TC-NMD: TRAJECTORY-CONDITIONED NON-MARKOVIAN DIFFUSION FOR DNA SEQUENCE DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Discrete diffusion models have emerged as a competitive framework for biological sequence design, achieving strong results on DNA enhancer and promoter generation benchmarks. Yet these models share a common structural assumption that the reverse process is Markovian, so each denoising step conditions only on the current noisy state. Moreover, standard reverse processes preserve earlier decisions, preventing decoded tokens or excluded candidates from being reconsidered. We propose TC-NMD, a general non-Markovian discrete diffusion framework for biological sequence design. TC-NMD augments the denoiser with a learned summary of previously visited states: a Gumbel-Softmax router scores the cached states according to their compatibility with the current state and combines them into a routed history context for predicting the clean sequence. During inference, intermediate predictions are independently re-corrupted under a biased corruption schedule, allowing previously decoded tokens to be remasked and previously excluded candidates to re-enter before the final prediction. We instantiate this framework in both simplex shortlisting diffusion and masked diffusion, yielding TC-SLM and TC-MDLM, respectively. Both variants consistently improve over their Markovian counterparts: on unconditional Fly Brain enhancer generation at five network function evaluations, TC-SLM reduces FBD from 19.57 to 4.79, while TC-MDLM reduces it from 12.76 to 7.37. We further observe improvements on conditional enhancer and promoter design while introducing only a lightweight router.

1 INTRODUCTION

The design of regulatory DNA elements such as promoters and enhancers has become a central application of generative modeling in genomics (Taskiran et al., 2023; Gosai et al., 2024; Avdeyev et al., 2023), underpinning gene therapy, synthetic biology, and cell-type-specific expression systems. Models must match the distribution of natural regulatory elements while supporting conditioning on signals such as cell type or expression level (Taskiran et al., 2023; Stärk et al., 2024). The setting differs from language and protein generation in ways that affect model design: regulatory elements are short ($L \approx 200$–1000 bp), the alphabet has only four symbols (versus $\sim 10^4$ for language and $K = 20$ for proteins), and function depends on local motifs and their arrangement (Gosai et al., 2024; Avdeyev et al., 2023).

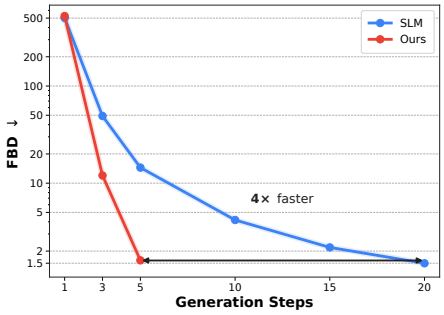

Figure 1: FBD versus number of sampling steps on conditional fly-brain enhancers. The non-Markovian sampler reaches the quality in roughly $4\times$ fewer steps

.

Discrete diffusion models for sequence design span D3PM and multinomial diffusion (Austin et al., 2021; Hoogeboom et al., 2021), continuous-time and score-based models (Campbell et al., 2022; Lou et al., 2024; Sahoo et al., 2024; Shi et al., 2024), and flow matching (Stärk et al., 2024; Gat et al., 2024; Davis et al., 2024; Campbell et al., 2024). For DNA, prior work has explored Dirichlet score matching (Avdeyev et al., 2023), latent diffusion (Li et al., 2024), reward-aligned fine-tuning (Wang

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.

et al., 2025a; Han et al., 2026), and nucleotide-substitution diffusion (Sarkar et al., 2025). The Shortlisting Model (SLM) (Song et al., 2025) uses simplex candidate-set centroids and achieves strong enhancer-design results.

Most of these methods use a Markovian reverse process that conditions each step only on the current noisy state. Recent work has begun to relax this assumption. CaDDi (Zhang et al., 2025) incorporates trajectory history by replacing the usual denoiser with a causal transformer that processes multiple noisy states as separate sequence blocks. This becomes computationally and memory intensive, its reported setup limits the context to five states, still giving the denoiser up to $5L$ input tokens. In contrast, TC-NMD retains the original bidirectional denoiser architecture and uses a lightweight router to compress all visited states into a single length-L input. This provides a substantially more efficient way to incorporate trajectory history. Relative to Markovian SLM, it adds only 0.14% forward MACs, 10% inference time, and 2% peak inference memory (Appendix E). The other work, DNDM (Chen et al., 2024) develops a discrete non-Markovian analogue of DDIM (Song et al., 2021) for accelerated sampling; both report improvements on language-generation tasks. Separately, another line of work addresses the irreversibility of standard discrete diffusion samplers. ReMDM (Wang et al., 2025b) and Ψ-samplers (Deschenaux et al., 2026) allow earlier decoding decisions to be revisited, while Wang et al. (2026) train the denoiser to correct its predictions. Our method draws on both ideas: an independent-corruption forward process makes the trajectory informative, while a learned router summarises the cached noisier states for each reverse step.

The standard diffusion reverse process makes intermediate decisions irreversible. In SLM the shortlist only shrinks as denoising proceeds, so a candidate excluded at an early step cannot return. We relax this restriction by independently constructing each intermediate shortlist from the current clean-sequence prediction using a biased schedular rate, and not forcing decisions from the previous shortlist to persist. An excluded candidate can therefore re-enter the shortlist and be reconsidered at a later step (Figure 3(c)). This flexibility is particularly important for regulatory DNA, where changing a single position can disrupt a functional motif. Independent corruption also makes the sampling history useful. Under a Markov process, the current state contains all the information provided by the already-visited noisier states, so $I(\mathbf{x}_0^{\mathbf{c}}; \mathbf{x}_t^{\mathbf{c}}, \mathbf{x}_{t+1}^{\mathbf{c}}) = I(\mathbf{x}_0^{\mathbf{c}}; \mathbf{x}_t^{\mathbf{c}})$ (Figure 3(b), top). When the states are corrupted independently, however, they become complementary noisy views of the clean sequence, and $I(\mathbf{x}_0^{\mathbf{c}}; \mathbf{x}_t^{\mathbf{c}}, \mathbf{x}_{t+1}^{\mathbf{c}}) > I(\mathbf{x}_0^{\mathbf{c}}; \mathbf{x}_t^{\mathbf{c}})$ (Figure 3(b), bottom). The denoiser can therefore benefit from the previously visited states, which our router combines into a learned history summary (Figure 3(a)). By coupling this informative trajectory with revisable sampling, TC-NMD reaches its quality floor in fewer sampling steps than the Markovian SLM baseline (Figure 1). The same principle can also be extended to MDLM, where independent re-corruption allows previously decoded tokens to be remasked and reconsidered.

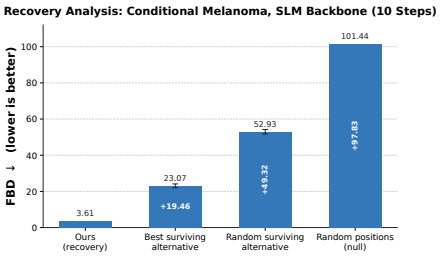

Figure 2: Value of recovered decisions on conditional melanoma enhancers using an SLM backbone with 10 sampling steps. A *recovery* occurs when an excluded candidate ($1 \to 0$) is later re-admitted ($0 \to 1$) and selected. Replacing recovered bases with candidates that survived their exclusion step substantially worsens FBD.

Having designed TC-NMD to reopen earlier shortlist decisions, we examine whether these recoveries improve generation. A *recovery* occurs when a candidate is excluded, later re-enters the shortlist, and is ultimately selected. For each recovery, the *best surviving alternative* replaces the recovered base with the highest-probability candidate that remained in the shortlist at the step when the recovered candidate was excluded. The *random surviving alternative* instead selects a random candidate that remained active at that step. Figure 2 shows that both replacements, as well as a random-position null control, produce substantially higher FBD, confirming that the recovered candidates represent useful corrections. The resulting framework admits a finite per-step ELBO, is trained with a stable gradient-rescaled cross-entropy objective on the $\mathbf{x}_0^{\mathbf{c}}$ prediction, and supports classifier-free guidance for conditional design. Our **Contributions** are as follows:

1. **Motivation.** We show that replacing Markovian state carry-over with independent corruption both allows early decisions to be reconsidered and makes the previously visited states informative for subsequent denoising.

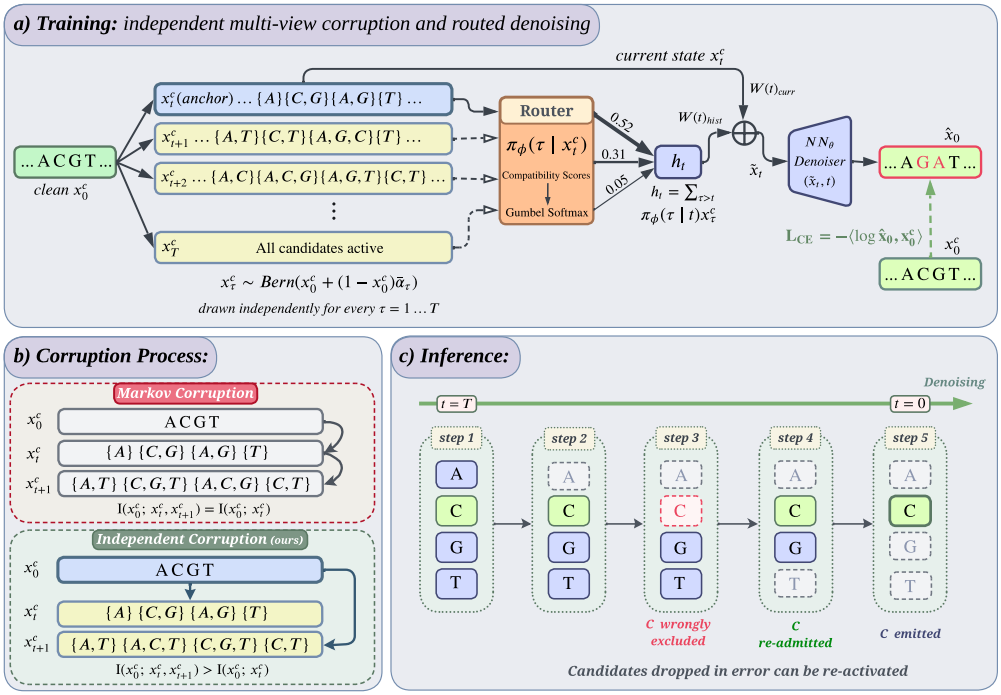

Figure 3: **Overview of TC-NMD. (a) Training.** The clean sequence $\mathbf{x}_0^\mathbf{c}$ is corrupted into T noisy views drawn *independently* at each noise level, $\mathbf{x}_\tau^\mathbf{c} \sim \mathrm{Bern}(\mathbf{x}_0^\mathbf{c} + (1 - \mathbf{x}_0^\mathbf{c})\bar{\alpha}_\tau)$. A Gumbel-Softmax router scores the views by their compatibility with the anchor state $\mathbf{x}_t^\mathbf{c}$ and mixes them into a history summary $\mathbf{h}_t = \sum_{\tau>t} \pi_\phi(\tau \mid t)\, \mathbf{x}_\tau^\mathbf{c}$ (arrow width $\propto \pi_\phi$); the denoiser predicts $\mathbf{x}_0^\mathbf{c}$ from the simplex-renormalised combination of $\mathbf{x}_t^\mathbf{c}$ and $\mathbf{h}_t$, trained with cross-entropy. **(b) Corruption process.** Under a Markov forward the trajectory is redundant, $I(\mathbf{x}_0^\mathbf{c}; \mathbf{x}_t^\mathbf{c}, \mathbf{x}_{t+1}^\mathbf{c}) = I(\mathbf{x}_0^\mathbf{c}; \mathbf{x}_t^\mathbf{c})$; under independent corruption the views are conditionally independent given $\mathbf{x}_0^\mathbf{c}$, so $I(\mathbf{x}_0^\mathbf{c}; \mathbf{x}_t^\mathbf{c}, \mathbf{x}_{t+1}^\mathbf{c}) > I(\mathbf{x}_0^\mathbf{c}; \mathbf{x}_t^\mathbf{c})$ and the trajectory is informative. **(c) Inference.** Intermediate steps allow earlier decisions to be reconsidered: an excluded candidate can re-enter the shortlist and later be selected. A final denoiser prediction produces the clean sequence without further re-corruption.

