one-hot count to the concentration vector simply multiplies the base Dirichlet density by the corresponding simplex coordinate.

Lemma 1. *Let $\boldsymbol{\alpha} \in \mathbb{R}^K_{>0}$ and let $\alpha_0 = \sum_{i=1}^K \alpha_i$. Then, for any $k \in \{1, \dots, K\}$,*

$$\mathrm{Dir}(\mathbf{w}; \boldsymbol{\alpha} + \mathbf{e}_k) = \frac{\alpha_0}{\alpha_k}\, w_k\, \mathrm{Dir}(\mathbf{w}; \boldsymbol{\alpha}). \tag{21}$$

Proof. By definition,

$$\mathrm{Dir}(\mathbf{w}; \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^K w_i^{\alpha_i - 1}, \qquad B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma(\alpha_0)}.$$

Table 15: Runtime and peak allocated GPU memory of Simplax samplers on OpenWebText. Runtime is median seconds per batch of 16 sequences of length 1024 ± sample standard deviation over five runs. Peak memory is the maximum observed across the three NFE settings. The best and second-best values are shown in bold and underlined, respectively.

Sampler	NFE = 16	NFE = 128	NFE = 1024	Peak mem. (GiB)
Stochastic	1.6722 ± 0.0001	12.975 ± 0.001	103.337 ± 0.010	26.48
$\mathbf{w}$-deterministic	0.8136 ± 0.0005	6.101 ± 0.002	48.338 ± 0.003	**20.34**
$\mathbf{z}$-deterministic	1.3867 ± 0.0004	10.742 ± 0.004	85.478 ± 0.011	26.48
Deterministic	$\underline{0.6566 \pm 0.0004}$	$\underline{4.887 \pm 0.004}$	$\underline{38.834 \pm 0.007}$	**20.34**
Expectation	$\mathbf{0.6148 \pm 0.0002}$	$\mathbf{4.5390 \pm 0.0032}$	$\mathbf{35.9733 \pm 0.0294}$	**20.34**

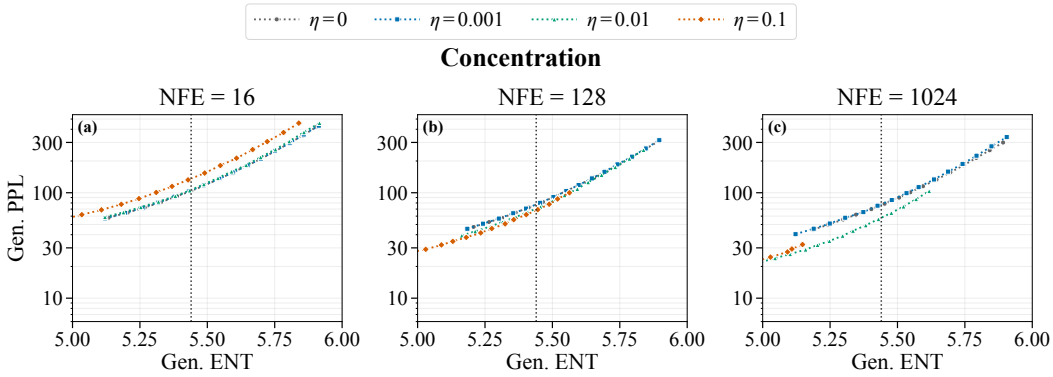

Figure 14: Temperature-swept bridge-concentration comparison on OpenWebText at NFE = 16, 128, and 1024. The vertical dotted line marks the data entropy, 5.44.

Hence

$$\mathrm{Dir}(\mathbf{w}; \boldsymbol{\alpha} + \mathbf{e}_k) = \frac{1}{B(\boldsymbol{\alpha} + \mathbf{e}_k)} w_k \prod_{i=1}^{K} w_i^{\alpha_i - 1}.$$

It remains to compare the normalizing constants:

$$\frac{B(\boldsymbol{\alpha} + \mathbf{e}_k)}{B(\boldsymbol{\alpha})} = \frac{\Gamma(\alpha_k + 1)}{\Gamma(\alpha_k)} \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_0 + 1)} = \frac{\alpha_k}{\alpha_0}.$$

Therefore

$$\frac{1}{B(\boldsymbol{\alpha} + \mathbf{e}_k)} = \frac{\alpha_0}{\alpha_k} \frac{1}{B(\boldsymbol{\alpha})},$$

which proves Eq. (21). □

Specializing this identity to concentrations of the form $\eta\mathbf{p}$ yields the cancellation that will be used throughout the appendix.

Corollary 1. *Let $\mathbf{p} \in \Delta^{K-1}$ satisfy $p_k > 0$ for all k, let $\eta > 0$, and define $\boldsymbol{\alpha} = \eta\mathbf{p}$. Then*

$$p_k \, \mathrm{Dir}(\mathbf{w}; \eta\mathbf{p} + \mathbf{e}_k) = w_k \, \mathrm{Dir}(\mathbf{w}; \eta\mathbf{p}). \tag{22}$$

Proof. Apply lemma 1 with $\boldsymbol{\alpha} = \eta\mathbf{p}$. Since $\alpha_0 = \eta$ and $\alpha_k = \eta p_k$,

$$\mathrm{Dir}(\mathbf{w}; \eta\mathbf{p} + \mathbf{e}_k) = \frac{\eta}{\eta p_k} w_k \, \mathrm{Dir}(\mathbf{w}; \eta\mathbf{p}) = \frac{w_k}{p_k} \, \mathrm{Dir}(\mathbf{w}; \eta\mathbf{p}).$$

Multiplying both sides by p_k gives Eq. (22). □

The content of corollary 1 is simple but important: a categorical mixture over one-hot shifts of a Dirichlet distribution collapses back to the unshifted Dirichlet density. This is precisely the mechanism that makes the simplex relaxation exact rather than approximate.

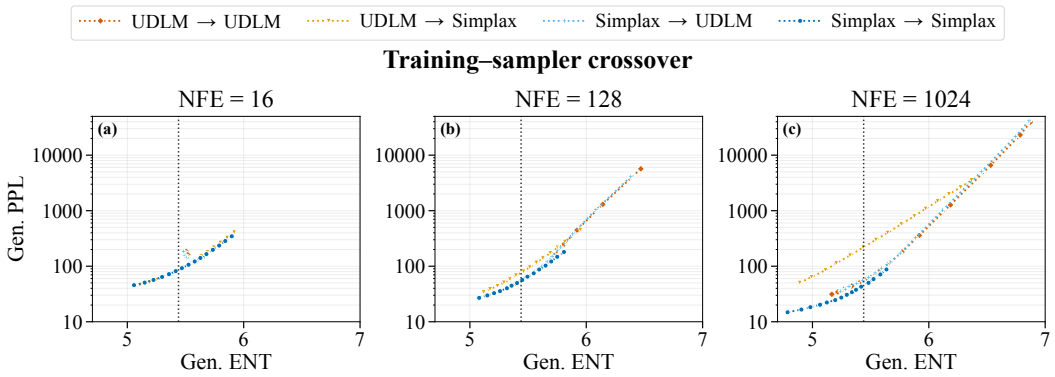

Figure 15: Temperature-swept training–sampling comparison on OpenWebText at NFE $= 16$, 128, and 1024. The vertical dotted line marks the data entropy, 5.44.

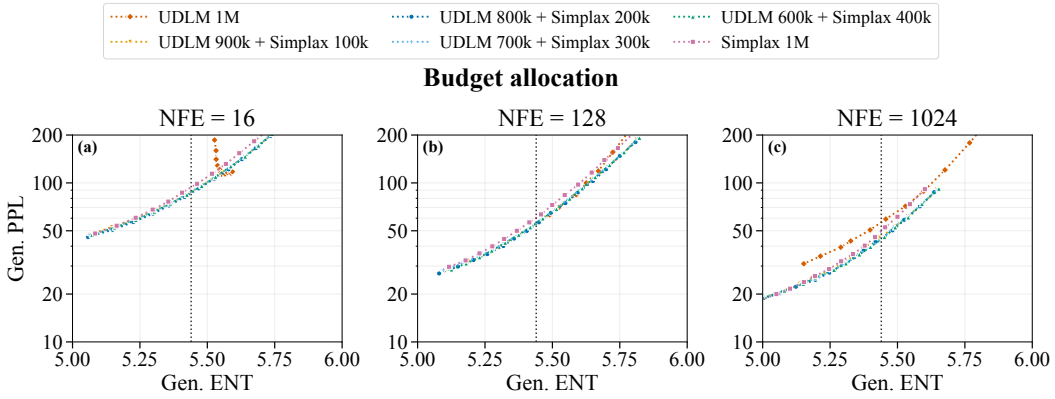

Figure 16: Temperature-swept budget-allocation comparison on OpenWebText at NFE $= 16$, 128, and 1024. The vertical dotted line marks the data entropy, 5.44.

D.2 Full Dirichlet–categorical hierarchy

We now return to the joint factorization

$$q(\mathbf{x}, \mathbf{z}_s, \mathbf{z}_t, \mathbf{w}_s, \mathbf{w}_t) = q(\mathbf{x})\, q(\mathbf{z}_s \mid \mathbf{x})\, q(\mathbf{w}_s \mid \mathbf{z}_s, \mathbf{x})\, q(\mathbf{z}_t \mid \mathbf{z}_s)\, q(\mathbf{w}_t \mid \mathbf{z}_t, \mathbf{x}), \tag{23}$$

together with

$$q(\mathbf{w}_t \mid \mathbf{z}_t, \mathbf{x}) = \mathrm{Dir}(\mathbf{w}_t; \eta_t \mathbf{p}_t + \mathbf{z}_t)\,. \tag{24}$$

The next proposition summarizes the full hierarchy induced by this construction. The first two statements identify the exact marginal and exact decoder at time t. The third lifts the standard discrete reverse posterior from $\mathbf{z}_t$ to $\mathbf{w}_t$. The last two show that, once this lift is performed, the reverse bridge over relaxed states becomes a mixture of shifted Dirichlet components.

Proposition 1. *Assume $t > 0$. The Dirichlet–categorical hierarchy satisfies the following properties; statements involving $\mathbf{w}_s$ additionally require $s > 0$.*

1. *The marginal distribution of the relaxed state is*

$$q(\mathbf{w}_t \mid \mathbf{x}) = \mathrm{Dir}(\mathbf{w}_t; \eta_t \mathbf{p}_t)\,. \tag{25}$$

2. *Given $\mathbf{w}_t$, the variables $\mathbf{x}$ and $\mathbf{z}_t$ are conditionally independent, and the discrete state can be recovered from the relaxed state via*

$$q(\mathbf{z}_t \mid \mathbf{w}_t, \mathbf{x}) = q(\mathbf{z}_t \mid \mathbf{w}_t) = \mathrm{Cat}(\mathbf{z}_t; \mathbf{w}_t)\,. \tag{26}$$

Table 16: Fixed-budget allocation between UDLM pretraining and Simplax training. We report GPT-2 Large Gen. PPL / Gen. ENT.

