LEARNING FROM CONTEXT: LEAVE-ONE-OUT PREDICTION IN JUMP DIFFUSIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Continuous diffusion models generate discrete data through stochastic dynamics in an embedding space. We consider jump diffusions that additionally reset coordinates to embeddings of categorical values. A coordinate's own corrupted state can make prediction easy, potentially reducing the pressure to learn dependencies between coordinates. We study leave-one-out (LOO) prediction, in which each coordinate is predicted from the corrupted states of the other coordinates. We show that, under stated assumptions, jump diffusions with complete coordinate resets and state-independent clocks preserve the prescribed marginals when reset destinations use exact LOO probabilities and the drift uses the full endpoint posterior. We further show that this drift can be constructed from LOO probabilities by analytic likelihood completion without additional network evaluations. We evaluate unrestricted, LOO, and completed predictions on Text8 character generation and ZINC250k molecular graph generation. Compared with unrestricted prediction, LOO improves lexical statistics on Text8 and the fraction of valid, connected molecules for absorbing jump diffusion on ZINC250k. For molecular pure diffusion, completion outperforms using LOO probabilities directly in the drift.

1 INTRODUCTION

Diffusion and flow matching generate discrete sequences either directly in token space or through continuous representations. Masked diffusion and discrete flow matching provide effective approaches to language modeling (Sahoo et al., 2024; Shi et al., 2024; Gat et al., 2024). Continuous approaches instead use token embeddings and continuous processes in the embedding space (Li et al., 2022; Dieleman et al., 2022; Gulrajani & Hashimoto, 2023). Recent models report competitive language-modeling performance for the latter approach (Chen et al., 2026; Yang et al., 2026). Another recent approach uses processes with discrete jumps in the embedding space (Jutras-Dubé et al., 2026). We investigate how the information available to individual predictions affects learning in the embedding space.

For masked discrete flow matching or diffusion, a masked position gives no direct information about its target token. A noisy embedding however can supply substantial evidence, especially near the data endpoint. CANDI analyzes a related tension: noise levels that preserve identifiable context can make continuous denoising trivial, while stronger noise can destroy useful information (Pynadath et al., 2026). This motivates our hypothesis that easy local prediction can reduce the pressure to learn dependencies between positions. In a small four-token parity experiment we illustrate this possibility: low prediction loss can precede learning the dependence needed for joint generation (Table 1). We therefore investigate leave-one-out (LOO) prediction, in which each token is predicted from the noisy states at all other positions. Let $Y = (Y_1, \dots, Y_L)$ be a sequence of L tokens and $X_t = (X_{1,t}, \dots, X_{L,t})$ its corrupted token embeddings at time t. Write $X_{-i,t}$ for all coordinates except i, and similarly for x_{-i}. For a token a, the full and LOO posteriors at position i are

$$p_{i,t}(a \mid x) = \mathbb{P}(Y_i = a \mid X_t = x),$$
$$q^{\star}_{i,t}(a \mid x_{-i}) = \mathbb{P}(Y_i = a \mid X_{-i,t} = x_{-i}).$$

Predicting $q^{\star}_{i,t}$ requires inferring Y_i without access to $X_{i,t}$. We implement this restriction by adapting T-TA's input-isolated attention architecture (Shin et al., 2020) to noisy token embeddings and time conditioning.

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.

Excluding an input coordinate from its own prediction has precedents. Blind-spot denoising imposes this restriction for self-supervised image restoration (Batson & Royer, 2019; Laine et al., 2019). In discrete diffusion, Sun et al. (2023) introduce hollow Transformers and evaluate them under uniform corruption, while Zhao et al. (2025) use related architectures for informed correction in masked diffusion. Gourevitch et al. (2026) identify the leave-one-out posterior as the target of the plug-in ELBO for uniform diffusion and derive conversions between this posterior, the denoiser, and the score. Casado Noguerales et al. (2026) derive these conversions for general token-factorizing discrete corruption using the cavity posterior, which excludes the local corrupted token. Dmitriev et al. (2026) analyze LOO-based samplers for uniform and remasking diffusion, with guarantees that depend on the target's dependence structure. These works establish LOO prediction for dynamics on discrete token states. We investigate its use in embedding-space processes that also evolve continuously between jumps. In this setting, jumps reset individual positions to token embeddings. Related hybrid methods combine discrete states with continuous representations, including CADD, CANDI, and CCDD (Zheng et al., 2026; Pynadath et al., 2026; Zhou et al., 2026). Sticky Jump Diffusions combine continuous evolution with absorbing jumps to token anchors and allow cross-position blending in the corruption process (Jutras-Dubé et al., 2026). Our construction uses an endpoint-directed bridge with repeated resets: continuous motion resumes after each reset, and the destination predictor excludes the current local state.