2. **General framework.** We introduce TC-NMD, a non-Markovian discrete diffusion framework that is not tied to a particular base diffusion model. It combines conditionally independent noisy views, a lightweight router that summarises the sampling history, and biased re-corruption that preserves the ability to revise intermediate predictions. We instantiate the framework in both simplex shortlisting diffusion and masked diffusion, yielding TC-SLM and TC-MDLM, respectively.
3. **Empirical validation.** Trained from scratch, both TC-SLM and TC-MDLM substantially outperform their Markovian counterparts on conditional and unconditional enhancer design, with the largest gains appearing in the few-step regime. At only five network function evaluations, our framework reduces FBD by up to 86% (from 14.65 to 2.02). TC-SLM also improves over SLM on conditional promoter design, demonstrating that the gains extend across regulatory sequence-design tasks.

2 Method

2.1 Background: Discrete Diffusion Backbones

Shortlisting Model. SLM (Song et al., 2025) is a discrete diffusion model over candidate sets. Each per-position state $\mathbf{x}_t^\mathbf{c} \in \{0, 1\}^K$ indicates which of the K tokens remain in the "shortlist" at

noise level t, with $\mathbf{x}_0^{\mathbf{c}}$ one-hot at the true token and $\mathbf{x}_T^{\mathbf{c}}$ containing all candidates. The forward process expands the shortlist according to a monotone schedule $\bar{\alpha}_t$ ($\bar{\alpha}_0 = 0$, $\bar{\alpha}_T = 1$), giving the marginal

$$q(\mathbf{x}_t^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}}) = \mathrm{Bern}(\mathbf{x}_0^{\mathbf{c}} + (\mathbf{1} - \mathbf{x}_0^{\mathbf{c}})\bar{\alpha}_t)\,, \tag{1}$$

sampled independently at each position, with the true token preserved at every t. The standard reverse process substitutes the network's $\mathbf{x}_0^{\mathbf{c}}$ prediction into the Bernoulli posterior and restricts the prediction to the current shortlist. Consequently, the shortlist can only shrink: once a candidate is excluded, it cannot return. Training uses a gradient-rescaled cross-entropy objective on the $\mathbf{x}_0^{\mathbf{c}}$ prediction. Appendix A provides the reverse parameterisation, training objective, and full ELBO derivation.

Masked Diffusion Language Model. MDLM (Sahoo et al., 2024) instead corrupts a sequence by independently replacing tokens with a mask symbol according to a monotone masking schedule. Its standard reverse process predicts tokens only at currently masked positions and preserves tokens that have already been decoded.

2.2 Overview of TC-NMD

The central design choice in TC-NMD is to give the reverse process access to multiple noisy views of the clean data at each step (Figure 3). We combine three mechanisms. First, a non-Markovian forward process samples the state at each timestep independently conditioned on $\mathbf{x}_0^{\mathbf{c}}$. This produces complementary noisy views of the clean sequence. Second, at each reverse step, a learned router combines the previously visited states into a history summary and supplies it to the denoiser together with $\mathbf{x}_t^{\mathbf{c}}$. Third, at inference, we independently re-corrupt each intermediate clean-sequence prediction using a biased corruption rate, allowing earlier decisions to be reconsidered. During training, the corrupted state at each timestep follows the same marginal distribution as in the original diffusion process. However, the states across timesteps are independent given $\mathbf{x}_0^{\mathbf{c}}$. Therefore, the previously visited states contain information beyond the current state and can guide the reverse process.

2.3 Non-Markovian Forward

We formalize TC-NMD using SLM notation. The MDLM instantiation follows the same construction, with token masking as the corruption operator. This section defines the independent-corruption forward process (Section 2.3.1), explains its joint structure (Section 2.3.2), and presents the reverse kernel and per-step objective (Section 2.3.3).

2.3.1 Definition

Given $\mathbf{x}_0^{\mathbf{c}}$, each timestep is sampled independently from the per-step marginal:

$$q(\mathbf{x}_{1:T}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}}) = \prod_{\tau=1}^{T} q(\mathbf{x}_\tau^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}}), \qquad q(\mathbf{x}_\tau^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}}) = \mathrm{Bern}(\mathbf{x}_0^{\mathbf{c}} + (\mathbf{1} - \mathbf{x}_0^{\mathbf{c}})\,\bar{\alpha}_\tau)\,. \tag{2}$$

The true token is preserved deterministically in every $\mathbf{x}_\tau^{\mathbf{c}}$; each non-true candidate is included independently with probability $\bar{\alpha}_\tau$, sampled afresh at every timestep.

2.3.2 Joint Structure

Equation 2 preserves the marginal $q(\mathbf{x}_t^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}})$ of standard SLM, so the distribution at each individual noise level remains unchanged. The joint distribution over the trajectory is different. Standard SLM produces a nested Markov trajectory, and the noisier states contain no additional information about $\mathbf{x}_0^{\mathbf{c}}$ once $\mathbf{x}_t^{\mathbf{c}}$ is known:

$$I(\mathbf{x}_0^{\mathbf{c}}; \mathbf{x}_{t:T}^{\mathbf{c}}) = I(\mathbf{x}_0^{\mathbf{c}}; \mathbf{x}_t^{\mathbf{c}}). \tag{3}$$

Our construction samples each state independently from $\mathbf{x}_0^{\mathbf{c}}$, making the previously visited states separate noisy observations of the clean sequence. Therefore,

$$I(\mathbf{x}_0^{\mathbf{c}}; \mathbf{x}_{t:T}^{\mathbf{c}}) = I(\mathbf{x}_0^{\mathbf{c}}; \mathbf{x}_t^{\mathbf{c}}) + I(\mathbf{x}_0^{\mathbf{c}}; \mathbf{x}_{t+1:T}^{\mathbf{c}} \mid \mathbf{x}_t^{\mathbf{c}}) \geq I(\mathbf{x}_0^{\mathbf{c}}; \mathbf{x}_t^{\mathbf{c}}), \tag{4}$$

with strict inequality whenever the history contains information not available in the current state.

2.3.3 Reverse Kernel and Per-Step Objective

By conditional independence, $\mathbf{x}^{\mathbf{c}}_{t-1} \perp \mathbf{x}^{\mathbf{c}}_t \mid \mathbf{x}^{\mathbf{c}}_0$. Therefore, the true posterior under the independent-corruption forward is

$$q(\mathbf{x}^{\mathbf{c}}_{t-1} \mid \mathbf{x}^{\mathbf{c}}_t, \mathbf{x}^{\mathbf{c}}_0) = q(\mathbf{x}^{\mathbf{c}}_{t-1} \mid \mathbf{x}^{\mathbf{c}}_0) = \mathrm{Bern}(\mathbf{x}^{\mathbf{c}}_0 + (\mathbf{1} - \mathbf{x}^{\mathbf{c}}_0)\bar{\alpha}_{t-1})\,. \tag{5}$$

We define the dataset-specific biased corruption rate as

$$r^{(d)}_{t-1} = \mathrm{clip}_{[0,1)}\left(\bar{\alpha}_{t-1} + b_d\right), \tag{6}$$

The reverse kernel is then:

$$p_{\theta,b}(\mathbf{x}^{\mathbf{c}}_{t-1} \mid \mathbf{x}^{\mathbf{c}}_t, \mathbf{h}_t) = \mathrm{Bern}\Big(\widehat{\mathbf{x}}^{\,\theta}_0 + (\mathbf{1} - \widehat{\mathbf{x}}^{\,\theta}_0) r^{(d)}_{t-1}\Big)\,, \tag{7}$$

where

$$\widehat{\mathbf{x}}^{\,\theta}_0 = \mathrm{NN}_\theta(\mathbf{x}^{\mathbf{c}}_t, \mathbf{h}_t, t). \tag{8}$$

Here, NN_θ denotes the denoising network, and $\mathbf{h}_t$ denotes the summary of the previously visited states. Section 2.4 defines the history summary and the exact input given to the denoiser. For $b_d = 0$, Equation 7 recovers the true posterior when $\widehat{\mathbf{x}}^{\,\theta}_0 = \mathbf{x}^{\mathbf{c}}_0$. A positive bias deliberately increases the expected corruption of the intermediate state, preserving more uncertainty and allowing earlier decisions to be reconsidered.

The corresponding per-step ELBO term is

$$\mathcal{L}_{t-1} = \mathbb{E}_{q(\mathbf{x}^{\mathbf{c}}_{t:T} \mid \mathbf{x}^{\mathbf{c}}_0)}\left[D_{\mathrm{KL}}\left(q(\mathbf{x}^{\mathbf{c}}_{t-1} \mid \mathbf{x}^{\mathbf{c}}_0) \,\|\, p_{\theta,b}(\mathbf{x}^{\mathbf{c}}_{t-1} \mid \mathbf{x}^{\mathbf{c}}_t, \mathbf{h}_t)\right)\right]. \tag{9}$$

Although the biased kernel does not generally equal the true posterior, it defines a valid reverse model. Since $\widehat{\mathbf{x}}^{\,\theta}_0 \in (0,1)^K$ and $r^{(d)}_{t-1} \in [0,1)$, its Bernoulli parameters lie strictly between zero and one. The reverse kernel therefore covers the support of the forward posterior, making every per-step KL finite. Appendix B provides the complete bound and proof.

Equation 7 is used for all intermediate transitions without intersecting the sampled state with $\mathbf{x}^{\mathbf{c}}_t$. At the final step, the denoiser directly predicts the clean sequence without further corruption.

Biased scheduler. At inference, we use $r^{(d)}_{t-1} = \mathrm{clip}_{[0,1)}(\bar{\alpha}_{t-1} + b_d)$, where the bias b_d is selected for each dataset and backbone. The higher background rate increases re-corruption at intermediate steps, giving the model more opportunities to reconsider and correct earlier errors as the sequence becomes more resolved. We ablate the bias in Section 3.2. The reverse kernel has a finite per-step KL; cross-entropy minimizes an upper bound on this KL, bringing it toward the bias-dependent floor (Appendix B).