	NFE = 16	NFE = 128	NFE = 1024
Allocation	PPL / ENT	PPL / ENT	PPL / ENT
Simplax 1M	98.7/5.46	63.4/5.46	52.6/5.45
UDLM 600k → Simplax 400k	84.2/5.42	52.6/5.42	45.6/5.45
UDLM 700k → Simplax 300k	84.0/5.42	52.4/5.42	45.4/5.45
UDLM 800k → Simplax 200k	81.5/5.42	56.5/5.45	42.8/5.42
UDLM 900k → Simplax 100k	84.5/5.42	53.0/5.42	45.2/5.44
UDLM 1M	186.0/5.53†	62.9/5.49†	59.2/5.46

† marks sweeps that do not reach the data entropy 5.44 ± 0.05; their PPL is not directly comparable at matched entropy. The shaded row is used in the main experiments.

3. *For $s < t$, the reverse conditional posterior of $\mathbf{z}_s$ given $\mathbf{w}_t$ and $\mathbf{x}$ is categorical:*

$$q(\mathbf{z}_s \mid \mathbf{w}_t, \mathbf{x}) = \mathrm{Cat}\big(\mathbf{z}_s; \boldsymbol{\rho}_{s|t}(\mathbf{x}, \mathbf{w}_t)\big)\,, \tag{27}$$

where

$$\boldsymbol{\rho}_{s|t}(\mathbf{x}, \mathbf{w}_t) \coloneqq \mathbf{p}_s \odot \big[\alpha_{t|s}\big(\mathbf{w}_t \oslash \mathbf{p}_t\big) + (1 - \alpha_{t|s})\ \langle \mathbf{w}_t,\ \boldsymbol{\pi} \oslash \mathbf{p}_t \rangle\ \mathbf{1}\big]\,. \tag{28}$$

4. *For $0 < s < t$, the reverse conditional posterior of $\mathbf{w}_s$ given $\mathbf{z}_t$ and $\mathbf{x}$ is a Dirichlet mixture:*

$$q(\mathbf{w}_s \mid \mathbf{z}_t, \mathbf{x}) = \sum_{k=1}^{K} r_{s|t,k}(\mathbf{x}, \mathbf{z}_t)\ \mathrm{Dir}(\mathbf{w}_s; \eta_s \mathbf{p}_s + \mathbf{e}_k)\,, \tag{29}$$

where $r_{s|t,k}(\mathbf{x}, \mathbf{z}_t)$ denotes the k-th component of $\mathbf{r}_{s|t}(\mathbf{x}, \mathbf{z}_t)$ defined in Eq. (4).

5. *For $0 < s < t$, the reverse conditional posterior of $\mathbf{w}_s$ given $\mathbf{w}_t$ and $\mathbf{x}$ is a Dirichlet mixture:*

$$q(\mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x}) = \sum_{k=1}^{K} \rho_{s|t,k}(\mathbf{x}, \mathbf{w}_t)\ \mathrm{Dir}(\mathbf{w}_s; \eta_s \mathbf{p}_s + \mathbf{e}_k)\,. \tag{30}$$

Proof. 1. We begin with the marginal law of $\mathbf{w}_t$. Marginalizing the discrete latent $\mathbf{z}_t \sim \mathrm{Cat}(\mathbf{p}_t)$ from the conditional bridge

$$q(\mathbf{w}_t \mid \mathbf{z}_t, \mathbf{x}) = \mathrm{Dir}(\mathbf{w}_t; \eta_t \mathbf{p}_t + \mathbf{z}_t)$$

gives

$$\begin{aligned} q(\mathbf{w}_t \mid \mathbf{x}) &= \sum_{k=1}^{K} q(\mathbf{w}_t \mid \mathbf{z}_t = \mathbf{e}_k, \mathbf{x})\, q(\mathbf{z}_t = \mathbf{e}_k \mid \mathbf{x}) \\ &= \sum_{k=1}^{K} \mathrm{Dir}(\mathbf{w}_t; \eta_t \mathbf{p}_t + \mathbf{e}_k)\, p_{t,k}(\mathbf{x}). \end{aligned} \tag{31}$$

Now corollary 1 applies termwise:

$$p_{t,k}(\mathbf{x})\, \mathrm{Dir}(\mathbf{w}_t; \eta_t \mathbf{p}_t + \mathbf{e}_k) = w_{t,k}\, \mathrm{Dir}(\mathbf{w}_t; \eta_t \mathbf{p}_t)\,.$$

Substituting this into Eq. (31) collapses the mixture:

$$\begin{aligned} q(\mathbf{w}_t \mid \mathbf{x}) &= \sum_{k=1}^{K} w_{t,k}\, \mathrm{Dir}(\mathbf{w}_t; \eta_t \mathbf{p}_t) \\ &= \left(\sum_{k=1}^{K} w_{t,k}\right) \mathrm{Dir}(\mathbf{w}_t; \eta_t \mathbf{p}_t) \\ &= \mathrm{Dir}(\mathbf{w}_t; \eta_t \mathbf{p}_t)\,, \end{aligned} \tag{32}$$

which proves Eq. (25).

2. We next derive the exact decoder from $\mathbf{w}_t$ back to $\mathbf{z}_t$. Fix $k \in \{1, \dots, K\}$. By Bayes' rule,

$$q(\mathbf{z}_t = \mathbf{e}_k \mid \mathbf{w}_t, \mathbf{x}) = \frac{q(\mathbf{w}_t \mid \mathbf{z}_t = \mathbf{e}_k, \mathbf{x})\, q(\mathbf{z}_t = \mathbf{e}_k \mid \mathbf{x})}{q(\mathbf{w}_t \mid \mathbf{x})}. \tag{33}$$

Using the previous computation together with corollary 1, the numerator becomes

$$q(\mathbf{w}_t \mid \mathbf{z}_t = \mathbf{e}_k, \mathbf{x})\, q(\mathbf{z}_t = \mathbf{e}_k \mid \mathbf{x}) = w_{t,k} \operatorname{Dir}(\mathbf{w}_t; \eta_t \mathbf{p}_t)\,,$$

while the denominator is exactly $\operatorname{Dir}(\mathbf{w}_t; \eta_t \mathbf{p}_t)$. Hence

$$q(\mathbf{z}_t = \mathbf{e}_k \mid \mathbf{w}_t, \mathbf{x}) = w_{t,k}.$$

Since this holds for every k, we obtain

$$q(\mathbf{z}_t \mid \mathbf{w}_t, \mathbf{x}) = \operatorname{Cat}(\mathbf{z}_t; \mathbf{w}_t),$$

and the right-hand side no longer depends on $\mathbf{x}$. This proves Eq. (26).

3. We now lift the discrete reverse posterior from $\mathbf{z}_t$ to $\mathbf{w}_t$. Marginalizing over $\mathbf{z}_t$ gives

$$q(\mathbf{z}_s \mid \mathbf{w}_t, \mathbf{x}) = \sum_{j=1}^{K} q(\mathbf{z}_s \mid \mathbf{z}_t = \mathbf{e}_j, \mathbf{w}_t, \mathbf{x})\, q(\mathbf{z}_t = \mathbf{e}_j \mid \mathbf{w}_t, \mathbf{x}). \tag{34}$$

Conditioned on $(\mathbf{z}_t, \mathbf{x})$, the variable $\mathbf{z}_s$ is independent of $\mathbf{w}_t$, so the first factor is just the usual reverse posterior

$$q(\mathbf{z}_s \mid \mathbf{z}_t = \mathbf{e}_j, \mathbf{w}_t, \mathbf{x}) = q(\mathbf{z}_s \mid \mathbf{z}_t = \mathbf{e}_j, \mathbf{x}) = \operatorname{Cat}\big(\mathbf{z}_s; \mathbf{r}_{s|t}(\mathbf{x}, \mathbf{e}_j)\big)\,.$$

The second factor is the exact decoder derived above:

$$q(\mathbf{z}_t = \mathbf{e}_j \mid \mathbf{w}_t, \mathbf{x}) = w_{t,j}.$$

Therefore

$$q(\mathbf{z}_s \mid \mathbf{w}_t, \mathbf{x}) = \sum_{j=1}^{K} w_{t,j}\ \operatorname{Cat}\big(\mathbf{z}_s; \mathbf{r}_{s|t}(\mathbf{x}, \mathbf{e}_j)\big)\,. \tag{35}$$

A mixture of categorical distributions is again categorical, with parameter vector equal to the same convex combination of the component parameters. Thus

$$q(\mathbf{z}_s \mid \mathbf{w}_t, \mathbf{x}) = \operatorname{Cat}\big(\mathbf{z}_s; \boldsymbol{\rho}_{s|t}(\mathbf{x}, \mathbf{w}_t)\big)\,, \qquad \boldsymbol{\rho}_{s|t}(\mathbf{x}, \mathbf{w}_t) = \sum_{j=1}^{K} w_{t,j}\, \mathbf{r}_{s|t}(\mathbf{x}, \mathbf{e}_j).$$

To obtain the closed form, substitute Eq. (4) with $\mathbf{z}_t = \mathbf{e}_j$:

$$\begin{aligned} \mathbf{r}_{s|t}(\mathbf{x}, \mathbf{e}_j) &= \frac{\big[\alpha_{t|s}\mathbf{e}_j + (1-\alpha_{t|s})\ \langle \mathbf{e}_j,\ \boldsymbol{\pi}\rangle\ \mathbf{1}\big] \odot \mathbf{p}_s}{\langle \mathbf{e}_j,\ \mathbf{p}_t\rangle} \\ &= \mathbf{p}_s \odot \left[\alpha_{t|s} \frac{\mathbf{e}_j}{p_{t,j}(\mathbf{x})} + (1-\alpha_{t|s}) \frac{\pi_j}{p_{t,j}(\mathbf{x})} \mathbf{1}\right]. \end{aligned} \tag{36}$$

Averaging this expression under the weights $w_{t,j}$ yields

$$\begin{aligned} \boldsymbol{\rho}_{s|t}(\mathbf{x}, \mathbf{w}_t) &= \mathbf{p}_s \odot \left[\alpha_{t|s} \sum_{j=1}^{K} w_{t,j} \frac{\mathbf{e}_j}{p_{t,j}(\mathbf{x})} + (1-\alpha_{t|s}) \sum_{j=1}^{K} w_{t,j} \frac{\pi_j}{p_{t,j}(\mathbf{x})} \mathbf{1}\right] \\ &= \mathbf{p}_s \odot \left[\alpha_{t|s}\big(\mathbf{w}_t \oslash \mathbf{p}_t\big) + (1-\alpha_{t|s})\, \langle \mathbf{w}_t,\ \boldsymbol{\pi} \oslash \mathbf{p}_t\rangle\, \mathbf{1}\right], \end{aligned} \tag{37}$$

which proves Eq. (27) and Eq. (28).