2.4 Reverse Process with Soft History

At each reverse step, the denoiser receives the current state $\mathbf{x}^{\mathbf{c}}_t$, the diffusion timestep t, and a history summary $\mathbf{h}_t$ constructed from the previously visited noisier states. The history summary is a soft convex combination of these states under router-supplied weights. Conditioning the denoiser on an auxiliary signal in addition to the current state is reminiscent of self-conditioning in Markov diffusion (Chen et al., 2023), which feeds the previous step's clean-data prediction $\widehat{\mathbf{x}}^{\,\theta}_0$ back into the denoiser. Our mechanism differs in two respects: the auxiliary signal is a router-weighted mixture of multiple past *views* rather than a single previous prediction, and the current state and the summary are combined with noise-dependent weights (defined below) rather than concatenated at fixed strength.

The reverse process runs for $t = T, T-1, \ldots, 1$, and the denoiser models $p_\theta(\mathbf{x}^{\mathbf{c}}_{t-1} \mid \mathbf{x}^{\mathbf{c}}_t, \mathbf{h}_t)$; the history at step t aggregates the noisier states already generated above it, $\{\mathbf{x}^{\mathbf{c}}_\tau\}_{\tau=t+1}^{T}$. This set is empty at the first reverse step ($t = T$, generating $\mathbf{x}^{\mathbf{c}}_{T-1}$ from the prior $\mathbf{x}^{\mathbf{c}}_T$), where the model falls back to a no-history denoiser $p_\theta(\mathbf{x}^{\mathbf{c}}_{T-1} \mid \mathbf{x}^{\mathbf{c}}_T)$; it grows by one state at each subsequent step. At training time the anchor t is drawn uniformly from $\{1, \ldots, T\}$, so every reverse transition the sampler will take, including the boundary $t = T$, is covered.

Router. Given $\mathbf{x}_t^{\mathbf{c}}$ and the already-generated noisier states $\{\mathbf{x}_\tau^{\mathbf{c}}\}_{\tau=t+1}^{T}$, the router π_ϕ outputs a distribution over their indices $\tau \in \{t+1, \ldots, T\}$:

$$\pi_\phi(\tau \mid t, \mathbf{x}_t^{\mathbf{c}}, \{\mathbf{x}_\tau^{\mathbf{c}}\}) \in \Delta^{T-t}. \tag{10}$$

We use a Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017) relaxation during training, which makes the discrete routing decision differentiable, and standard softmax at inference. The router is a lightweight two-layer module that scores each candidate τ by its compatibility with $\mathbf{x}_t^{\mathbf{c}}$; full architectural details are given in Appendix D.1.

History summary. The history summary is the router-weighted convex combination of the already-generated noisier states:

$$\mathbf{h}_t = \sum_{\tau=t+1}^{T} \pi_\phi(\tau \mid t)\, \mathbf{x}_\tau^{\mathbf{c}}, \tag{11}$$

with the convention that $\mathbf{h}_T$ is undefined and the $t = T$ step uses the no-history denoiser. Each $\mathbf{x}_\tau^{\mathbf{c}}$ lies on the per-position simplex, so $\mathbf{h}_t$ is itself a simplex point. Its entries lie strictly inside the simplex rather than at vertices, so it is no longer a candidate-set centroid but a soft summary that mixes information across noise levels.

Denoiser input. The denoiser receives a simplex-renormalized combination of the current state and the history summary, with noise-dependent weights that anneal how much the history contributes:

$$\tilde{\mathbf{x}^{\mathbf{c}}}_t = \frac{w_{\text{cur}}(t)\, \mathbf{x}_t^{\mathbf{c}} + w_{\text{hist}}(t)\, \mathbf{h}_t}{\|w_{\text{cur}}(t)\, \mathbf{x}_t^{\mathbf{c}} + w_{\text{hist}}(t)\, \mathbf{h}_t\|_1}, \qquad \widehat{\mathbf{x}}_0^{\,\theta} = \text{NN}_\theta(\tilde{\mathbf{x}^{\mathbf{c}}}_t, t), \tag{12}$$

where $w_{\text{cur}}(t) = 1 + c_t$ and $w_{\text{hist}}(t) = 1 - c_t$, and $c_t \in [0, 1]$ is the per-step corruption level of the noise schedule ($c_t \to 1$ at high noise, $c_t \to 0$ near the clean end). The current state is therefore up-weighted (up to $2\times$) and the history down-weighted at high noise, where individual past views are least reliable, and the two approach equal weight as denoising proceeds and the history becomes more informative. Given the network's $\mathbf{x}_0^{\mathbf{c}}$-prediction $\widehat{\mathbf{x}}_0^{\,\theta}$, the reverse step samples from the Bernoulli kernel of Equation 7.

2.5 Training Objective

TC-NMD is trained on the history-conditioned $\mathbf{x}_0^{\mathbf{c}}$ prediction using cross-entropy, with gradient rescaling during optimisation:

$$\mathcal{L} = -\mathbb{E}_{t,\, \mathbf{x}_0^{\mathbf{c}},\, \mathbf{x}_t^{\mathbf{c}},\, \mathbf{h}_t} \left[\left\langle \log \text{NN}_\theta(\tilde{\mathbf{x}^{\mathbf{c}}}_t, t, \pi_\phi), \mathbf{x}_0^{\mathbf{c}} \right\rangle \right]. \tag{13}$$

The router is trained end-to-end through this objective, with the Gumbel-Softmax relaxation supplying gradients to its scores. During training, the denoiser predicts probabilities over all K candidates.

This cross-entropy is a surrogate for the KL of the biased reverse kernel. Appendix B shows that the per-step KL is finite (Proposition 2) and bounds its excess above the bias-induced KL floor using cross-entropy (Proposition 3). You can find the class-conditional generation in Appendix C. We summarise training and generation as pseudocode in Algorithms 1 and 2 (Appendix D).

3 Experiments

We evaluate TC-NMD on the DNA sequence design suite of Dirichlet flow matching (Stärk et al., 2024), later adopted by SLM (Song et al., 2025). The suite includes conditional promoter design and cell-type enhancer design. We inherit its splits, preprocessing, evaluators, and metrics; hyperparameters are selected on validation and evaluated on test. We also ablate the inference-time bias that controls re-corruption.

Tasks and Metrics We evaluate two tasks. **Promoter design** (conditional): generate a 1024 bp human promoter (Avdeyev et al., 2023) matching a target CAGE transcription profile, scored by SP-MSE against the Sei predictor (Chen et al., 2022) (lower is better). **Enhancer design**: generate 500 bp cell-type enhancers for human melanoma (Kalender Atak et al., 2021) and fly brain (Janssens et al., 2021), evaluated both unconditionally and with cell-type CFG, and scored by Fréchet Biological Distance (FBD $\downarrow$) in a pretrained classifier's feature space. Dataset sources, train/validation/test split sizes, and metric definitions follow Stärk et al. (2024) and are given in Appendix F.

Table 1: Enhancer sequence design on Human Melanoma (MEL) and Fly Brain (FB). Generation quality is measured by Fréchet Biological Distance (FBD ↓). Rows for SLM, TC-SLM, MDLM, and TC-MDLM are produced by us at 5 network function evaluations (NFE); all other rows are quoted from Song et al. (2025) at their reported operating points (100 NFE unless noted). SLM and MDLM are our reproductions of the corresponding Markov models. CFG denotes classifier-free guidance.

	Human Melanoma		**Fly Brain**	
Model	**FBD ↓**	**NFE**	**FBD ↓**	**NFE**
Random	619.0 ± 0.8	100	832.4 ± 0.3	100
Autoregressive	35.4 ± 0.5	100	25.7 ± 1.0	100
Dirichlet FM	12.38 ± 0.16	5	5.38 ± 0.09	100
BFN	76.64 ± 1.76	5	10.74 ± 0.97	100
Fisher-Flow	48.7 ± 4.6	5	5.9 ± 0.5	5
MDLM	15.92	5	12.76	5
TC-MDLM (ours)	**10.15**	**5**	**7.37**	**5**
SLM	10.11 ± 1.08	5	19.57 ± 0.52	5
TC-SLM (ours)	$\mathbf{5.76 \pm 0.64}$	**5**	$\mathbf{4.79 \pm 0.13}$	**5**
BFN + CFG	62.41 ± 1.07	5	2.3 ± 0.2	100
Dirichlet FM + CFG	8.47 ± 0.23	5	7.54 ± 0.04	5
MDLM + CFG	14.34	5	9.79	5
TC-MDLM + CFG (ours)	**6.48**	**5**	**6.07**	**5**
SLM + CFG	9.95 ± 0.31	5	14.65 ± 0.16	5
TC-SLM + CFG (ours)	$\mathbf{4.82 \pm 0.12}$	**5**	$\mathbf{2.02 \pm 0.63}$	**5**

3.1 Main Results

Baselines We compare against the baselines reported by Song et al. (2025): an autoregressive language model, Bayesian-flow and flow-matching models (BFN, Dirichlet flow matching (Stärk et al., 2024), Fisher-Flow), MDLM, and SLM. We implement TC-NMD on both SLM and MDLM backbones and compare each with our reproduction of its Markovian baseline. Full attribution is given in Appendix G.

Implementation details. We train each model for 800 epochs on four H200 GPUs and report results with 5 and 10 sampling steps. Architecture, training, sampling, and hyperparameter selection are detailed in Appendix D.1.

Table 2: Conditional human promoter generation measured by Sei-predictor MSE (↓). Prior non-autoregressive baselines use 100 NFE and the autoregressive baseline uses 1000 NFE. SLM and TC-SLM use 5 NFE and are reported as mean ± s.d. over three seeds. The 100-NFE results are from Song et al. (2025).

Model	**MSE ↓**	**NFE**
Autoregressive	0.034 ± 0.001	1000
Bit Diffusion (bit enc.)	0.041	100
Bit Diffusion (one-hot)	0.040	100
D3PM-uniform	0.038	100
UDLM	0.030 ± 0.001	100
MDLM	0.028 ± 0.001	100
Dirichlet FM	0.034 ± 0.004	100
Fisher-Flow	0.029 ± 0.001	100
BFN	0.0405 ± 0.0003	100
DDSM ($400/4\times$)	0.0339 ± 0.0006	696
DDSM	0.0536 ± 0.0008	5
Fisher-Flow	0.0553 ± 0.0003	5
SLM	0.0303 ± 0.0007	5
TC-SLM (ours)	$\mathbf{0.0267 \pm 0.0001}$	**5**

Table 1 reports enhancer design on Human Melanoma (MEL) and Fly Brain (FB). At 5 NFE, TC-NMD improves every matched comparison across both backbones, both datasets, and both guidance settings. The SLM-based model achieves lower mean FBD than every other 5-NFE method in all four enhancer settings. It also outperforms several baselines evaluated at 100 NFE, showing that these gains do not require a long sampling trajectory. To verify that our implementation of the baseline is not under-tuned, we report its results at 100 NFE in Appendix 10.

The largest gains are on fly brain. Without guidance, TC-NMD reduces SLM's FBD from 19.57 to 4.79 ($4.1\times$ lower); with CFG, it reduces FBD from 14.65 to 2.02 ($7.3\times$ lower). On melanoma,

Table 3: Ablations on the Melanoma enhancer task using the best checkpoint for each configuration. FBD (↓) is reported for conditional (Cond.) and unconditional (Uncond.) generation with 5 and 10 generation steps.