4. We next derive the reverse bridge over relaxed states conditioned on $\mathbf{z}_t$. Since $\mathbf{w}_s$ is conditionally independent of $\mathbf{z}_t$ given $(\mathbf{z}_s, \mathbf{x})$, marginalizing over $\mathbf{z}_s$ gives

$$\begin{aligned} q(\mathbf{w}_s \mid \mathbf{z}_t, \mathbf{x}) &= \sum_{k=1}^{K} q(\mathbf{w}_s \mid \mathbf{z}_s = \mathbf{e}_k, \mathbf{z}_t, \mathbf{x})\, q(\mathbf{z}_s = \mathbf{e}_k \mid \mathbf{z}_t, \mathbf{x}) \\ &= \sum_{k=1}^{K} q(\mathbf{w}_s \mid \mathbf{z}_s = \mathbf{e}_k, \mathbf{x})\, q(\mathbf{z}_s = \mathbf{e}_k \mid \mathbf{z}_t, \mathbf{x}). \end{aligned} \tag{38}$$

The first factor is exactly the shifted Dirichlet bridge at time s:

$$q(\mathbf{w}_s \mid \mathbf{z}_s = \mathbf{e}_k, \mathbf{x}) = \mathrm{Dir}(\mathbf{w}_s; \eta_s \mathbf{p}_s + \mathbf{e}_k)\,.$$

The second factor is the discrete reverse posterior

$$q(\mathbf{z}_s = \mathbf{e}_k \mid \mathbf{z}_t, \mathbf{x}) = r_{s|t,k}(\mathbf{x}, \mathbf{z}_t).$$

Substituting these identities into Eq. (38) yields

$$q(\mathbf{w}_s \mid \mathbf{z}_t, \mathbf{x}) = \sum_{k=1}^{K} r_{s|t,k}(\mathbf{x}, \mathbf{z}_t)\ \mathrm{Dir}(\mathbf{w}_s; \eta_s \mathbf{p}_s + \mathbf{e}_k)\,,$$

which proves Eq. (29).

5. Finally, we lift this mixture from $\mathbf{z}_t$ to $\mathbf{w}_t$. Since $\mathbf{w}_s$ is also conditionally independent of $\mathbf{w}_t$ given $(\mathbf{z}_s, \mathbf{x})$, we obtain

$$\begin{aligned} q(\mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x}) &= \sum_{k=1}^{K} q(\mathbf{w}_s \mid \mathbf{z}_s = \mathbf{e}_k, \mathbf{w}_t, \mathbf{x})\, q(\mathbf{z}_s = \mathbf{e}_k \mid \mathbf{w}_t, \mathbf{x}) \\ &= \sum_{k=1}^{K} q(\mathbf{w}_s \mid \mathbf{z}_s = \mathbf{e}_k, \mathbf{x})\, q(\mathbf{z}_s = \mathbf{e}_k \mid \mathbf{w}_t, \mathbf{x}). \end{aligned} \tag{39}$$

The first factor is again $\mathrm{Dir}(\mathbf{w}_s; \eta_s \mathbf{p}_s + \mathbf{e}_k)$, while the second factor is now the lifted reverse posterior:

$$q(\mathbf{z}_s = \mathbf{e}_k \mid \mathbf{w}_t, \mathbf{x}) = \rho_{s|t,k}(\mathbf{x}, \mathbf{w}_t).$$

Therefore

$$q(\mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x}) = \sum_{k=1}^{K} \rho_{s|t,k}(\mathbf{x}, \mathbf{w}_t)\ \mathrm{Dir}(\mathbf{w}_s; \eta_s \mathbf{p}_s + \mathbf{e}_k)\,,$$

which proves Eq. (30).

□

Taken together, these identities show that the simplex variable sits inside the discrete diffusion process in a fully coherent way. The relaxed state has the correct Dirichlet marginal, the discrete state can be decoded from it exactly, and the reverse bridge remains available in closed form after replacing either $\mathbf{z}_t$ or $\mathbf{z}_s$ by their simplex-conditioned posteriors. This is the structural reason that the later training objectives and samplers can be written directly in terms of $\mathbf{w}_t$ without abandoning the original categorical process.

E Rao–Blackwellized Reverse-Bridge Objective

This section proves the closed form of the relaxed discrete bridge objective used in the main text. The auxiliary decoder sample $\widetilde{\mathbf{z}}_t$ is marginalized analytically; the independently sampled denoiser input $\mathbf{z}_t$ is not marginalized.

Proposition 2. *For $0 \le s < t \le 1$ with $t > 0$, the relaxed discrete bridge objective satisfies*

$$\bar{\mathcal{L}}_{z_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; s, t) = \langle \mathbf{w}_t,\ \log \hat{\mathbf{p}}_t - \log \mathbf{p}_t \rangle + \big\langle \boldsymbol{\rho}_{s|t}(\mathbf{x}, \mathbf{w}_t),\ \log \mathbf{p}_s - \log \hat{\mathbf{p}}_s \big\rangle\,. \tag{40}$$

Proof. We first compute the Rao–Blackwellized categorical term $\bar{\mathcal{L}}_{z_s|z_t,w_t}$. By definition,

$$\bar{\mathcal{L}}_{z_s|z_t,w_t} = \mathbb{E}_{q(\widetilde{\mathbf{z}}_t|\mathbf{w}_t)}[D_{\mathrm{KL}}[q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{x}) \,\|\, q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t, \hat{\mathbf{x}}_\theta)]]\,. \tag{41}$$

Using Eq. (26), we have

$$q(\widetilde{\mathbf{z}}_t = \mathbf{e}_j \mid \mathbf{w}_t) = w_{t,j}.$$

For fixed $\widetilde{\mathbf{z}}_t = \mathbf{e}_j$, both reverse conditionals are categorical:

$$q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t = \mathbf{e}_j, \mathbf{x}) = \mathrm{Cat}\big(\mathbf{z}_s; \mathbf{r}_{s|t}(\mathbf{x}, \mathbf{e}_j)\big)\,, \qquad q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t = \mathbf{e}_j, \hat{\mathbf{x}}_\theta) = \mathrm{Cat}\big(\mathbf{z}_s; \mathbf{r}_{s|t}(\hat{\mathbf{x}}_\theta, \mathbf{e}_j)\big)\,.$$

Expanding the expectation over $\widetilde{\mathbf{z}}_t$ and the KL divergence between categorical distributions gives

$$\bar{\mathcal{L}}_{z_s|z_t,w_t} = \sum_{j=1}^{K} w_{t,j} \sum_{k=1}^{K} r_{s|t,k}(\mathbf{x}, \mathbf{e}_j) \log \frac{r_{s|t,k}(\mathbf{x}, \mathbf{e}_j)}{r_{s|t,k}(\hat{\mathbf{x}}_\theta, \mathbf{e}_j)}. \tag{42}$$

Substituting the explicit form of the reverse posterior components, we obtain

$$\begin{aligned} \frac{r_{s|t,k}(\mathbf{x}, \mathbf{e}_j)}{r_{s|t,k}(\hat{\mathbf{x}}_\theta, \mathbf{e}_j)} &= \frac{\big[\alpha_{t|s}\delta_{j,k} + (1-\alpha_{t|s})\pi_j\big] p_{s,k}(\mathbf{x}) / p_{t,j}(\mathbf{x})}{\big[\alpha_{t|s}\delta_{j,k} + (1-\alpha_{t|s})\pi_j\big] p_{s,k}(\hat{\mathbf{x}}_\theta) / p_{t,j}(\hat{\mathbf{x}}_\theta)} \\ &= \frac{p_{s,k}(\mathbf{x})}{p_{s,k}(\hat{\mathbf{x}}_\theta)} \frac{p_{t,j}(\hat{\mathbf{x}}_\theta)}{p_{t,j}(\mathbf{x})}. \end{aligned} \tag{43}$$

Hence

$$\log \frac{r_{s|t,k}(\mathbf{x}, \mathbf{e}_j)}{r_{s|t,k}(\hat{\mathbf{x}}_\theta, \mathbf{e}_j)} = \log \frac{p_{t,j}(\hat{\mathbf{x}}_\theta)}{p_{t,j}(\mathbf{x})} + \log \frac{p_{s,k}(\mathbf{x})}{p_{s,k}(\hat{\mathbf{x}}_\theta)}. \tag{44}$$

Substituting Eq. (44) into Eq. (42), the first contribution is

$$\sum_{j=1}^{K} w_{t,j} \sum_{k=1}^{K} r_{s|t,k}(\mathbf{x}, \mathbf{e}_j) \log \frac{p_{t,j}(\hat{\mathbf{x}}_\theta)}{p_{t,j}(\mathbf{x})} = \sum_{j=1}^{K} w_{t,j} \log \frac{p_{t,j}(\hat{\mathbf{x}}_\theta)}{p_{t,j}(\mathbf{x})}, \tag{45}$$

because $r_{s|t}(\mathbf{x}, \mathbf{e}_j)$ is a probability vector. For the second contribution, exchanging the order of summation gives

$$\sum_{j=1}^{K} w_{t,j} \sum_{k=1}^{K} r_{s|t,k}(\mathbf{x}, \mathbf{e}_j) \log \frac{p_{s,k}(\mathbf{x})}{p_{s,k}(\hat{\mathbf{x}}_\theta)} = \sum_{k=1}^{K} \left(\sum_{j=1}^{K} w_{t,j} r_{s|t,k}(\mathbf{x}, \mathbf{e}_j) \right) \log \frac{p_{s,k}(\mathbf{x})}{p_{s,k}(\hat{\mathbf{x}}_\theta)}. \tag{46}$$

The coefficient in parentheses is exactly the k-th component of $q(\mathbf{z}_s \mid \mathbf{w}_t, \mathbf{x})$, namely

$$\sum_{j=1}^{K} w_{t,j} r_{s|t,k}(\mathbf{x}, \mathbf{e}_j) = \rho_{s|t,k}(\mathbf{x}, \mathbf{w}_t).$$

Therefore

$$\begin{aligned} \bar{\mathcal{L}}_{z_s|z_t,w_t} &= \sum_{j=1}^{K} w_{t,j} \log \frac{p_{t,j}(\hat{\mathbf{x}}_\theta)}{p_{t,j}(\mathbf{x})} + \sum_{k=1}^{K} \rho_{s|t,k}(\mathbf{x}, \mathbf{w}_t) \log \frac{p_{s,k}(\mathbf{x})}{p_{s,k}(\hat{\mathbf{x}}_\theta)} \\ &= \langle \mathbf{w}_t,\, \log \hat{\mathbf{p}}_t - \log \mathbf{p}_t \rangle + \big\langle \boldsymbol{\rho}_{s|t}(\mathbf{x}, \mathbf{w}_t),\, \log \mathbf{p}_s - \log \hat{\mathbf{p}}_s \big\rangle, \end{aligned} \tag{47}$$

which proves Eq. (40). □

Equation Eq. (40) depends on the auxiliary state only through the current-time average $\langle \mathbf{w}_t,\, \log \hat{\mathbf{p}}_t \rangle$ and the lifted reverse posterior $\boldsymbol{\rho}_{s|t}(\mathbf{x}, \mathbf{w}_t)$. This is the expression used in the main objective and in the continuous-time analysis.

F Continuous-Time Limit of the Main Objective

The main text shows that the relaxed discrete bridge objective $\bar{\mathcal{L}}_{z_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; s, t)$ admits a non-degenerate first-order continuous-time limit. This section gives the full proof of the first-order local limit stated in the main text.

F.1 Proof of the main continuous-time limit

Proposition 3. *Up to θ-independent additive terms, the relaxed discrete bridge objective satisfies*

$$\bar{\mathcal{L}}_{z_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; t-\Delta, t) = \Delta\, \ell_{\mathrm{ct}}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}, t) + o(\Delta), \tag{48}$$

where

$$\ell_{\mathrm{ct}}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}, t) = \lambda(t) \Big[\langle \mathbf{w}_t,\, \boldsymbol{\pi} \oslash \hat{\mathbf{p}}_t \rangle - \langle \mathbf{w}_t,\, \boldsymbol{\pi} \oslash \mathbf{p}_t \rangle \, \langle \mathbf{p}_t,\, \log \hat{\mathbf{p}}_t \rangle + \langle \boldsymbol{\pi} \odot (\mathbf{w}_t \oslash \mathbf{p}_t),\, \log \hat{\mathbf{p}}_t \rangle \Big]. \tag{49}$$

and $\lambda(t) \coloneqq -\frac{d}{dt} \log \alpha(t)$. *Consequently, the corresponding continuous-time objective is*

$$\mathcal{L}_{\mathrm{ct}} = \int_0^1 \mathbb{E}_{q(\mathbf{x})} \big[\mathbb{E}_{q(\mathbf{w}_t|\mathbf{x})} \big[\mathbb{E}_{q(\mathbf{z}_t|\mathbf{w}_t)} \big[\ell_{\mathrm{ct}}\big(\mathbf{w}_t, f_\theta(\mathbf{z}_t, t), \mathbf{x}, t\big) \big] \big] \big] \, \mathrm{d}t. \tag{50}$$

with $q(\mathbf{w}_t \mid \mathbf{x}) = \mathrm{Dir}(\mathbf{w}_t; \eta_t \mathbf{p}_t)$.

Proof of Proposition 3. Up to θ-independent additive terms, Eq. (40) can be written as

$$\bar{\mathcal{L}}_{z_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; s, t) \equiv \langle \mathbf{w}_t,\, \log \hat{\mathbf{p}}_t \rangle - \big\langle \boldsymbol{\rho}_{s|t}(\mathbf{x}, \mathbf{w}_t),\, \log \hat{\mathbf{p}}_s \big\rangle \,. \tag{51}$$

We now set $s = t - \Delta$ and let $\Delta \downarrow 0$.

First, by definition,

$$\lambda(t) = -\frac{d}{dt} \log \alpha(t), \tag{52}$$

so that

$$\alpha_{t|t-\Delta} = \frac{\alpha(t)}{\alpha(t-\Delta)} = 1 - \Delta\, \lambda(t) + o(\Delta). \tag{53}$$

Moreover,

$$\partial_t \mathbf{p}_t = \partial_t(\alpha(t)\mathbf{x} + (1-\alpha(t))\boldsymbol{\pi}) = -\lambda(t)\big(\mathbf{p}_t - \boldsymbol{\pi}\big), \tag{54}$$

hence

$$\mathbf{p}_{t-\Delta} = \mathbf{p}_t + \Delta\, \lambda(t)(\mathbf{p}_t - \boldsymbol{\pi}) + o(\Delta). \tag{55}$$

Next, we expand $\boldsymbol{\rho}_{t-\Delta|t}(\mathbf{x}, \mathbf{w}_t)$ using Eq. (28). Substituting Eq. (53) and Eq. (55) gives

$$\begin{aligned}
&\boldsymbol{\rho}_{t-\Delta|t}(\mathbf{x}, \mathbf{w}_t) \\
&= \mathbf{p}_{t-\Delta} \odot \big[\alpha_{t|t-\Delta}\big(\mathbf{w}_t \oslash \mathbf{p}_t\big) + \big(1 - \alpha_{t|t-\Delta}\big)\, \langle \mathbf{w}_t,\, \boldsymbol{\pi} \oslash \mathbf{p}_t \rangle\, \mathbf{1}\big] \\
&= \Big(\mathbf{p}_t + \Delta\, \lambda(t)(\mathbf{p}_t - \boldsymbol{\pi})\Big) \odot \Big[\mathbf{w}_t \oslash \mathbf{p}_t + \Delta\, \lambda(t)\Big(\langle \mathbf{w}_t,\, \boldsymbol{\pi} \oslash \mathbf{p}_t \rangle\, \mathbf{1} - \mathbf{w}_t \oslash \mathbf{p}_t\Big)\Big] + o(\Delta) \\
&= \mathbf{w}_t + \Delta\, \lambda(t)\, [\langle \mathbf{w}_t,\, \boldsymbol{\pi} \oslash \mathbf{p}_t \rangle\, \mathbf{p}_t - \boldsymbol{\pi} \odot (\mathbf{w}_t \oslash \mathbf{p}_t)] + o(\Delta).
\end{aligned} \tag{56}$$

We now expand $\log \hat{\mathbf{p}}_{t-\Delta}$. Since $\hat{\mathbf{x}}_\theta$ is held fixed in the local limit,

$$\hat{\mathbf{p}}_t = \alpha(t)\hat{\mathbf{x}}_\theta + (1-\alpha(t))\boldsymbol{\pi}, \tag{57}$$

and therefore

$$\partial_t \hat{\mathbf{p}}_t = -\lambda(t)\big(\hat{\mathbf{p}}_t - \boldsymbol{\pi}\big). \tag{58}$$

Dividing componentwise by $\hat{\mathbf{p}}_t$ yields

$$\partial_t \log \hat{\mathbf{p}}_t = -\lambda(t)\, (\mathbf{1} - \boldsymbol{\pi} \oslash \hat{\mathbf{p}}_t) \,. \tag{59}$$

Hence

$$\log \hat{\mathbf{p}}_{t-\Delta} = \log \hat{\mathbf{p}}_t - \Delta\, \partial_t \log \hat{\mathbf{p}}_t + o(\Delta) = \log \hat{\mathbf{p}}_t + \Delta\, \lambda(t)\, (\mathbf{1} - \boldsymbol{\pi} \oslash \hat{\mathbf{p}}_t) + o(\Delta). \tag{60}$$

For compactness, define

$$A_t \coloneqq \langle \mathbf{w}_t,\, \boldsymbol{\pi} \oslash \hat{\mathbf{p}}_t \rangle \,, \tag{61}$$

$$B_t \coloneqq \langle \mathbf{w}_t,\, \boldsymbol{\pi} \oslash \mathbf{p}_t \rangle\, \langle \mathbf{p}_t,\, \log \hat{\mathbf{p}}_t \rangle \,, \tag{62}$$

$$C_t \coloneqq \langle \boldsymbol{\pi} \odot (\mathbf{w}_t \oslash \mathbf{p}_t),\, \log \hat{\mathbf{p}}_t \rangle \,. \tag{63}$$

Substituting Eqs. (56) and (60) into Eq. (51) and collecting first-order terms gives

$$\bar{\mathcal{L}}_{z_{t-\Delta}|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; t-\Delta, t) \equiv \Delta\lambda(t)(A_t - B_t + C_t - 1) + o(\Delta). \tag{64}$$

The term $-\Delta\,\lambda(t)$ is independent of θ, so it can be discarded. Therefore,

$$\bar{\mathcal{L}}_{z_{t-\Delta}|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; t-\Delta, t) = \Delta\,\ell_{\mathrm{ct}}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}, t) + o(\Delta), \tag{65}$$

where, up to θ-independent additive terms,

$$\ell_{\mathrm{ct}}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}, t) \equiv \lambda(t)\left[\langle \mathbf{w}_t,\ \boldsymbol{\pi} \oslash \hat{\mathbf{p}}_t\rangle - \langle \mathbf{w}_t,\ \boldsymbol{\pi} \oslash \mathbf{p}_t\rangle\,\langle \mathbf{p}_t,\ \log \hat{\mathbf{p}}_t\rangle + \langle \boldsymbol{\pi} \odot (\mathbf{w}_t \oslash \mathbf{p}_t),\ \log \hat{\mathbf{p}}_t\rangle\right]. \tag{66}$$

This proves Eq. (17) and Eq. (18). Finally, integrating the density over time and taking expectation under $q(\mathbf{x})$ and $q(\mathbf{w}_t \mid \mathbf{x})$ yields Eq. (50). □

G Alternative Surrogate Objectives

In the main text, we optimize the relaxed discrete bridge objective $\bar{\mathcal{L}}_{z_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; s, t)$. The denoiser prediction $\hat{\mathbf{x}}_\theta = f_\theta(\mathbf{z}_t, t)$ uses an independently sampled categorical network input. In this appendix, $\widetilde{\mathbf{z}}_t$ denotes only the auxiliary categorical decode that is averaged inside the objective. This choice is best understood relative to a broader family of surrogate objectives induced by the same simplex relaxation. The natural starting point is the exact relaxed bridge

$$\mathcal{L}_{w_s|w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; s, t) \coloneqq D_{\mathrm{KL}}[q(\mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x}) \,\|\, q(\mathbf{w}_s \mid \mathbf{w}_t, \hat{\mathbf{x}}_\theta)]\,. \tag{67}$$

This objective is the most direct one, but is generally intractable because $q(\mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x})$ is a Dirichlet mixture. The tractable surrogates introduced below differ in two orthogonal ways: first, whether they match only the discrete reverse state $\mathbf{z}_s$ or the joint state $(\mathbf{z}_s, \mathbf{w}_s)$, and second, whether they condition directly on $\mathbf{w}_t$ or first decode $\widetilde{\mathbf{z}}_t \sim q(\widetilde{\mathbf{z}}_t \mid \mathbf{w}_t)$ and average.

G.1 A broader surrogate family

The simplest tractable surrogate matches only the lifted reverse posterior over $\mathbf{z}_s$:

$$\mathcal{L}_{z_s|w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; s, t) \coloneqq D_{\mathrm{KL}}[q(\mathbf{z}_s \mid \mathbf{w}_t, \mathbf{x}) \,\|\, q(\mathbf{z}_s \mid \mathbf{w}_t, \hat{\mathbf{x}}_\theta)]\,. \tag{68}$$

This objective preserves the full relaxed conditioning information in $\mathbf{w}_t$, but discards the relaxed target $\mathbf{w}_s$.