Configuration	SLM-based				MDLM-based			
	Conditional		Unconditional		Conditional		Unconditional	
	5 steps	10 steps	5 steps	10 steps	5 steps	10 steps	5 steps	10 steps
Non-Markovian (ours)	**4.71**	**3.52**	**5.31**	**3.79**	**6.48**	**3.21**	10.15	**6.14**
Reverse-process ablations								
No history – single view (Equivalent to the baseline)	9.61	4.09	9.10	5.53	14.34	7.36	15.92	6.81
No history – full mask history	41.12	56.25	28.82	44.10	27.01	21.87	25.87	24.99
Uniform history weights (no router)	5.53	7.75	6.73	11.98	8.96	7.15	**9.90**	12.14
Baseline with our sampler	33.53	47.75	21.85	43.21	47.09	40.57	24.49	25.62

the corresponding SLM results improve from 10.11 to 5.76 without guidance and from 9.95 to 4.82 with CFG. The improvement also carries over to MDLM in every setting, including a reduction from 14.34 to 6.48 on melanoma with CFG (2.2× lower).

On conditional promoter design (Table 2), the SLM-based model reduces Sei-predictor MSE from 0.0303 to 0.0267, the lowest mean in the table. Together, these results show consistent gains across diffusion backbones, enhancer datasets, guidance settings, and promoter design.

Robustness to guidance scale. Figure 4 examines whether the advantage of TC-NMD depends on a particular classifier-free guidance scale γ. Across the full range of guidance strengths, TC-SLM consistently achieves lower FBD than the corresponding SLM model on both Fly Brain and Human Melanoma. Although the optimal guidance strength varies across datasets, the performance gap persists throughout the sweep.

3.2 Ablations

Contribution of trajectory history. To isolate the contribution of history, we replace every cached trajectory state with a fully masked state while retaining the history-conditioning pathway. We also report a single-view Markovian baseline, which ignores history entirely. The full-mask replacement sharply increases FBD in every setting, showing that the denoiser uses information from the generated trajectory. To check that this result is not solely due to the full-mask replacement, we run the history-free baseline with our sampler, without supplying artificial history inputs. It also performs substantially worse than TC-NMD in every setting (Table 3). Together, these comparisons support the contribution of informative history across both backbones.

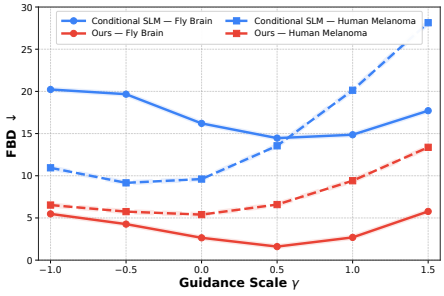

Figure 4: Sensitivity to classifier-free guidance scale γ on Fly Brain and Human Melanoma. FBD is shown for SLM and the SLM-based TC-NMD model at 5 NFE; lower is better.

Contribution of learned routing. Learned routing improves three of four settings at 5 steps, including the conditional tasks for both SLM and MDLM. At 10 steps, it improves all four settings, with larger gains over uniform weighting. This is consistent with routing becoming more useful as the number of past states grows (Table 3).

Bias controls reconsideration. Figure 5 (left) and Table 7 vary the inference bias for SLM on Melanoma. *Re-expansion* measures candidates reopened during sampling, whether or not they are ultimately selected. *Reconsideration* measures cases where the final emitted base was absent from the shortlist at an earlier intermediate step. From bias 0 to 0.07, reconsideration rises from 10.65% to 12.67% and FBD falls from 6.10 to 5.31. Further increasing the bias does not improve quality: at 0.20, FBD rises to 8.75. Re-expansion falls across the sweep, consistent with fewer intermediate decisions being made early enough to reopen. The best result occurs at an intermediate bias that permits useful corrections while allowing the sequence to resolve.

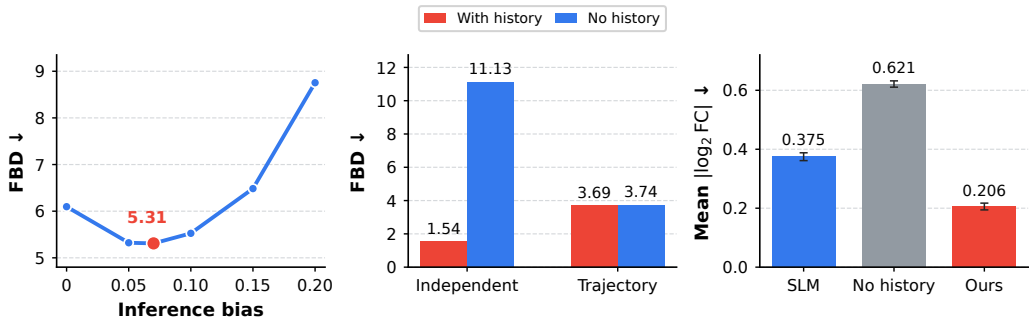

Figure 5: **Left:** FBD versus SLM inference bias on unconditional Melanoma. **Middle:** FBD on conditional SLM FLyBrain for each corruption process crossed with history. **Right:** mean absolute $\log_2$ fold-change to real motif frequencies, on conditional FLyBrain dataset. Additional metrics and full numerical results are in Appendix H.1.

History is most useful when decisions can be reconsidered. Figure 5 (middle) separates the effect of our sampler from that of the trained denoiser (full numbers in Table 8). *Trajectory* uses the original diffusion sampling process with our trained denoiser; *independent* uses our re-corrupting sampler. The no-history setting replaces cached states with all-mask states. Under trajectory sampling, history makes little difference (3.69 versus 3.74 FBD). Under independent sampling, removing informative history raises FBD from 1.54 to 11.13. Independent sampling with history also improves substantially over trajectory sampling with the same denoiser (1.54 versus 3.69). Thus, this gain comes from allowing decisions to be reconsidered and using the resulting history to guide those decisions.

Computational overhead. The router and multi-view machinery add little cost: 0.056% parameters, a $1.0014\times$ increase in forward MACs, and 1.10–$1.15\times$ wall-clock at inference and training respectively. We give the full measurement, including memory and its scaling with the number of history states, in Appendix E (Table 6).

3.3 Motif-Profile Agreement

Finally, we assess biological plausibility through motif frequencies, independently of the learned FBD classifier. For each transcription-factor motif, we count matches in generated and real unconditional fly-brain enhancers using its HOMER threshold and the real base composition as background. Figure 5 (right) compares the resulting frequency profiles for SLM-based models at 5 generation steps; Pearson and Spearman correlations are reported in Table 9.

TC-NMD achieves the strongest motif-profile agreement on every metric. It reaches a Pearson correlation of 0.9697 and a Spearman correlation of 0.9837, compared with 0.9180 and 0.9590 for SLM. Its mean absolute $\log_2$ fold-change is 0.2057, approximately 45% lower than SLM and 67% lower than the no-history variant. The gains in both correlation and fold-change show that TC-NMD more closely reproduces the ordering and frequencies of natural enhancer motifs. Because motif counting does not use the FBD classifier, this provides independent evidence for improved sequence quality.

4 Conclusion

We introduced TC-NMD, a non-Markovian discrete diffusion framework whose reverse step conditions on noisier states already visited during sampling, alongside the current state. Its independent-corruption forward makes this trajectory informative beyond any single state, while a biased inference schedule allows earlier decisions to be reconsidered. The SLM-based formulation admits a variational bound with finite per-step KL terms and is trained with a stable cross-entropy surrogate. Applied to both SLM and MDLM, TC-NMD improves cell-type enhancer design; the SLM-based model also improves conditional promoter design. Ablations show that the denoiser uses informative history, with learned routing becoming particularly valuable as the trajectory grows.

AI USAGE CLARIFICATION

We used large language models (LLMs) as a general-purpose writing and coding assistant. Specifically, LLMs were used to help organise the sectioning and structure of the manuscript, to improve the clarity and readability, and to assist with routine software engineering tasks such as writing boilerplate code, refactoring, and debugging. All research contributions reported in this paper- the problem formulation, the non-Markovian forward process, the routed reverse, the theoretical analysis, and the experimental design- originate from the authors. LLMs were not used to generate research ideas, to design the method, or to interpret results. Every LLM-assisted component was reviewed, verified, and edited by the authors: generated text was checked against our own claims and results, and generated code was read, tested, and validated against the numbers reported in the paper. The authors take full responsibility for all content of this submission.

ETHICS STATEMENT

All experiments use publicly available data (the human promoter, human melanoma enhancer, and *Drosophila* brain enhancer collections distributed with the Dirichlet flow matching benchmark (Stärk et al., 2024)). Generative models for biological sequence design carry dual-use considerations. Our models operate on cis-regulatory elements, which modulate expression of existing genes rather than encoding novel protein function, so we regard the immediate misuse potential as low relative to methods that design coding sequences.

REPRODUCIBILITY STATEMENT

We provide our implementation, including training and evaluation scripts for both TC-SLM and TC-MDLM and the scripts that reproduce the tables and figures in this paper, as supplementary material. The architecture, training, and sampling setup are described in Appendix D.1, with all hyperparameters listed in Table 4. Dataset sources and the exact train/validation/test splits are given in Appendix F, and Appendix G states which results we produce and which are quoted from prior work.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *CoRR*, abs/2107.03006, 2021. URL https://arxiv.org/abs/2107.03006.

Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1276–1301. PMLR, 2023. URL https://proceedings.mlr.press/v202/avdeyev23a.html.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b5b528767aa35f5b1a60fe0aaeca0563-Abstract-Conference.html.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi S. Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5453–5512. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/campbell24a.html.

Kathleen Chen, Aaron Wong, Olga Troyanskaya, and Jian Zhou. A sequence-based global map of regulatory activity for deciphering human genetics. *Nature Genetics*, 54:1–10, 07 2022. doi: 10.1038/s41588-022-01102-2.

Ting Chen, Ruixiang Zhang, and Geoffrey E. Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=3itjR9QxFw.

Zixiang Chen, Huizhuo Yuan, Yongqian Li, Yiwen Kou, Junkai Zhang, and Quanquan Gu. Fast sampling via discrete non-markov diffusion models with predetermined transition time. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/c153077e44a810cc8728460953af54f1-Abstract-Conference.html.

Oscar Davis, Samuel Kessler, Mircea Petrache, .Ismail .Ilkan Ceylan, Michael M. Bronstein, and Avishek Joey Bose. Fisher flow matching for generative modeling over discrete data. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/fadec8f2e65f181d777507d1df69b92f-Abstract-Conference.html.

Justin Deschenaux, Caglar Gulcehre, and Subham Sekhar Sahoo. The diffusion duality, chapter II: ψ-samplers. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RSIoYWIzaP.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.),

Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/f0d629a734b56a642701bba7bc8bb3ed-Abstract-Conference.html.

Sager J. Gosai, Rodrigo I. Castro, N. Fuentes, John C. Butts, Kousuke Mouri, Michael Alasoadura, Susan Kales, Thanh Thanh L. Nguyen, Ramil R. Noche, Arya S. Rao, Mary T. Joy, Pardis C. Sabeti, Steven K. Reilly, and Ryan Tewhey. Machine-guided design of cell-type-targeting cis-regulatory elements. *Nature*, 634:1211 – 1220, 2024. URL https://api.semanticscholar.org/CorpusID:273548985.