The objective used in the main text instead decodes $\widetilde{\mathbf{z}}_t$ from $\mathbf{w}_t$ and averages the standard discrete reverse KL:

$$\bar{\mathcal{L}}_{z_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; s, t) \coloneqq \mathbb{E}_{q(\widetilde{\mathbf{z}}_t|\mathbf{w}_t)}[D_{\mathrm{KL}}[q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{x}) \,\|\, q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t, \hat{\mathbf{x}}_\theta)]]\,. \tag{69}$$

Compared with Eq. (68), this objective is looser because $\mathbf{w}_t$ influences the reverse matching step only through the decoded categorical latent $\widetilde{\mathbf{z}}_t$. Its advantage is that it remains close to the standard discrete-diffusion objective and admits the non-degenerate continuous-time limit derived in the main text.

A richer alternative is to match the joint bridge over $(\mathbf{z}_s, \mathbf{w}_s)$ directly under the relaxed conditioning state:

$$\mathcal{L}_{z_s,w_s|w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; s, t) \coloneqq D_{\mathrm{KL}}[q(\mathbf{z}_s, \mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x}) \,\|\, q(\mathbf{z}_s, \mathbf{w}_s \mid \mathbf{w}_t, \hat{\mathbf{x}}_\theta)]\,. \tag{70}$$

This objective is richer than the discrete surrogates because it also matches the relaxed bridge at time s.

Finally, one may combine the decoded conditioning of Eq. (69) with the joint target of Eq. (70):

$$\bar{\mathcal{L}}_{z_s,w_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; s, t) \coloneqq \mathbb{E}_{q(\widetilde{\mathbf{z}}_t|\mathbf{w}_t)}[D_{\mathrm{KL}}[q(\mathbf{z}_s, \mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \mathbf{x}) \,\|\, q(\mathbf{z}_s, \mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \hat{\mathbf{x}}_\theta)]]\,. \tag{71}$$

This is the richest tractable surrogate in the family: it keeps the relaxed target $\mathbf{w}_s$ while also conditioning through the decoded latent $\widetilde{\mathbf{z}}_t$.

The loose joint objective admits a useful chain-rule decomposition. It shows that the main-text objective $\bar{\mathcal{L}}_{z_s|z_t,w_t}$ is precisely the categorical part of the loose joint bridge.

Proposition 4. *The tight and loose joint objectives admit the decompositions*

$$\mathcal{L}_{z_s,w_s|w_t}(\mathbf{w}_t,\hat{\mathbf{x}}_\theta,\mathbf{x};s,t) = \mathcal{L}_{z_s|w_t}(\mathbf{w}_t,\hat{\mathbf{x}}_\theta,\mathbf{x};s,t) + \bar{\mathcal{L}}_{w_s|z_s,w_t}(\mathbf{w}_t,\hat{\mathbf{x}}_\theta,\mathbf{x};s,t), \tag{72}$$

$$\bar{\mathcal{L}}_{z_s,w_s|z_t,w_t}(\mathbf{w}_t,\hat{\mathbf{x}}_\theta,\mathbf{x};s,t) = \bar{\mathcal{L}}_{z_s|z_t,w_t}(\mathbf{w}_t,\hat{\mathbf{x}}_\theta,\mathbf{x};s,t) + \bar{\mathcal{L}}_{w_s|z_s,w_t}(\mathbf{w}_t,\hat{\mathbf{x}}_\theta,\mathbf{x};s,t), \tag{73}$$

where

$$\bar{\mathcal{L}}_{w_s|z_s,w_t}(\mathbf{w}_t,\hat{\mathbf{x}}_\theta,\mathbf{x};s,t) \coloneqq \mathbb{E}_{q(\mathbf{z}_s|\mathbf{w}_t,\mathbf{x})}[D_{\mathrm{KL}}[q(\mathbf{w}_s \mid \mathbf{z}_s,\mathbf{x}) \,\|\, q(\mathbf{w}_s \mid \mathbf{z}_s,\hat{\mathbf{x}}_\theta)]]\,. \tag{74}$$

Proof. We first derive the tight decomposition Eq. (72).

By the joint graphical model, we have the conditional independences

$$\mathbf{w}_s \perp\!\!\!\perp \mathbf{w}_t \mid (\mathbf{z}_s,\mathbf{x}), \qquad q(\mathbf{z}_s,\mathbf{w}_s \mid \mathbf{w}_t,\mathbf{x}) = q(\mathbf{z}_s \mid \mathbf{w}_t,\mathbf{x})\, q(\mathbf{w}_s \mid \mathbf{z}_s,\mathbf{x}),$$

and similarly on the model side,

$$q(\mathbf{z}_s,\mathbf{w}_s \mid \mathbf{w}_t,\hat{\mathbf{x}}_\theta) = q(\mathbf{z}_s \mid \mathbf{w}_t,\hat{\mathbf{x}}_\theta)\, q(\mathbf{w}_s \mid \mathbf{z}_s,\hat{\mathbf{x}}_\theta).$$

Substituting these factorizations into $\mathcal{L}_{z_s,w_s|w_t}$ gives

$$\mathcal{L}_{z_s,w_s|w_t} = D_{\mathrm{KL}}[q(\mathbf{z}_s \mid \mathbf{w}_t,\mathbf{x})\, q(\mathbf{w}_s \mid \mathbf{z}_s,\mathbf{x}) \,\|\, q(\mathbf{z}_s \mid \mathbf{w}_t,\hat{\mathbf{x}}_\theta)\, q(\mathbf{w}_s \mid \mathbf{z}_s,\hat{\mathbf{x}}_\theta)]\,. \tag{75}$$

Applying the chain rule of KL yields

$$\begin{aligned}\mathcal{L}_{z_s,w_s|w_t} &= D_{\mathrm{KL}}[q(\mathbf{z}_s \mid \mathbf{w}_t,\mathbf{x}) \,\|\, q(\mathbf{z}_s \mid \mathbf{w}_t,\hat{\mathbf{x}}_\theta)] \\ &\quad + \mathbb{E}_{q(\mathbf{z}_s|\mathbf{w}_t,\mathbf{x})}[D_{\mathrm{KL}}[q(\mathbf{w}_s \mid \mathbf{z}_s,\mathbf{x}) \,\|\, q(\mathbf{w}_s \mid \mathbf{z}_s,\hat{\mathbf{x}}_\theta)]]\,.\end{aligned} \tag{76}$$

The first term is exactly $\mathcal{L}_{z_s|w_t}$, while the second term is $\bar{\mathcal{L}}_{w_s|z_s,w_t}$ by definition. This proves Eq. (72).

We next derive the loose decomposition Eq. (73).

The same graphical model implies the conditional independences

$$\mathbf{z}_s \perp\!\!\!\perp \mathbf{w}_t \mid (\widetilde{\mathbf{z}}_t,\mathbf{x}), \qquad \mathbf{w}_s \perp\!\!\!\perp (\widetilde{\mathbf{z}}_t,\mathbf{w}_t) \mid (\mathbf{z}_s,\mathbf{x}),$$

and therefore

$$q(\mathbf{z}_s,\mathbf{w}_s \mid \widetilde{\mathbf{z}}_t,\mathbf{w}_t,\mathbf{x}) = q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t,\mathbf{x})\, q(\mathbf{w}_s \mid \mathbf{z}_s,\mathbf{x}), \tag{77}$$

$$q(\mathbf{z}_s,\mathbf{w}_s \mid \widetilde{\mathbf{z}}_t,\mathbf{w}_t,\hat{\mathbf{x}}_\theta) = q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t,\hat{\mathbf{x}}_\theta)\, q(\mathbf{w}_s \mid \mathbf{z}_s,\hat{\mathbf{x}}_\theta). \tag{78}$$

Substituting Eq. (77) and Eq. (78) into $\bar{\mathcal{L}}_{z_s,w_s|z_t,w_t}$ gives

$$\bar{\mathcal{L}}_{z_s,w_s|z_t,w_t} = \mathbb{E}_{q(\widetilde{\mathbf{z}}_t|\mathbf{w}_t)}[D_{\mathrm{KL}}[q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t,\mathbf{x})\, q(\mathbf{w}_s \mid \mathbf{z}_s,\mathbf{x}) \,\|\, q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t,\hat{\mathbf{x}}_\theta)\, q(\mathbf{w}_s \mid \mathbf{z}_s,\hat{\mathbf{x}}_\theta)]]\,. \tag{79}$$

Applying the chain rule of KL inside the expectation yields

$$\begin{aligned}\bar{\mathcal{L}}_{z_s,w_s|z_t,w_t} &= \mathbb{E}_{q(\widetilde{\mathbf{z}}_t|\mathbf{w}_t)}[D_{\mathrm{KL}}[q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t,\mathbf{x}) \,\|\, q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t,\hat{\mathbf{x}}_\theta)]] \\ &\quad + \mathbb{E}_{q(\widetilde{\mathbf{z}}_t|\mathbf{w}_t)}\left[\mathbb{E}_{q(\mathbf{z}_s|\widetilde{\mathbf{z}}_t,\mathbf{x})}[D_{\mathrm{KL}}[q(\mathbf{w}_s \mid \mathbf{z}_s,\mathbf{x}) \,\|\, q(\mathbf{w}_s \mid \mathbf{z}_s,\hat{\mathbf{x}}_\theta)]]\right]\,.\end{aligned} \tag{80}$$

The first term is exactly $\bar{\mathcal{L}}_{z_s|z_t,w_t}$. For the second term, apply the law of total expectation:

$$\mathbb{E}_{q(\widetilde{\mathbf{z}}_t|\mathbf{w}_t)}\left[\mathbb{E}_{q(\mathbf{z}_s|\widetilde{\mathbf{z}}_t,\mathbf{x})}[[\cdot]]\right] = \mathbb{E}_{q(\mathbf{z}_s|\mathbf{w}_t,\mathbf{x})}[[\cdot]]\,.$$

Hence

$$\begin{aligned}\bar{\mathcal{L}}_{z_s,w_s|z_t,w_t} &= \bar{\mathcal{L}}_{z_s|z_t,w_t} \\ &\quad + \mathbb{E}_{q(\mathbf{z}_s|\mathbf{w}_t,\mathbf{x})}[D_{\mathrm{KL}}[q(\mathbf{w}_s \mid \mathbf{z}_s,\mathbf{x}) \,\|\, q(\mathbf{w}_s \mid \mathbf{z}_s,\hat{\mathbf{x}}_\theta)]]\,,\end{aligned} \tag{81}$$

which is exactly Eq. (73). □

The decomposition in Eq. (73) clarifies the role of the selected main-text objective. The relaxed discrete bridge keeps the categorical part of the joint bridge while discarding the additional simplex-matching term. This is exactly the simplification that later makes the continuous-time limit non-degenerate.

G.2 Auxiliary KL inequalities

We next record three standard KL inequalities that will be used to compare the surrogate objectives.

Lemma 2 (Data processing inequality for KL divergence)**.** *Let P and Q be two probability distributions on a space $\mathcal{X}$, and let $\mathcal{K}(\mathbf{y} \mid \mathbf{x})$ be a stochastic kernel from $\mathcal{X}$ to $\mathcal{Y}$. Define the pushforward distributions*

$$(P\mathcal{K})(\mathbf{y}) \coloneqq \mathbb{E}_{P(\mathbf{x})}[\mathcal{K}(\mathbf{y} \mid \mathbf{x})], \qquad (Q\mathcal{K})(\mathbf{y}) \coloneqq \mathbb{E}_{Q(\mathbf{x})}[\mathcal{K}(\mathbf{y} \mid \mathbf{x})]. \tag{82}$$

Then

$$D_{\mathrm{KL}}[P\mathcal{K} \,\|\, Q\mathcal{K}] \le D_{\mathrm{KL}}[P \,\|\, Q]. \tag{83}$$

Proof. This follows directly from Theorem 4.1 of Kullback & Leibler (1951) by taking T to be the stochastic kernel $\mathcal{K}$. □

A particularly important special case is marginalization.

Corollary 2 (Marginalization cannot increase KL)**.** *Let $P_{U,V}$ and $Q_{U,V}$ be two joint distributions, with marginals P_U and Q_U. Then*

$$D_{\mathrm{KL}}[P_U \,\|\, Q_U] \le D_{\mathrm{KL}}[P_{U,V} \,\|\, Q_{U,V}]. \tag{84}$$

Proof. Take $\mathcal{K}$ to be the projection kernel $(u, v) \mapsto u$ in Lemma 2. □

Lemma 3 (Joint convexity of KL divergence)**.** *Let $\lambda_i \ge 0$ with $\sum_{i=1}^m \lambda_i = 1$, and let P_i, Q_i be probability distributions on a common space. Then*

$$D_{\mathrm{KL}}\left[\sum_{i=1}^m \lambda_i P_i \,\middle\|\, \sum_{i=1}^m \lambda_i Q_i\right] \le \sum_{i=1}^m \lambda_i D_{\mathrm{KL}}[P_i \,\|\, Q_i]. \tag{85}$$

Proof. This follows from the joint convexity of relative entropy; see Theorem 2.7.2 in Cover & Thomas (2006). □

G.3 Relations among the surrogate objectives

The valid relations among the surrogate objectives follow from marginalization and joint convexity of KL divergence. Marginalizing either $\mathbf{z}_s$ or $\mathbf{w}_s$ from a joint bridge gives the corresponding categorical or simplex bound. In addition, averaging over the auxiliary decoded state $\widetilde{\mathbf{z}}_t \sim q(\widetilde{\mathbf{z}}_t \mid \mathbf{w}_t)$ gives the bounds from the direct categorical and direct joint objectives to their decoded counterparts.

To state the decoded simplex bound, define

$$\bar{\mathcal{L}}_{w_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; s, t) \coloneqq \mathbb{E}_{q(\widetilde{\mathbf{z}}_t|\mathbf{w}_t)}[D_{\mathrm{KL}}[q(\mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \mathbf{x}) \,\|\, q(\mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \hat{\mathbf{x}}_\theta)]]. \tag{86}$$

Proposition 5. *For all $\mathbf{w}_t \in \Delta^{K-1}$, $\hat{\mathbf{x}}_\theta \in \Delta^{K-1}$, and $\mathbf{x} \in \mathcal{V}$,*

$$\begin{aligned} \mathcal{L}_{z_s|w_t} &\le \mathcal{L}_{z_s,w_s|w_t}, \\ \mathcal{L}_{w_s|w_t} &\le \mathcal{L}_{z_s,w_s|w_t}, \\ \mathcal{L}_{z_s,w_s|w_t} &\le \bar{\mathcal{L}}_{z_s,w_s|z_t,w_t}. \end{aligned} \tag{87}$$

Moreover,

$$\begin{aligned} \mathcal{L}_{z_s|w_t} &\le \bar{\mathcal{L}}_{z_s|z_t,w_t} \le \bar{\mathcal{L}}_{z_s,w_s|z_t,w_t}, \\ \bar{\mathcal{L}}_{w_s|z_t,w_t} &\le \bar{\mathcal{L}}_{z_s,w_s|z_t,w_t}. \end{aligned} \tag{88}$$

Proof. We first prove Eq. (87).

1. The distributions $q(\mathbf{z}_s \mid \mathbf{w}_t, \mathbf{x})$ and $q(\mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x})$ are the corresponding marginals of $q(\mathbf{z}_s, \mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x})$. The same statement holds on the model side. Hence, by Corollary 2,

$$\begin{aligned} \mathcal{L}_{z_s|w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}) &\le D_{\mathrm{KL}}[q(\mathbf{z}_s, \mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x}) \,\|\, q(\mathbf{z}_s, \mathbf{w}_s \mid \mathbf{w}_t, \hat{\mathbf{x}}_\theta)] \\ &= \mathcal{L}_{z_s,w_s|w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}), \end{aligned} \tag{89}$$

$$\begin{aligned} \mathcal{L}_{w_s|w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}) &\le D_{\mathrm{KL}}[q(\mathbf{z}_s, \mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x}) \,\|\, q(\mathbf{z}_s, \mathbf{w}_s \mid \mathbf{w}_t, \hat{\mathbf{x}}_\theta)] \\ &= \mathcal{L}_{z_s,w_s|w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}). \end{aligned} \tag{90}$$

2. By marginalizing over $\widetilde{\mathbf{z}}_t$, we have

$$q(\mathbf{z}_s, \mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x}) = \mathbb{E}_{q(\widetilde{\mathbf{z}}_t|\mathbf{w}_t)}[q(\mathbf{z}_s, \mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \mathbf{x})], \tag{91}$$

$$q(\mathbf{z}_s, \mathbf{w}_s \mid \mathbf{w}_t, \hat{\mathbf{x}}_\theta) = \mathbb{E}_{q(\widetilde{\mathbf{z}}_t|\mathbf{w}_t)}[q(\mathbf{z}_s, \mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \hat{\mathbf{x}}_\theta)]. \tag{92}$$

The mixing law is shared by both sides because it is always $q(\widetilde{\mathbf{z}}_t \mid \mathbf{w}_t)$. Applying Lemma 3 yields

$$\begin{aligned}\mathcal{L}_{z_s,w_s|w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}) &= D_{\mathrm{KL}}[q(\mathbf{z}_s, \mathbf{w}_s \mid \mathbf{w}_t, \mathbf{x}) \,\|\, q(\mathbf{z}_s, \mathbf{w}_s \mid \mathbf{w}_t, \hat{\mathbf{x}}_\theta)] \\ &\le \mathbb{E}_{q(\widetilde{\mathbf{z}}_t|\mathbf{w}_t)}[D_{\mathrm{KL}}[q(\mathbf{z}_s, \mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \mathbf{x}) \,\|\, q(\mathbf{z}_s, \mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \hat{\mathbf{x}}_\theta)]] \\ &= \bar{\mathcal{L}}_{z_s,w_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}). \end{aligned} \tag{93}$$

Combining Eq. (89), Eq. (90), and Eq. (93) proves Eq. (87).

We now prove Eq. (88).

1. Using the conditional independence

$$\mathbf{z}_s \perp\!\!\!\perp \mathbf{w}_t \mid (\widetilde{\mathbf{z}}_t, \mathbf{x}),$$

and marginalizing over $\widetilde{\mathbf{z}}_t$, we have

$$q(\mathbf{z}_s \mid \mathbf{w}_t, \mathbf{x}) = \mathbb{E}_{q(\widetilde{\mathbf{z}}_t|\mathbf{w}_t)}[q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{x})], \tag{94}$$

$$q(\mathbf{z}_s \mid \mathbf{w}_t, \hat{\mathbf{x}}_\theta) = \mathbb{E}_{q(\widetilde{\mathbf{z}}_t|\mathbf{w}_t)}[q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t, \hat{\mathbf{x}}_\theta)]. \tag{95}$$

Applying Lemma 3 to these mixtures yields

$$\mathcal{L}_{z_s|w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}) \le \bar{\mathcal{L}}_{z_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}). \tag{96}$$

2. For each fixed $\widetilde{\mathbf{z}}_t$, $q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{x})$ and $q(\mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \mathbf{x})$ are the corresponding marginals of

$$q(\mathbf{z}_s, \mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \mathbf{x}),$$

where

$$q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \mathbf{x}) = q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{x}).$$

The same statements hold on the model side. Therefore, by Corollary 2,

$$\begin{aligned} &D_{\mathrm{KL}}[q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{x}) \,\|\, q(\mathbf{z}_s \mid \widetilde{\mathbf{z}}_t, \hat{\mathbf{x}}_\theta)] \\ &\quad \le D_{\mathrm{KL}}[q(\mathbf{z}_s, \mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \mathbf{x}) \,\|\, q(\mathbf{z}_s, \mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \hat{\mathbf{x}}_\theta)], \end{aligned} \tag{97}$$

$$\begin{aligned} &D_{\mathrm{KL}}[q(\mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \mathbf{x}) \,\|\, q(\mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \hat{\mathbf{x}}_\theta)] \\ &\quad \le D_{\mathrm{KL}}[q(\mathbf{z}_s, \mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \mathbf{x}) \,\|\, q(\mathbf{z}_s, \mathbf{w}_s \mid \widetilde{\mathbf{z}}_t, \mathbf{w}_t, \hat{\mathbf{x}}_\theta)]. \end{aligned} \tag{98}$$

Averaging both inequalities over $q(\widetilde{\mathbf{z}}_t \mid \mathbf{w}_t)$ gives

$$\bar{\mathcal{L}}_{z_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}) \le \bar{\mathcal{L}}_{z_s,w_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}), \tag{99}$$

$$\bar{\mathcal{L}}_{w_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}) \le \bar{\mathcal{L}}_{z_s,w_s|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}). \tag{100}$$

Combining Eq. (96), Eq. (99), and Eq. (100) proves Eq. (88). □

The joint surrogate contains an additional simplex-matching term. The following identity and proposition give its closed form.