Jiaqi Han, Austin Wang, Minkai Xu, Wenda Chu, Meihua Dang, Haotian Ye, Huayu Chen, Yisong Yue, and Stefano Ermon. Discrete diffusion trajectory alignment via stepwise decomposition. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=h9b5h69v3p.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598, 2022. doi: 10.48550/ARXIV.2207.12598. URL https://doi.org/10.48550/arXiv.2207.12598.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 12454–12465, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/67d96d458abdef21792e6d8e590244e7-Abstract.html.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

Jasper Janssens, Sara Aibar, Ibrahim Ihsan Taskiran, Joy N. Ismail, Alicia Estacio Gomez, Gabriel N. Aughey, Katina I. Spanier, Florian V. De Rop, Carmen Bravo González-Blas, Marc S. Dionne, Krista Grimes, Xiao-Jiang Quan, Dafni Papasokrati, Gert J. Hulselmans, Samira Makhzami, Maxime de Waegeneer, Valerie Christiaens, Tony D. Southall, and Stein Aerts. Decoding gene regulation in the fly brain. *Nature*, 601:630 – 636, 2021. URL https://api.semanticscholar.org/CorpusID:237099830.

Zeynep Kalender Atak, Ibrahim Taskiran, Jonas Demeulemeester, Christopher Flerin, David Mauduit, Liesbeth Minnoye, Gert Hulselmans, Valerie Christiaens, Ghanem-Elias Ghanem, Jasper Wouters, and Stein Aerts. Interpretation of allele-specific chromatin accessibility using cell state–aware deep learning. *Genome Research*, 31:1082–1096, 04 2021. doi: 10.1101/gr.260851.120.

Zehui Li, Yuhao Ni, William A. V. Beardall, Guoxuan Xia, Akashaditya Das, Guy-Bart Stan, and Yiren Zhao. Discdiff: Latent diffusion model for DNA sequence generation. *CoRR*, abs/2402.06079, 2024. doi: 10.48550/ARXIV.2402.06079. URL https://doi.org/10.48550/arXiv.2402.06079.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32819–32848. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

Subham S. Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Anirban Sarkar, Yijie Kang, Nirali Somia, Pablo Mantilla, Jessica Zhou, Masayuki Nagai, Ziqi Tang, Chris Zhao, and Peter Koo. Designing dna with tunable regulatory activity using score-entropy discrete diffusion, 05 2025.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/bad233b9849f019aead5e5cc60cef70f-Abstract-Conference.html.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Yuxuan Song, Zhe Zhang, Yu Pei, Jingjing Gong, Qiying Yu, Zheng Zhang, Mingxuan Wang, Hao Zhou, Jingjing Liu, and Wei-Ying Ma. Shortlisting model: A streamlined simplex diffusion for discrete variable generation. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/42c4ca570eec68981c20094de1cc760e-Abstract-Conference.html.

Hannes Stärk, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi S. Jaakkola. Dirichlet flow matching with applications to DNA sequence design. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46495–46513. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/stark24b.html.

Ibrahim Taskiran, Katina Spanier, Hannah Dickmänken, Niklas Kempynck, Alexandra Pančíková, Eren Ekşi, Gert Hulselmans, Joy Ismail, Koen Theunis, Roel Vandepoel, Valerie Christiaens, David Mauduit, and Stein Aerts. Cell type directed design of synthetic enhancers. *Nature*, 626, 12 2023. doi: 10.1038/s41586-023-06936-2.

Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Avantika Lal, Tommi S. Jaakkola, Sergey Levine, Aviv Regev, Hanchen Wang, and Tommaso Biancalani. Fine-tuning discrete diffusion models via reward optimization with applications to DNA and protein design. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a. URL https://openreview.net/forum?id=G328D1xt4W.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *The Thirty-ninth Annual Conference on Neural*

Information Processing Systems, 2025b. URL https://openreview.net/forum?id=IJryQAOy0p.

Linxuan Wang, Ziyi Wang, Yikun Bai, Wei Deng, Guang Lin, and Qifan Song. Generalized discrete diffusion with self-correction. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=zQKlzKB6I9.

Yangtian Zhang, Sizhuang He, Daniel Levine, Lawrence Zhao, David Zhang, Syed A Rizvi, Shiyang Zhang, Emanuele Zappala, Rex Ying, and David van Dijk. Non-markovian discrete diffusion with causal language models, 2025. URL https://arxiv.org/abs/2502.09767.

A Detailed SLM Background

This appendix expands on the SLM background of Section 2.1, giving the reverse parameterisation and training loss in full, together with the ELBO derivation and the gradient-rescaling argument behind the simplified cross-entropy objective.

A.1 Reverse Parameterisation and Training Loss

SLM parameterises the reverse step as a Bernoulli posterior, substituting the network's $\mathbf{x}_0^{\mathbf{c}}$-prediction for the unknown clean state:

$$p_\theta(\mathbf{x}_{t-1}^{\mathbf{c}} \mid \mathbf{x}_t^{\mathbf{c}}) = \mathrm{Bern}\left(\widehat{\mathbf{x}}_0^\theta + (\mathbf{1} - \widehat{\mathbf{x}}_0^\theta)\,\frac{\bar{\alpha}_{t-1}}{\bar{\alpha}_t}\right) \odot \mathbf{x}_t^{\mathbf{c}}, \tag{14}$$

where $\widehat{\mathbf{x}}_0^\theta = \mathrm{NN}_\theta(\mathbf{x}_t^{\mathbf{c}}, t)$ is a softmax over K candidates and the elementwise mask $\odot\,\mathbf{x}_t^{\mathbf{c}}$ enforces that only candidates currently in the shortlist receive non-zero probability (implemented by adding $-\infty$ to logits outside $\mathbf{x}_t^{\mathbf{c}}$ before the softmax). Training uses the simplified cross-entropy loss

$$\mathcal{L}_{\mathrm{SLM}} = -\mathbb{E}_{t,\mathbf{x}_0^{\mathbf{c}},\mathbf{x}_t^{\mathbf{c}}}\left[\langle \log \mathrm{NN}_\theta(\mathbf{x}_t^{\mathbf{c}}, t),\, \mathbf{x}_0^{\mathbf{c}}\rangle\right], \tag{15}$$

a gradient-rescaled approximation of the per-step Bernoulli KL induced by equation 1 and equation 14, derived below.

A.2 Markov Forward and Posterior

The SLM forward process is a first-order Markov chain over candidate-set states:

$$q(\mathbf{x}_t^{\mathbf{c}} \mid \mathbf{x}_{t-1}^{\mathbf{c}}) = \mathrm{Bern}\big(\mathbf{x}_{t-1}^{\mathbf{c}} + (\mathbf{1} - \mathbf{x}_{t-1}^{\mathbf{c}}) \odot \boldsymbol{\alpha}_t\big), \tag{16}$$

where $\boldsymbol{\alpha}_t = (\bar{\alpha}_t - \bar{\alpha}_{t-1})/(1 - \bar{\alpha}_{t-1})$ is the per-step addition rate that produces the marginal of Equation 1. The chain has deterministic monotonicity: if a candidate is in the shortlist at step $t-1$, it remains in the shortlist at step t. Combined with $\mathbf{x}_0^{\mathbf{c}} \subseteq \mathbf{x}_1^{\mathbf{c}}$ at the boundary, this gives $\mathbf{x}_0^{\mathbf{c}} \subseteq \mathbf{x}_t^{\mathbf{c}}$ for all t, so the true token is always in the shortlist.

The closed-form posterior, derived in Song et al. (2025), is

$$q(\mathbf{x}_{t-1}^{\mathbf{c}} \mid \mathbf{x}_t^{\mathbf{c}}, \mathbf{x}_0^{\mathbf{c}}) = \mathrm{Bern}\left(\mathbf{x}_0^{\mathbf{c}} + \left[(\mathbf{1} - \mathbf{x}_0^{\mathbf{c}}) \odot \mathbf{x}_t^{\mathbf{c}}\right]\,\frac{\bar{\alpha}_{t-1}}{\bar{\alpha}_t}\right). \tag{17}$$

Note the elementwise mask $\odot\mathbf{x}_t^{\mathbf{c}}$ in the non-true-label term: the target distribution places mass on a non-true candidate only if that candidate is currently in the shortlist. This mask is what makes the posterior have the same support as the SLM-form reverse model (Equation 14).

A.3 Per-Step ELBO

The ELBO decomposes per timestep as

$$\mathcal{L}_{\mathrm{ELBO}} = -\sum_{t=1}^{T} L_{t-1}, \tag{18}$$

with

$$L_{t-1} = \mathbb{E}_{q(\mathbf{x}_t^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}})}\left[D_{\mathrm{KL}}\left[q(\mathbf{x}_{t-1}^{\mathbf{c}} \mid \mathbf{x}_t^{\mathbf{c}}, \mathbf{x}_0^{\mathbf{c}}) \,\middle\|\, p_\theta(\mathbf{x}_{t-1}^{\mathbf{c}} \mid \mathbf{x}_t^{\mathbf{c}})\right]\right]. \tag{19}$$

Both arguments are Bernoulli with closed-form parameters; the KL decomposes per-position and per-candidate as

$$L_{t-1} = \mathbb{E}_{q(\mathbf{x}_t^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}})}\left[\sum_{i:\,(\mathbf{x}_t^{\mathbf{c}})^i > 0} D_{\mathrm{KL}}\left[\mathrm{Bern}(g_t^i) \,\middle\|\, \mathrm{Bern}(p_{\theta,t}^i)\right]\right], \tag{20}$$

where

$$g_t^i = (\mathbf{x}_0^{\mathbf{c}})^i + (1 - (\mathbf{x}_0^{\mathbf{c}})^i)\,\frac{\bar{\alpha}_{t-1}}{\bar{\alpha}_t}, \tag{21}$$

$$p_{\theta,t}^i = \mathrm{NN}_\theta^i + (1 - \mathrm{NN}_\theta^i)\,\frac{\bar{\alpha}_{t-1}}{\bar{\alpha}_t}. \tag{22}$$

A.4 Gradient Rescaling and the Simplified Loss

Song et al. (2025) (§3.4) observe that direct optimization of the Bernoulli KL is prone to vanishing gradients near the boundary of the schedule. The gradient of L_{t-1} with respect to the network softmax output NN_θ^i contains a multiplicative factor $(\bar\alpha_t - \bar\alpha_{t-1})/\bar\alpha_t$ that approaches zero as $t \to T$. SLM addresses this by reparameterizing through a log-softmax and rescaling the per-position gradient by $1/\mathrm{NN}_\theta^i$, yielding the reweighted cross-entropy

$$L_{t-1}^{\text{weight}} = -\frac{\bar\alpha_t - \bar\alpha_{t-1}}{\bar\alpha_t}\,\langle \log \mathrm{NN}_\theta(\mathbf{x}_t^{\mathbf{c}}, t),\, \mathbf{x}_0^{\mathbf{c}} \rangle. \tag{23}$$

The simplified loss $\mathcal{L}_{\mathrm{SLM}}$ (Equation 15) drops the time-weighting factor for further simplicity. Both are gradient-rescaled approximations of the underlying ELBO and are empirically stable.

B Non-Markovian Variational Bound and the Training Surrogate

This appendix (i) derives the variational bound induced by the non-Markovian generative factorisation, (ii) shows its per-step term is finite for the reverse kernel of Equation 7, (iii) proves that the cross-entropy surrogate is population-consistent for the low-noise kernel, and (iv) records where masking is and is not applied.The bound and the finiteness result concern the full-support kernel of Equation 7; the hard support intersection used during high-noise decoding is a separate inference-time projection and is treated as a decoding heuristic.