Lemma 4. *For any $j, k \in \{1, \dots, K\}$,*

$$\mathbb{E}_{\mathrm{Dir}(\cdot;\eta_s\mathbf{p}_s+\mathbf{e}_k)}[\log w_j] = \psi(\eta_s p_{s,j}(\mathbf{x}) + \delta_{j,k}) - \psi(\eta_s + 1). \tag{101}$$

Proof. For a Dirichlet random vector with parameter $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)$, the standard identity is

$$\mathbb{E}_{\mathrm{Dir}(\cdot;\boldsymbol{\alpha})}[\log w_j] = \psi(\alpha_j) - \psi\left(\sum_{m=1}^{K} \alpha_m\right).$$

Applying this with

$$\boldsymbol{\alpha} = \eta_s \mathbf{p}_s + \mathbf{e}_k$$

gives

$$\alpha_j = \eta_s p_{s,j}(\mathbf{x}) + \delta_{j,k}, \qquad \sum_{m=1}^{K} \alpha_m = \eta_s \sum_{m=1}^{K} p_{s,m}(\mathbf{x}) + 1 = \eta_s + 1,$$

which proves Eq. (101). □

Proposition 6. *For $0 < s < t \le 1$, the simplex-matching term satisfies*

$$\bar{\mathcal{L}}_{w_s|z_s,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; s, t) = D_{\mathrm{KL}}[\mathrm{Dir}(\cdot; \eta_s \mathbf{p}_s) \,\|\, \mathrm{Dir}(\cdot; \eta_s \hat{\mathbf{p}}_s)] + \left\langle \boldsymbol{\rho}_{s|t}(\mathbf{x}, \mathbf{w}_t),\, \log \hat{\mathbf{p}}_s - \log \mathbf{p}_s + \mathbf{1} - \hat{\mathbf{p}}_s \oslash \mathbf{p}_s \right\rangle. \tag{102}$$

Proof. We compute the simplex term $\bar{\mathcal{L}}_{w_s|z_s,w_t}$. By definition,

$$\bar{\mathcal{L}}_{w_s|z_s,w_t} = \sum_{k=1}^{K} \rho_{s|t,k}(\mathbf{x}, \mathbf{w}_t)\, D_{\mathrm{KL}}[\mathrm{Dir}(\cdot; \eta_s \mathbf{p}_s + \mathbf{e}_k) \,\|\, \mathrm{Dir}(\cdot; \eta_s \hat{\mathbf{p}}_s + \mathbf{e}_k)]\,. \tag{103}$$

Fix $k \in \{1, \dots, K\}$. By Lemma 1,

$$\mathrm{Dir}(\mathbf{w}; \eta_s \mathbf{p}_s + \mathbf{e}_k) = \frac{w_k}{p_{s,k}(\mathbf{x})}\, \mathrm{Dir}(\mathbf{w}; \eta_s \mathbf{p}_s)\,, \tag{104}$$

$$\mathrm{Dir}(\mathbf{w}; \eta_s \hat{\mathbf{p}}_s + \mathbf{e}_k) = \frac{w_k}{p_{s,k}(\hat{\mathbf{x}}_\theta)}\, \mathrm{Dir}(\mathbf{w}; \eta_s \hat{\mathbf{p}}_s)\,. \tag{105}$$

Taking the logarithm of the ratio gives

$$\log \frac{\mathrm{Dir}(\mathbf{w}; \eta_s \mathbf{p}_s + \mathbf{e}_k)}{\mathrm{Dir}(\mathbf{w}; \eta_s \hat{\mathbf{p}}_s + \mathbf{e}_k)} = \log \frac{\mathrm{Dir}(\mathbf{w}; \eta_s \mathbf{p}_s)}{\mathrm{Dir}(\mathbf{w}; \eta_s \hat{\mathbf{p}}_s)} + \log p_{s,k}(\hat{\mathbf{x}}_\theta) - \log p_{s,k}(\mathbf{x}). \tag{106}$$

We now compare the expectation of the unshifted log-ratio under the shifted Dirichlet law with the KL between the unshifted Dirichlet distributions. Expanding the Dirichlet density gives

$$\log \frac{\mathrm{Dir}(\mathbf{w}; \eta_s \mathbf{p}_s)}{\mathrm{Dir}(\mathbf{w}; \eta_s \hat{\mathbf{p}}_s)} = \log \frac{B(\eta_s \hat{\mathbf{p}}_s)}{B(\eta_s \mathbf{p}_s)} + \eta_s \sum_{j=1}^{K} \big(p_{s,j}(\mathbf{x}) - p_{s,j}(\hat{\mathbf{x}}_\theta)\big) \log w_j. \tag{107}$$

Taking expectation under $\mathrm{Dir}(\cdot; \eta_s \mathbf{p}_s + \mathbf{e}_k)$ and applying Lemma 4 gives

$$\mathbb{E}_{\mathrm{Dir}(\cdot; \eta_s \mathbf{p}_s + \mathbf{e}_k)} \left[\log \frac{\mathrm{Dir}(\mathbf{w}; \eta_s \mathbf{p}_s)}{\mathrm{Dir}(\mathbf{w}; \eta_s \hat{\mathbf{p}}_s)} \right] = \log \frac{B(\eta_s \hat{\mathbf{p}}_s)}{B(\eta_s \mathbf{p}_s)} + \eta_s \sum_{j=1}^{K} \big(p_{s,j}(\mathbf{x}) - p_{s,j}(\hat{\mathbf{x}}_\theta)\big) \big(\psi(\eta_s p_{s,j}(\mathbf{x}) + \delta_{j,k}) - \psi(\eta_s + 1)\big). \tag{108}$$

On the other hand, the KL divergence between the unshifted Dirichlet distributions is

$$D_{\mathrm{KL}}[\mathrm{Dir}(\cdot; \eta_s \mathbf{p}_s) \,\|\, \mathrm{Dir}(\cdot; \eta_s \hat{\mathbf{p}}_s)] = \log \frac{B(\eta_s \hat{\mathbf{p}}_s)}{B(\eta_s \mathbf{p}_s)} + \eta_s \sum_{j=1}^{K} \big(p_{s,j}(\mathbf{x}) - p_{s,j}(\hat{\mathbf{x}}_\theta)\big) \big(\psi(\eta_s p_{s,j}(\mathbf{x})) - \psi(\eta_s)\big). \tag{109}$$

Subtracting Eq. (109) from Eq. (108), and using

$$\psi(a + 1) = \psi(a) + \frac{1}{a}, \qquad \psi(\eta_s + 1) = \psi(\eta_s) + \frac{1}{\eta_s},$$

yields

$$\mathbb{E}_{\mathrm{Dir}(\cdot; \eta_s \mathbf{p}_s + \mathbf{e}_k)} \left[\log \frac{\mathrm{Dir}(\mathbf{w}; \eta_s \mathbf{p}_s)}{\mathrm{Dir}(\mathbf{w}; \eta_s \hat{\mathbf{p}}_s)} \right] = D_{\mathrm{KL}}[\mathrm{Dir}(\cdot; \eta_s \mathbf{p}_s) \,\|\, \mathrm{Dir}(\cdot; \eta_s \hat{\mathbf{p}}_s)] + 1 - \frac{p_{s,k}(\hat{\mathbf{x}}_\theta)}{p_{s,k}(\mathbf{x})}. \tag{110}$$

Combining Eq. (106) with Eq. (110), we obtain

$$D_{\mathrm{KL}}[\mathrm{Dir}(\cdot; \eta_s \mathbf{p}_s + \mathbf{e}_k) \,\|\, \mathrm{Dir}(\cdot; \eta_s \hat{\mathbf{p}}_s + \mathbf{e}_k)] = D_{\mathrm{KL}}[\mathrm{Dir}(\cdot; \eta_s \mathbf{p}_s) \,\|\, \mathrm{Dir}(\cdot; \eta_s \hat{\mathbf{p}}_s)] + \log p_{s,k}(\hat{\mathbf{x}}_\theta) - \log p_{s,k}(\mathbf{x}) + 1 - \frac{p_{s,k}(\hat{\mathbf{x}}_\theta)}{p_{s,k}(\mathbf{x})}. \tag{111}$$

Substituting Eq. (111) into Eq. (103), and using $\sum_{k=1}^{K} \rho_{s|t,k}(\mathbf{x}, \mathbf{w}_t) = 1$, yields

$$\begin{aligned}\bar{\mathcal{L}}_{w_s|z_s,w_t} &= D_{\mathrm{KL}}[\mathrm{Dir}(\cdot;\eta_s\mathbf{p}_s) \,\|\, \mathrm{Dir}(\cdot;\eta_s\hat{\mathbf{p}}_s)] \\ &\quad + \sum_{k=1}^{K} \rho_{s|t,k}(\mathbf{x}, \mathbf{w}_t) \left(\log p_{s,k}(\hat{\mathbf{x}}_\theta) - \log p_{s,k}(\mathbf{x}) + 1 - \frac{p_{s,k}(\hat{\mathbf{x}}_\theta)}{p_{s,k}(\mathbf{x})} \right) \\ &= D_{\mathrm{KL}}[\mathrm{Dir}(\cdot;\eta_s\mathbf{p}_s) \,\|\, \mathrm{Dir}(\cdot;\eta_s\hat{\mathbf{p}}_s)] + \left\langle \boldsymbol{\rho}_{s|t}(\mathbf{x}, \mathbf{w}_t),\, \log \hat{\mathbf{p}}_s - \log \mathbf{p}_s + \mathbf{1} - \hat{\mathbf{p}}_s \oslash \mathbf{p}_s \right\rangle, \end{aligned} \tag{112}$$

which proves Eq. (102). □

G.4 Why the other surrogate objectives do not yield suitable continuous-time objectives

The relaxed discrete bridge is distinguished by its first-order scaling. We now show that the remaining surrogates behave differently in the local limit: the tight discrete objective vanishes at second order, whereas the joint objectives retain an $O(1)$ simplex-matching term.