B.1 Trajectory-Level Non-Markov Variational Bound

We first derive the complete variational bound associated with the independent-corruption forward. This establishes that conditioning the reverse on the previously generated trajectory does not alter the standard variational argument; it changes only the factorisation of the forward posterior in each per-step term.

Let $q(\mathbf{x}_{1:T}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}}) = \prod_{\tau=1}^T q_\tau(\mathbf{x}_\tau^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}})$ be the independent-corruption forward of Equation 2, and let $\mathbf{h}_t = H_\phi(\mathbf{x}_{t:T}^{\mathbf{c}})$ be the history summary produced by the router. We define the non-Markov generative model by the backward factorisation

$$p_{\theta,\phi}(\mathbf{x}_{0:T}^{\mathbf{c}}) = p_T(\mathbf{x}_T^{\mathbf{c}}) \prod_{t=1}^T p_\theta\big(\mathbf{x}_{t-1}^{\mathbf{c}} \mid \mathbf{x}_t^{\mathbf{c}}, \mathbf{h}_t\big)\,. \tag{24}$$

Equation 24 is a normalised joint because, when sampling backward, $\mathbf{x}_{t:T}^{\mathbf{c}}$ (and hence $\mathbf{h}_t$) is available before drawing $\mathbf{x}_{t-1}^{\mathbf{c}}$.

Proposition 1 (Non-Markov variational bound)**.** *For the forward and generative distributions above,*

$$\log p_{\theta,b}(\mathbf{x}_0^{\mathbf{c}}) \;\geq\; -\mathcal{L}_T - \sum_{t=1}^T \mathcal{L}_{t-1}, \tag{25}$$

where $\mathcal{L}_T = D_{\mathrm{KL}}[q_T(\mathbf{x}_T^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}}) \,\|\, p_T(\mathbf{x}_T^{\mathbf{c}})]$ *and, for* $t \geq 2$*,*

$$\mathcal{L}_{t-1} = \mathbb{E}_{q(\mathbf{x}_{t:T}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}})}\big[D_{\mathrm{KL}}\big[q_{t-1}(\mathbf{x}_{t-1}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}}) \,\|\, p_\theta(\mathbf{x}_{t-1}^{\mathbf{c}} \mid \mathbf{x}_t^{\mathbf{c}}, \mathbf{h}_t)\big]\big]\,. \tag{26}$$

Defining $q_0(\cdot \mid \mathbf{x}_0^{\mathbf{c}}) = \delta_{\mathbf{x}_0^{\mathbf{c}}}$*, the* $t = 1$ *term is the reconstruction* $\mathcal{L}_0 = -\mathbb{E}_{q(\mathbf{x}_{1:T}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}})} \log p_\theta(\mathbf{x}_0^{\mathbf{c}} \mid \mathbf{x}_1^{\mathbf{c}}, \mathbf{h}_1)$.

Proof. Introducing the forward process as a variational distribution and applying Jensen's inequality gives

$$\log p_{\theta,\phi,b}(\mathbf{x}_0^{\mathbf{c}}) = \log \sum_{\mathbf{x}_{1:T}^{\mathbf{c}}} q(\mathbf{x}_{1:T}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}})\, \frac{p_{\theta,\phi,b}(\mathbf{x}_{0:T}^{\mathbf{c}})}{q(\mathbf{x}_{1:T}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}})} \;\geq\; \mathbb{E}_{q(\mathbf{x}_{1:T}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}})}\left[\log \frac{p_{\theta,\phi,b}(\mathbf{x}_{0:T}^{\mathbf{c}})}{q(\mathbf{x}_{1:T}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}})}\right]. \tag{27}$$

By the reverse chain rule, $q(\mathbf{x}_{1:T}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}}) = q_T(\mathbf{x}_T^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}}) \prod_{t=2}^T q(\mathbf{x}_{t-1}^{\mathbf{c}} \mid \mathbf{x}_{t:T}^{\mathbf{c}}, \mathbf{x}_0^{\mathbf{c}})$. The forward states are conditionally independent given $\mathbf{x}_0^{\mathbf{c}}$, so

$$q(\mathbf{x}_{t-1}^{\mathbf{c}} \mid \mathbf{x}_{t:T}^{\mathbf{c}}, \mathbf{x}_0^{\mathbf{c}}) = q_{t-1}(\mathbf{x}_{t-1}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}}). \tag{28}$$

Substituting Equations 24 and 28 into 27 and grouping terms gives

$$\begin{aligned}\log p_{\theta,b}(\mathbf{x}_0^{\mathbf{c}}) \geq\ & -D_{\mathrm{KL}}[q_T(\mathbf{x}_T^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}}) \,\|\, p_T(\mathbf{x}_T^{\mathbf{c}})] \\ & - \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_{t:T}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}})} D_{\mathrm{KL}}\left[q_{t-1}(\mathbf{x}_{t-1}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}}) \,\|\, p_{\theta,b}(\mathbf{x}_{t-1}^{\mathbf{c}} \mid \mathbf{x}_t^{\mathbf{c}}, \mathbf{h}_t)\right] \\ & + \mathbb{E}_{q(\mathbf{x}_{1:T}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}})} \log p_{\theta,b}(\mathbf{x}_0^{\mathbf{c}} \mid \mathbf{x}_1^{\mathbf{c}}, \mathbf{h}_1). \end{aligned} \tag{29}$$

Writing the final line as the negative KL from $\delta_{\mathbf{x}_0^{\mathbf{c}}}$ to the reconstruction distribution yields Equation 25. □

Equation 28 makes the per-step target the marginal $q_{t-1}(\mathbf{x}_{t-1}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}})$. The reverse kernel uses the current state and trajectory history to approximate this target; the training surrogate is discussed in Section B.3.

B.2 Per-Step Posterior, Reverse Kernel, and Finiteness

For an intermediate reverse step $t \geq 2$, conditional independence gives

$$q_{t-1}(\mathbf{x}_{t-1}^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}}) = \mathrm{Bern}(\mathbf{x}_0^{\mathbf{c}} + (\mathbf{1} - \mathbf{x}_0^{\mathbf{c}})\bar{\alpha}_{t-1})\,. \tag{30}$$

We define the biased reverse rate as

$$r_{t-1}^{(d)} = \mathrm{clip}_{[0,1)}(\bar{\alpha}_{t-1} + b_d)\,, \tag{31}$$

where $b_d \geq 0$ is selected for each dataset. The reverse kernel is

$$p_{\theta,b}(\mathbf{x}_{t-1}^{\mathbf{c}} \mid \mathbf{x}_t^{\mathbf{c}}, \mathbf{h}_t) = \mathrm{Bern}\Big(\widehat{\mathbf{x}}_0^{\,\theta} + (\mathbf{1} - \widehat{\mathbf{x}}_0^{\,\theta}) r_{t-1}^{(d)}\Big)\,, \tag{32}$$

where $\widehat{\mathbf{x}}_0^{\,\theta} = \mathrm{softmax}(\mathrm{NN}_\theta(\mathbf{x}_t^{\mathbf{c}}, \mathbf{h}_t, t))$, NN_θ is the denoising network returning logits, and $\mathbf{h}_t$ is the history summary. The bias is therefore included in the reverse distribution appearing in the ELBO.

Proposition 2 (Finite per-step KL)**.** *For finite network logits and $r_{t-1}^{(d)} \in [0, 1)$, every intermediate per-step KL in Equation 26 is finite.*

Proof. For each candidate i, the target and reverse inclusion probabilities are

$$g_i = (\mathbf{x}_0^{\mathbf{c}})_i + (1 - (\mathbf{x}_0^{\mathbf{c}})_i)\,\bar{\alpha}_{t-1}, \qquad p_i = (\widehat{\mathbf{x}}_0^{\,\theta})_i + \big(1 - (\widehat{\mathbf{x}}_0^{\,\theta})_i\big)\, r_{t-1}^{(d)}. \tag{33}$$

Finite softmax logits give $0 < (\widehat{\mathbf{x}}_0^{\,\theta})_i < 1$. Since $0 \leq r_{t-1}^{(d)} < 1$, we also have $0 < p_i < 1$. Thus every outcome with positive probability under the forward posterior has positive probability under the reverse.

The KL decomposes into a sum over candidates, where $d_{\mathrm{B}}(u\|v) := D_{\mathrm{KL}}(\mathrm{Bern}(u)\|\,\mathrm{Bern}(v))$ denotes the Bernoulli KL divergence:

$$d_{\mathrm{B}}(g_i\|p_i) = g_i \log \frac{g_i}{p_i} + (1 - g_i) \log \frac{1 - g_i}{1 - p_i}. \tag{34}$$

using $0 \log 0 = 0$. Each term is finite, including when g_i is zero or one. Summing over candidates and positions, then averaging over the finite forward state space, preserves finiteness. □

Choosing the all-candidate prior gives $\mathcal{L}_T = 0$. At the final step, no re-corruption is applied. A categorical clean-token likelihood with probabilities $\widehat{\mathbf{x}}_0^{\,\theta}$ gives a finite cross-entropy reconstruction term.

B.3 Cross-Entropy Surrogate and the Bias-Dependent Minimum

The rate bias leaves a KL term even under a perfect clean-token prediction. To quantify it, consider one position whose clean token is j. Write $a = \bar{\alpha}_{t-1}$, $r = r_{t-1}^{(d)}$, and $v_i = (\widehat{\mathbf{x}}_0^\theta)_i$, and assume $a \le r < 1$. The per-position KL is

$$D_t(\mathbf{v}) = -\log(r + (1-r)v_j) + \sum_{i \neq j} d_{\mathrm{B}}(a \| r + (1-r)v_i)\,. \tag{35}$$

If $\mathbf{v}$ approaches the one-hot clean token, the true candidate contributes zero and each of the $K-1$ incorrect candidates contributes $d_{\mathrm{B}}(a\|r)$. We call their sum the KL floor induced by the bias:

$$C_{t,b} = (K-1) d_{\mathrm{B}}(a\|r). \tag{36}$$

For a fixed dataset, timestep, and bias, $C_{t,b}$ is independent of the denoiser parameters. It is positive whenever $r > a$.

Proposition 3 (Cross-entropy bound)**.** *Let $\ell_{\mathrm{CE}}(\mathbf{v}) = -\log v_j$. The infimum of $D_t(\mathbf{v})$ is $C_{t,b}$, approached as $\mathbf{v} \to \mathbf{x}_0^{\mathbf{c}}$. Moreover,*

$$C_{t,b} \le D_t(\mathbf{v}) \le (2 - a - r)\ell_{\mathrm{CE}}(\mathbf{v}) + C_{t,b}. \tag{37}$$

Proof. For an incorrect candidate $i \neq j$, the forward inclusion probability is a, and the reverse inclusion probability is $p_i = r + (1-r)v_i$. Since $v_i \ge 0$ and $r \ge a$, we have $p_i \ge r \ge a$. The Bernoulli KL is minimised when its two probabilities agree. Its derivative with respect to p_i is

$$\frac{\partial}{\partial p_i} d_{\mathrm{B}}(a\|p_i) = \frac{p_i - a}{p_i(1-p_i)} \ge 0 \qquad \text{for } p_i \ge a.$$

Thus, among the reverse probabilities allowed here, the incorrect candidate contributes at least $d_{\mathrm{B}}(a\|r)$, approached as $v_i \to 0$.

For the true candidate j, the forward inclusion probability is 1, so its KL contribution is $d_{\mathrm{B}}(1\|p_j) = -\log p_j \ge 0$. This term approaches zero as $v_j \to 1$. Because $\mathbf{v}$ sums to one, $v_j \to 1$ also makes every incorrect $v_i \to 0$. All the lower bounds are therefore approached together, giving $C_{t,b} = (K-1)d_{\mathrm{B}}(a\|r)$.

For the upper bound, concavity of the logarithm gives

$$-\log(r + (1-r)v_j) \le (1-r)(-\log v_j).$$

For an incorrect candidate $i \neq j$, the forward inclusion probability is a, and the reverse inclusion probability is $p_i = r + (1-r)v_i$. If $v_i = 0$, then $p_i = r$, and this candidate contributes $d_{\mathrm{B}}(a\|r)$ to the KL floor. We bound its additional contribution when $v_i > 0$:

$$\begin{aligned} d_{\mathrm{B}}(a\|p_i) - d_{\mathrm{B}}(a\|r) &= a\left(\log\frac{a}{p_i} - \log\frac{a}{r}\right) + (1-a)\left(\log\frac{1-a}{1-p_i} - \log\frac{1-a}{1-r}\right) \\ &= a\log\frac{r}{p_i} + (1-a)\log\frac{1-r}{1-p_i}. \end{aligned}$$

Because $1 - p_i = (1-r)(1-v_i)$, the second logarithm equals $-\log(1-v_i)$. Also, $p_i \ge r$, so $\log(r/p_i) \le 0$. Removing this nonpositive term gives

$$d_{\mathrm{B}}(a\|p_i) - d_{\mathrm{B}}(a\|r) \le -(1-a)\log(1-v_i). \tag{38}$$

We now sum over all incorrect candidates. Since $\mathbf{v}$ is a probability vector, $\sum_{i\neq j} v_i = 1 - v_j$. For values in $[0,1]$,

$$\prod_{i\neq j}(1-v_i) \ge 1 - \sum_{i\neq j} v_i = v_j.$$

Taking negative logarithms therefore gives

$$\sum_{i\neq j} -\log(1-v_i) = -\log\prod_{i\neq j}(1-v_i) \le -\log v_j.$$

Consequently, the total KL above the floor from incorrect candidates is at most $(1-a)(-\log v_j)$. The true candidate contributes at most $(1-r)(-\log v_j)$, as shown above. Adding the two bounds yields

$$D_t(\mathbf{v}) - C_{t,b} \le \big[(1-a)+(1-r)\big](-\log v_j) = (2-a-r)\ell_{\mathrm{CE}}(\mathbf{v}).$$

□

At a fixed noise level, minimising expected cross-entropy minimises the upper bound in Equation 37, because $C_{t,b}$, a, and r do not depend on the denoiser. The bound guarantees that the KL approaches $C_{t,b}$ as the cross-entropy approaches zero.

C Classifier-Free Guidance

For class-conditional generation we use classifier-free guidance (Ho & Salimans, 2022): training drops the class label c with probability p_{drop} (replaced by a null token $\varnothing$), and at sampling the $\mathbf{x}_0^{\mathbf{c}}$-prediction extrapolates from the unconditional toward the conditional path,

$$\widehat{\mathbf{x}}_0^{\mathrm{cfg}} = (1+w)\,\mathrm{NN}_\theta(\tilde{\mathbf{x}^{\mathbf{c}}}_t, t, \pi_\phi, c) - w\,\mathrm{NN}_\theta(\tilde{\mathbf{x}^{\mathbf{c}}}_t, t, \pi_\phi, \varnothing), \tag{39}$$

followed by simplex projection. The history summary $\mathbf{h}_t$ and router output π_ϕ are identical across the two paths, so the multi-view structure is preserved in both branches.

D Algorithms

This section gives the training and generation procedures for TC-NMD as pseudocode in 2, 1.

D.1 Implementation Details

Architecture. We retain each model's denoiser backbone and add the router described in Section 2.4. For SLM, the denoiser is a 20-layer 1D convolutional network with hidden dimension 128 and receives the history-augmented input $\tilde{\mathbf{x}^{\mathbf{c}}}_t$. The router has two layers, with one 1D convolution per state stream and a bilinear compatibility head. It adds 0.056% parameters relative to the SLM denoiser.

Diffusion schedule. We use $T = 1000$ diffusion steps. The SLM-based model follows the monotone Bernoulli schedule $\bar{\alpha}_t$, and the MDLM-based model follows its masking schedule. In both models, noisy states at different timesteps are sampled independently given the clean sequence, as described in Section 2.3.

Training. Models are trained for 800 epochs with batch size 64/32 using AdamW (Loshchilov & Hutter, 2019), learning rate 5×10^{-4}, and a cosine schedule. The router's Gumbel temperature is 0.01 for Fly Brain and unconditional Melanoma, and 0.05 for conditional Melanoma. Classifier-free guidance uses label dropout $p_{\mathrm{drop}} = 0.3$. All experiments use four H200 GPUs; training takes approximately three hours per task.

Sampling. We report results with 5 and 10 sampling steps. Intermediate steps permit earlier decisions to be reconsidered, and the final prediction resolves the remaining uncertainty without further re-corruption. For conditional generation, we sweep guidance scale $\gamma \in \{-0.5, 0, 0.5, 1, 1.5, 2, 3\}$. We select the guidance scale and re-corruption bias separately for each dataset and backbone on validation data, then evaluate the selected settings on test.

D.2 Router Architecture

The router π_ϕ is a two-layer module that scores each future trajectory state $\mathbf{x}_\tau^{\mathbf{c}}$ by its compatibility with $\mathbf{x}_t^{\mathbf{c}}$. Concretely, for each $\tau \in \{t+1, \dots, T\}$:

1. Project $\mathbf{x}_t^{\mathbf{c}}$ and $\mathbf{x}_\tau^{\mathbf{c}}$ to a shared embedding space via per-position 1D convolutions with kernel size $k = 1$.

Algorithm 1 Training step for TC-NMD.

Require: Clean sample $\mathbf{x}_0^{\mathbf{c}} \in \{0,1\}^{L\times K}$, optional class label c.
Require: Number of diffusion steps T, marginal schedule $\{\bar{\alpha}_\tau\}_{\tau=1}^{T}$.
Require: Denoiser NN_θ, router π_ϕ, classifier-free drop probability p_{drop}.

Sample trajectory: for each $\tau = 1, \dots, T$, draw independently

$$\mathbf{x}_\tau^{\mathbf{c}} \sim q(\mathbf{x}_\tau^{\mathbf{c}} \mid \mathbf{x}_0^{\mathbf{c}}) = \mathrm{Bern}(\mathbf{x}_0^{\mathbf{c}} + (\mathbf{1} - \mathbf{x}_0^{\mathbf{c}})\,\bar{\alpha}_\tau)\,.$$

Stack into $\mathbf{V} = \{\mathbf{x}_\tau^{\mathbf{c}}\}_{\tau=1}^{T}$.
Pick anchor timestep: sample $t \sim \mathrm{Uniform}\{1, \dots, T\}$.
Conditioning dropout (CFG): with probability p_{drop}, replace c with the null class $\varnothing$.
if $t = T$ **then** ▷ boundary: no noisier states above the anchor
    No-history bypass: set $\mathbf{h}_t \leftarrow \varnothing$ and $\tilde{\mathbf{x}^{\mathbf{c}}}_t \leftarrow \mathbf{x}_t^{\mathbf{c}}$.
else
    Router forward: compute compatibility scores between $\mathbf{x}_t^{\mathbf{c}}$ and the states $\{\mathbf{x}_\tau^{\mathbf{c}}\}_{\tau=t+1}^{T}$, mean-pool over positions, and apply Gumbel-Softmax:

$$\pi_\phi(\tau \mid t) = \mathrm{GumbelSoftmax}\big(\,\mathrm{score}(\mathbf{x}_t^{\mathbf{c}}, \mathbf{x}_\tau^{\mathbf{c}})\,;\, t < \tau \le T\,\big).$$

    History summary: $\mathbf{h}_t = \sum_{\tau=t+1}^{T} \pi_\phi(\tau \mid t)\,\mathbf{x}_\tau^{\mathbf{c}}$.
    Denoiser input: build the simplex-renormalized weighted combination

$$\tilde{\mathbf{x}^{\mathbf{c}}}_t = \frac{w_{\mathrm{cur}}(t)\,\mathbf{x}_t^{\mathbf{c}} + w_{\mathrm{hist}}(t)\,\mathbf{h}_t}{\|w_{\mathrm{cur}}(t)\,\mathbf{x}_t^{\mathbf{c}} + w_{\mathrm{hist}}(t)\,\mathbf{h}_t\|_1}, \quad w_{\mathrm{cur}}(t) = 1 + c_t,\ w_{\mathrm{hist}}(t) = 1 - c_t.$$

end if
Predict $\mathbf{x}_0^{\mathbf{c}}$**:** $\widehat{\mathbf{x}}_0^{\theta} \leftarrow \mathrm{softmax}(\mathrm{NN}_\theta(\tilde{\mathbf{x}^{\mathbf{c}}}_t,\, t,\, \pi_\phi,\, c))$ over all K candidates (no masking at training).
Cross-entropy loss:

$$\mathcal{L}_{\mathrm{CE}} = -\langle \log \widehat{\mathbf{x}}_0^{\theta},\, \mathbf{x}_0^{\mathbf{c}} \rangle.$$

Loss: $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{CE}}$ (gradient-rescaled cross-entropy on $\mathbf{x}_0^{\mathbf{c}}$).
Update: backpropagate through θ and ϕ jointly.

2. Compute per-position bilinear compatibility scores between the projected current state and the projected anchor state.
3. Mean-pool over positions to obtain a single scalar score s_τ per candidate timestep.
4. Apply Gumbel-Softmax (training) or softmax (inference) over $\{s_\tau\}_{\tau>t}$ to obtain $\pi_\phi(\tau \mid t)$.

Total parameters: $2{,}048$, or 0.056% of the denoiser parameter count in our experiments (measured; see Appendix E).

D.3 Hyperparameters

Table 4 lists the training and sampling hyperparameters used for the DNA enhancer and promoter experiments.

E Computational Overhead

Table 6 reports the cost of the router and history machinery relative to the Markovian SLM baseline, measured on the SLM path ($B = 64$, $L = 500$). The router adds negligible arithmetic: its added multiply-accumulates are 0.14% of the denoiser, so total forward MACs rise by only $1.0014\times$. The wall-clock and memory overheads are larger than the FLOP count because the cost is dominated by constructing and moving the multi-view buffer rather than by computation: training time rises by $1.15\times$ (average case, $M = 499$ history states) and sampling by $1.10\times$, with a $1.37\times$ training-memory increase from holding the view buffer and its construction pipeline resident. At inference the

Algorithm 2 Generation with TC-NMD using the SLM backbone.

Require: Denoiser NN_θ, router π_ϕ, sampling grid $T = \tau_S > \cdots > \tau_0 = 0$, schedule $\{\bar{\alpha}_\tau\}$, dataset–backbone bias b_d, clipping constant $\varepsilon > 0$, optional class c, guidance scale w.