Proposition 7. *Let $s = t - \Delta$ with $\Delta \downarrow 0$, and assume that η_t is continuous in t.*

1. *The tight discrete objective satisfies*

$$\mathcal{L}_{z_{t-\Delta}|w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; t-\Delta, t) = O(\Delta^2). \tag{113}$$

2. *Define*

$$\mathcal{S}_t(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}) \coloneqq D_{\mathrm{KL}}[\mathrm{Dir}(\cdot;\eta_t\mathbf{p}_t) \,\|\, \mathrm{Dir}(\cdot;\eta_t\hat{\mathbf{p}}_t)] + \langle \mathbf{w}_t,\, \log \hat{\mathbf{p}}_t - \log \mathbf{p}_t + \mathbf{1} - \hat{\mathbf{p}}_t \oslash \mathbf{p}_t \rangle. \tag{114}$$

   Then the simplex term satisfies

$$\bar{\mathcal{L}}_{w_{t-\Delta}|z_{t-\Delta},w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; t-\Delta, t) = \mathcal{S}_t(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}) + o(1). \tag{115}$$

3. *Consequently,*

$$\mathcal{L}_{z_{t-\Delta},w_{t-\Delta}|w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; t-\Delta, t) = \mathcal{S}_t(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}) + o(1), \tag{116}$$

$$\bar{\mathcal{L}}_{z_{t-\Delta},w_{t-\Delta}|z_t,w_t}(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}; t-\Delta, t) = \mathcal{S}_t(\mathbf{w}_t, \hat{\mathbf{x}}_\theta, \mathbf{x}) + o(1). \tag{117}$$

Proof. 1. We first analyze the tight discrete objective. Using Eq. (28) together with the same expansions Eq. (53) and Eq. (55) as above, we obtain

$$\boldsymbol{\rho}_{t-\Delta|t}(\mathbf{x}, \mathbf{w}_t) = \mathbf{w}_t + \Delta\, \mathbf{a}_t(\mathbf{x}, \mathbf{w}_t) + o(\Delta), \tag{118}$$

$$\boldsymbol{\rho}_{t-\Delta|t}(\hat{\mathbf{x}}_\theta, \mathbf{w}_t) = \mathbf{w}_t + \Delta\, \hat{\mathbf{a}}_t(\hat{\mathbf{x}}_\theta, \mathbf{w}_t) + o(\Delta), \tag{119}$$

where

$$\mathbf{a}_t(\mathbf{x}, \mathbf{w}_t) = \lambda(t) \left[\langle \mathbf{w}_t,\, \boldsymbol{\pi} \oslash \mathbf{p}_t \rangle\, \mathbf{p}_t - \boldsymbol{\pi} \odot (\mathbf{w}_t \oslash \mathbf{p}_t) \right], \tag{120}$$

$$\hat{\mathbf{a}}_t(\hat{\mathbf{x}}_\theta, \mathbf{w}_t) = \lambda(t) \left[\langle \mathbf{w}_t,\, \boldsymbol{\pi} \oslash \hat{\mathbf{p}}_t \rangle\, \hat{\mathbf{p}}_t - \boldsymbol{\pi} \odot (\mathbf{w}_t \oslash \hat{\mathbf{p}}_t) \right]. \tag{121}$$

Since both vectors in Eq. (118) and Eq. (119) are probability vectors, their first-order perturbations satisfy

$$\langle \mathbf{1},\, \mathbf{a}_t(\mathbf{x}, \mathbf{w}_t) \rangle = 0, \qquad \langle \mathbf{1},\, \hat{\mathbf{a}}_t(\hat{\mathbf{x}}_\theta, \mathbf{w}_t) \rangle = 0.$$

Therefore, expanding the categorical KL around the common base point $\mathbf{w}_t$ shows that the first-order term cancels:

$$\begin{aligned}\mathcal{L}_{z_{t-\Delta}|w_t} &= D_{\mathrm{KL}}[\mathrm{Cat}(\cdot;\mathbf{w}_t + \Delta\, \mathbf{a}_t + o(\Delta)) \,\|\, \mathrm{Cat}(\cdot;\mathbf{w}_t + \Delta\, \hat{\mathbf{a}}_t + o(\Delta))] \\ &= O(\Delta^2). \end{aligned} \tag{122}$$

This proves Eq. (113).

2. We next analyze the simplex term using its closed form Eq. (102). Since η_t is continuous,

$$\eta_{t-\Delta} = \eta_t + o(1).$$

Moreover, by Eq. (55) and its analogue for $\hat{\mathbf{p}}_{t-\Delta}$,

$$\mathbf{p}_{t-\Delta} = \mathbf{p}_t + O(\Delta), \qquad \hat{\mathbf{p}}_{t-\Delta} = \hat{\mathbf{p}}_t + O(\Delta).$$

Finally, Eq. (56) gives

$$\boldsymbol{\rho}_{t-\Delta|t}(\mathbf{x}, \mathbf{w}_t) = \mathbf{w}_t + O(\Delta).$$

Substituting these expansions into Eq. (102) yields

$$\begin{aligned}\bar{\mathcal{L}}_{w_{t-\Delta}|z_{t-\Delta},w_t} &= D_{\mathrm{KL}}[\mathrm{Dir}(\cdot;\eta_t\mathbf{p}_t) \,\|\, \mathrm{Dir}(\cdot;\eta_t\hat{\mathbf{p}}_t)] \\ &\quad + \langle \mathbf{w}_t,\, \log\hat{\mathbf{p}}_t - \log\mathbf{p}_t + \mathbf{1} - \hat{\mathbf{p}}_t \oslash \mathbf{p}_t\rangle + o(1), \end{aligned} \tag{123}$$

which is exactly Eq. (115).

3. The asymptotics of the two joint objectives now follow from the decompositions in Eqs. (72) and (73). For the tight joint objective,

$$\mathcal{L}_{z_{t-\Delta},w_{t-\Delta}|w_t} = \mathcal{L}_{z_{t-\Delta}|w_t} + \bar{\mathcal{L}}_{w_{t-\Delta}|z_{t-\Delta},w_t},$$

and combining Eq. (113) with Eq. (115) gives Eq. (116).

For the loose joint objective,

$$\bar{\mathcal{L}}_{z_{t-\Delta},w_{t-\Delta}|z_t,w_t} = \bar{\mathcal{L}}_{z_{t-\Delta}|z_t,w_t} + \bar{\mathcal{L}}_{w_{t-\Delta}|z_{t-\Delta},w_t}.$$

The first term is $o(1)$ by Proposition 3, while the second is given by Eq. (115). This yields Eq. (117).

□

The proposition makes the selection of the main-text objective precise. The tight discrete objective is too small in the local limit: after dividing by Δ, it vanishes. The joint objectives behave in the opposite way: they contain a generally nonzero $O(1)$ simplex-matching term, so they do not reduce to a finite first-order training density. The relaxed discrete bridge sits exactly between these two extremes, which is why it is the natural objective for the continuous-time formulation.

H Sampling Details

The sampler in Sec. 3.3 is obtained by replacing the unknown clean state in the reverse bridge with the denoiser prediction

$$\hat{\mathbf{x}}_\theta = f_\theta(\mathbf{z}_t, t) \in \Delta^{K-1}. \tag{124}$$

For adjacent times $s < t$, this gives the plug-in reverse hierarchy

$$q_\theta(\mathbf{z}_s \mid \mathbf{z}_t, \mathbf{w}_t) = \mathrm{Cat}\big(\mathbf{z}_s; \boldsymbol{\rho}_{s|t}(\hat{\mathbf{x}}_\theta, \mathbf{w}_t)\big)\,, \tag{125}$$

$$q_\theta(\mathbf{w}_s \mid \mathbf{z}_s, \mathbf{z}_t, \mathbf{w}_t) = \mathrm{Dir}(\mathbf{w}_s; \eta_s\hat{\mathbf{p}}_s + \mathbf{z}_s)\,. \tag{126}$$

The stochastic sampler follows these two conditionals ancestrally.

Deterministic categorical update. The categorical transition can be determinized by replacing the draw from Eq. (125) with the mode of the reverse categorical posterior:

$$k^\star = \arg\max_k \rho_{s|t,k}(\hat{\mathbf{x}}_\theta, \mathbf{w}_t)\,, \qquad \mathbf{z}_s \leftarrow \mathbf{e}_{k^\star}. \tag{127}$$

Deterministic simplex update. Conditioned on a categorical state $\mathbf{z}_s$, the simplex transition is

$$q_\theta(\mathbf{w}_s \mid \mathbf{z}_s, \mathbf{z}_t, \mathbf{w}_t) = \mathrm{Dir}(\mathbf{w}_s; \eta_s\hat{\mathbf{p}}_s + \mathbf{z}_s)\,.$$

Since $\hat{\mathbf{p}}_s$ and $\mathbf{z}_s$ both sum to one, its conditional mean is

$$\mathbb{E}\left[\mathbf{w}_s \mid \mathbf{z}_s, \mathbf{z}_t, \mathbf{w}_t\right] = \frac{\eta_s\hat{\mathbf{p}}_s + \mathbf{z}_s}{\eta_s + 1}. \tag{128}$$

The $\mathbf{w}$-deterministic update therefore replaces the Dirichlet draw with

$$\mathbf{w}_s \leftarrow \frac{\eta_s \hat{\mathbf{p}}_s + \mathbf{z}_s}{\eta_s + 1}. \tag{129}$$

Sampler variants. The categorical and simplex transitions admit stochastic and deterministic updates, giving four basic sampler variants. Let

$$k^\star = \arg\max_k \rho_{s|t,k}(\hat{\mathbf{x}}_\theta, \mathbf{w}_t)\,. \tag{130}$$

The four variants are

$$\text{Stochastic:} \quad \mathbf{z}_s \sim \mathrm{Cat}\big(\boldsymbol{\rho}_{s|t}(\hat{\mathbf{x}}_\theta, \mathbf{w}_t)\big)\,, \qquad \mathbf{w}_s \sim \mathrm{Dir}(\eta_s \hat{\mathbf{p}}_s + \mathbf{z}_s)\,, \tag{131}$$

$$\mathbf{w}\text{-deterministic:} \quad \mathbf{z}_s \sim \mathrm{Cat}\big(\boldsymbol{\rho}_{s|t}(\hat{\mathbf{x}}_\theta, \mathbf{w}_t)\big)\,, \qquad \mathbf{w}_s \leftarrow \frac{\eta_s \hat{\mathbf{p}}_s + \mathbf{z}_s}{\eta_s + 1}, \tag{132}$$

$$\mathbf{z}\text{-deterministic:} \quad \mathbf{z}_s \leftarrow \mathbf{e}_{k^\star}\,, \qquad \mathbf{w}_s \sim \mathrm{Dir}(\eta_s \hat{\mathbf{p}}_s + \mathbf{z}_s)\,, \tag{133}$$

$$\text{Deterministic:} \quad \mathbf{z}_s \leftarrow \mathbf{e}_{k^\star}\,, \qquad \mathbf{w}_s \leftarrow \frac{\eta_s \hat{\mathbf{p}}_s + \mathbf{z}_s}{\eta_s + 1}. \tag{134}$$

We additionally consider an *expectation* sampler, which marginalizes over the categorical transition when updating the simplex state. Since

$$\mathbb{E}\left[\mathbf{w}_s \mid \mathbf{z}_t, \mathbf{w}_t\right] = \frac{\eta_s \hat{\mathbf{p}}_s + \boldsymbol{\rho}_{s|t}(\hat{\mathbf{x}}_\theta, \mathbf{w}_t)}{\eta_s + 1}, \tag{135}$$

the expectation sampler uses

$$\mathbf{w}_s \leftarrow \frac{\eta_s \hat{\mathbf{p}}_s + \boldsymbol{\rho}_{s|t}(\hat{\mathbf{x}}_\theta, \mathbf{w}_t)}{\eta_s + 1}. \tag{136}$$

Unlike the deterministic sampler, which first selects the categorical mode and then takes the corresponding conditional Dirichlet mean, the expectation sampler averages over the categorical transition before updating $\mathbf{w}_s$.