$\mathbf{z}_{\tau_S} \leftarrow \mathbf{1}$ ▷ all candidates active
for $s = S, \dots, 2$ **do**
  $t \leftarrow \tau_s, \quad u \leftarrow \tau_{s-1}$
  if $s = S$ **then**
    $\tilde{\mathbf{z}}_t \leftarrow \mathbf{z}_t, \quad \pi_\phi \leftarrow \varnothing$ ▷ no earlier states
  else
    $\pi_\phi(\tau_j \mid t) \leftarrow \mathrm{softmax}_{j>s}\big(\mathrm{score}(\mathbf{z}_t, \mathbf{z}_{\tau_j})\big)$
    $\mathbf{h}_t \leftarrow \sum_{j=s+1}^{S} \pi_\phi(\tau_j \mid t)\mathbf{z}_{\tau_j}$
    $\tilde{\mathbf{z}}_t \leftarrow \dfrac{w_{\mathrm{cur}}(t)\mathbf{z}_t + w_{\mathrm{hist}}(t)\mathbf{h}_t}{\|w_{\mathrm{cur}}(t)\mathbf{z}_t + w_{\mathrm{hist}}(t)\mathbf{h}_t\|_1}$
  end if
  $\widehat{\mathbf{x}}_0 \leftarrow \mathrm{NN}_\theta(\tilde{\mathbf{z}}_t, t, \pi_\phi, c)$
  if $c \neq \varnothing$ and $w \neq 0$ **then**
    Apply CFG to $\widehat{\mathbf{x}}_0$ using Equation 39, then project onto the simplex
  end if
  $r_u^{(d)} \leftarrow \min\{1 - \varepsilon, \bar{\alpha}_u + b_d\}$
  $\mathbf{z}_u \sim \mathrm{Bern}\Big(\widehat{\mathbf{x}}_0 + (\mathbf{1} - \widehat{\mathbf{x}}_0) r_u^{(d)}\Big)$ ▷ excluded candidates may return
end for
Using $\mathbf{z}_1$ and the cached states $\{\mathbf{z}_{\tau_s}\}_{s=2}^{S}$, compute the routed history and make one final denoiser prediction at $\tau = 0$
Apply CFG if requested.
return the highest-probability candidate at each position ▷ no further re-corruption

Table 4: Hyperparameters for the DNA experiments.

Hyperparameter	**Value**
Denoiser backbone	20-layer 1D CNN (4 stacks $\times$ 5 layers), hidden dim 128 kernel 9, dilations $1/1/4/16/64$ per stack
Denoiser parameters	3,666,628
Router	two 1×1 convolutions ($4 \to 256$), scaled dot-product score
Router parameters	2,048 (0.056% of denoiser)
Router temperature τ	0.01; 0.05 (Melanoma, conditional SLM) Gumbel-softmax at training, softmax at inference
Optimiser	AdamW (Loshchilov & Hutter, 2019), weight decay 0.01
Learning rate	5×10^{-4}
EMA decay	0.9999
Epochs	800 (enhancer) / 200 (promoter)
Batch size	64 (Fly Brain) / 32 (Melanoma) / 32 (promoter)
Sequence length	500 (enhancer) / 1024 (promoter)
Classes	81 (Fly Brain) / 47 (Melanoma)
CFG dropout p_{drop}	0.3
Diffusion steps T (training)	1000
Noise schedule	log-linear Bernoulli
Sampling NFE	5
Hardware	4$\times$ H200 GPUs

overhead is $1.10\times$ time and $1.02\times$ memory. The router itself accounts for only $2{,}048$ parameters, 0.056% of the model.

Table 5: Sampling hyperparameters. The corrector bias b is the additive floor on the re-activation (SLM) / re-masking (MDLM) probability; w is the classifier-free guidance scale. The simple baselines have no corrector and therefore no b.

Model	**Setting**	**Bias** b	**Guidance** w
SLM (ours)	Fly Brain, conditional	0	0.5
	Fly Brain, unconditional	0	0.5
	Melanoma, conditional	0	0
	Melanoma, unconditional	0.07	—
MDLM (ours)	Fly Brain, conditional	0.3	1
	Fly Brain, unconditional	0.3	1
	Melanoma, conditional	0.15	1
	Melanoma, unconditional	0.3	—
Simple baselines	Fly Brain	—	3
	Melanoma	—	2
Corrector onset threshold		0 (active from the first reverse step)	
Sampling NFE		5	

Table 6: Computational overhead of our non-Markovian SLM compared with the standard Markovian SLM. Training results correspond to the average case ($M = 499$). MACs are reported for the forward pass.

Metric	**Markovian SLM**	**Ours**	**Relative**
MACs (GMAC) ↓	95.08	95.21	1.0014×
Training time (ms/step) ↓	19.67	22.69	1.15×
Inference time (ms/batch) ↓	69.66	76.28	1.10×
Training peak memory (MB) ↓	1314	1803	1.37×
Inference peak memory (MB) ↓	339.1	347.3	1.02×

Scaling. The overhead scales predictably with the number of conditioning history states M (we write M for the count of past states routed over, to avoid clashing with the alphabet size K). Training step time is linear in M at $5.36\,\mu$s per state ($B = 64$, $L = 500$), so the $M = 0, 99, 499, 999$ configurations take 19.9, 20.9, 22.7, and 25.3 ms respectively. Peak training memory follows baseline $+\,2\times$ the view buffer, where the factor of two comes from the construction pipeline holding the noise draw, the samples, and the pre- and post-normalised views resident simultaneously rather than from storing the buffer twice. At sampling the buffer is $B \cdot \text{steps} \cdot L \cdot 16$ bytes, so the memory overhead grows with the step count: negligible at 5–10 steps and around 0.5 GB at 1000 steps. The arithmetic itself is minor throughout: the router's added MACs are 0.14% of the denoiser, so the wall-clock overhead is dominated by memory bandwidth and kernel launches rather than computation.

F Datasets and Splits

We inherit the exact datasets, train/validation/test splits, preprocessing, evaluator classifiers, and metric definitions of Stärk et al. (2024) for DNA.

Promoter (conditional). The human promoter dataset of Avdeyev et al. (2023): $100{,}251$ sequences of length $L = 1024$ bp paired with CAGE transcription profiles, split $88{,}470/3{,}933/7{,}497$ (train/val/test). The task generates a promoter conditioned on a target profile; SP-MSE is the mean squared error between the Sei-predicted (Chen et al., 2022) and target profiles.

Enhancer (unconditional and cell-type CFG). Two datasets of length $L = 500$ bp: human **melanoma** (Kalender Atak et al., 2021), $88{,}870$ sequences, $K = 47$ cell types, split

Table 7: SLM bias ablation on Melanoma ($T = 5$).

Bias	Re-expand (%)	Recons. (%)	FBD ↓
0.00	27.26	10.65	6.0965
0.05	26.77	12.17	5.3229
0.07	26.09	12.67	**5.3095**
0.10	24.66	13.27	5.5253
0.15	21.65	13.75	6.4845
0.20	18.22	13.54	8.7542

Table 8: Corruption and history ablation on conditional Fly Brain ($T = 5$). "With history" uses trajectory history; "No history" denotes the uniform all-mask setting. We picked the best checkpoint for the single run.

Corruption	History	FBD ↓
Trajectory	With history	3.6861
Trajectory	No history (all-mask)	3.7405
Independent	With history	**1.5431**
Independent	No history (all-mask)	11.1326

Table 9: Motif-profile agreement between generated and real unconditional FlyBrain enhancer sequences using the SLM-based models with 5 generation steps. Results are reported as mean ± s.d. over three seeds. Higher Pearson and Spearman correlations indicate better profile agreement, while lower mean absolute $\log_2$ fold-change indicates motif frequencies closer to those of the real data.

Set	Pearson ↑	Spearman ↑	Mean $\lvert \log_2 \mathrm{FC} \rvert$ ↓
SLM	0.9180 ± 0.0020	0.9590 ± 0.0010	0.3747 ± 0.0133
No history	0.8563 ± 0.0196	0.9323 ± 0.0045	0.6213 ± 0.0107
Non-Markovian (ours)	$\mathbf{0.9697 \pm 0.0057}$	$\mathbf{0.9837 \pm 0.0015}$	$\mathbf{0.2057 \pm 0.0115}$

$70{,}892/8{,}966/9{,}012$; and **fly brain** (Janssens et al., 2021), $104{,}665$ sequences, $K = 81$ cell types, split $83{,}726/10{,}505/10{,}434$. FBD is the discrete analogue of FID in the feature space of a classifier trained on the corresponding cell-type labels; each dataset is evaluated both unconditionally and with cell-type CFG.

G Baseline Details and Attribution of Numbers

Within each pair, we use the same backbone, data, and training budget; TC-NMD adds independent corruption, routed history conditioning, and a biased inference schedule. The tables identify our reproduced results and their sampling steps; results quoted from prior work use their reported operating points. To keep the comparison unambiguous we distinguish results we produce from results quoted from prior work. The models are trained and evaluated under an identical pipeline and reported at 5 network function evaluations (NFE). All other baselines are quoted from Stärk et al. (2024) and Song et al. (2025) at their reported operating points (typically 100 NFE), as marked in each table caption. All DNA methods use identical evaluator classifiers and an identical number of generated sequences per run.

H Additional Ablations

H.1 Additional Metrics

This subsection reports the full numerical results summarised in Figure 5, including the metrics omitted from the figure.

H.2 Motif Raw Counts

Table 9 reports profile agreement; for reference, the underlying raw counts are 22.6 total motif hits per real sequence (15.3 distinct motifs present), against 21.5 / 14.8 for the non-Markovian model, 21.1 / 14.5 for the Markovian sampler, and 21.5 / 14.8 for the single-view baseline. The dinucleotide-shuffled floor produces *more* raw hits than real (26.0 / 17.6), because the degenerate motifs in this library are largely determined by dinucleotide composition, which the shuffle preserves. This is

Table 10: Reproduction check for the Markovian SLM baseline at its reported operating point. We re-evaluate our SLM reproduction at 100 NFE on Fly Brain (FB) and Human Melanoma (MEL), using the same checkpoints, evaluator classifiers, and test splits as Table **??**. FBD (↓), mean ± s.d. over three seeds. The 5-NFE rows are repeated from Table **??** for reference. The baseline improves by an order of magnitude when given 100 steps, confirming that our 5-NFE reproduction is a correctly configured SLM

Model	**NFE**	**FB** FBD ↓	**MEL** FBD ↓
SLM + CFG	5	14.65 ± 0.16	9.95 ± 0.31
SLM + CFG	100	0.57 ± 0.08	1.21 ± 0.06
SLM	5	19.57 ± 0.52	10.11 ± 1.08
SLM	100	3.10 ± 0.20	2.82 ± 0.31

why the comparison is framed as profile agreement rather than motif abundance: raw counts do not separate the methods from the floor, whereas the profile correlations do.

H.3 SLM IMPLEMENTATION VERIFICATION

I LIMITATIONS

Our approach has two limitations. The re-corruption bias is tuned for each dataset and backbone; adapting it to the denoiser's uncertainty could reduce this tuning. The router assigns one weight to each past state across sequence positions, so it cannot select different views for different motifs or neighbourhoods. Context-aware routing is a natural extension.