Using LOO prediction in these dynamics need not preserve the prescribed marginal path. We investigate when this path, and hence its data endpoint, is preserved. Cross-entropy training with LOO inputs targets the token distribution conditioned on the other positions, whereas posterior averaging of conditional generators uses the token distribution conditioned on the full current state (Holderrieth et al., 2025). These quantities are generally different and only for the latter the correct marginal probability path is established. We give a counterexample showing that substituting the leave-one-out posterior into the continuous drift can change the prescribed marginal path (Example B.2). Our construction instead retains the drift based on the full current state and substitutes LOO probabilities only for the jump part of the process. We show that this implicates the correct marginal probability path, though trajectory laws may differ (Proposition 3.2). We use input-isolated attention to learn LOO probabilities and analytic likelihood completion to obtain full-posterior estimates for the drift. Combining contextual predictions with local likelihoods is established in blind-spot denoising (Laine et al., 2019), Plaid's output parameterization (Gulrajani & Hashimoto, 2023, Section 3.3), and cavity-to-denoiser conversions for discrete corruption (Casado Noguerales et al., 2026). Our contribution concerns their use in the generator: for complete coordinate resets with state-independent clocks, LOO destination probabilities and full-posterior drift can jointly preserve the prescribed marginal path. This permits the reset predictor to exclude local information while retaining the marginal-preservation property of the continuous dynamics.

Our Text8 experiments compare unrestricted prediction, raw LOO prediction—used directly in the drift, without likelihood completion, and analytic likelihood completion for pure diffusion and jump diffusion, with masked discrete flow matching and embedded pure jumps as baselines. We match training and sampling budgets and study jump sampling rules and time schedules.

Our contributions are:

- We establish marginal preservation for our repeated-reset jump diffusion process with exact LOO reset predictions and full-posterior drift, under mild assumptions. A counterexample shows that the analogous substitution in the drift can fail.
- We use this construction with a Brownian bridge with repeated resets whose marginals converge to the joint data law. A single network supplies LOO reset jump predictions and, through analytic likelihood completion, full-posterior estimates for the drift.
- In matched Text8 experiments, LOO prediction improves the reported generation metrics over unrestricted prediction. We examine conditional paths, jump sampling, and schedules, and show that raw LOO drift improves validity and four-gram agreement over analytic completion despite falling outside our guarantee. On ZINC250k, LOO prediction increases the fraction of valid, connected molecules relative to unrestricted prediction for absorbing jump diffusion. For pure diffusion, analytic likelihood completion improves validity and distributional agreement over raw LOO drift.

2 CONDITIONAL PATHS AND LEAVE-ONE-OUT PREDICTION

Let $\mathcal{V}$ be a finite vocabulary and ler $p_{\text{data}} \in \mathcal{P}(\mathcal{V}^L)$ a data distribution for L the sequence length. Choose a random variable $Y = (Y_1, \dots, Y_L) \sim p_{\text{data}}$, and consider an embedding $e : \mathcal{V} \to \mathbb{R}^d$. For recovery of token sequences, we assume that e is injective. Write $E(Y) = (e(Y_1), \dots, e(Y_L))$ and for $x = (x_1, \dots, x_L) \in \mathbb{R}^{dL}$ let $x^{i \leftarrow z}$ denote x with position i replaced by z i.e. $x^{i \leftarrow z} = (x_1, \dots, x_{i-1}, z, \dots, x_L)$. We write $x_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_L)$ for the sequence with position i omitted, and use $X_{-i,t}$ analogously. Time runs from the source at $t = 0$ toward data at $t = 1$. For each process below, write $p_t = \mathcal{L}(X_t)$ for its state law and $p_{i,t}(a \mid x) = \mathbb{P}(Y_i = a \mid X_t = x)$ for the endpoint posterior at position i.

2.1 DISCRETE FLOW MATCHING

Discrete flow matching (DFM) uses a probability path on token sequences and learns a jump rate kernel that generates its marginals (Gat et al., 2024). Appendix A.1 summarizes the terminology. For the masked path, let $\mathsf{m} \notin \mathcal{V}$ be a mask and we consider the following discrete process $X_t^y \in (\mathcal{V} \cup \{\mathsf{m}\})^L$ starting from $X_0^y = (\mathsf{m}, \dots, \mathsf{m})$. Let $\kappa : [0, 1] \to [0, 1]$ be continuous and nondecreasing, continuously differentiable on $[0, 1)$, with $\kappa(0) = 0$, $\kappa(1) = 1$, and $\kappa(t) < 1$ for $t < 1$. For a fixed endpoint y, take independent reveal times $T_1, \dots, T_L$ with $\mathbb{P}(T_i \le t) = \kappa(t)$ and define the conditional process $X_t^y = (X_{1,t}^y, \dots, X_{L,t}^y)$ by setting $X_{i,t}^y = \mathsf{m}$ for $t < T_i$ and $X_{i,t}^y = y_i$ for $t \ge T_i$. Each position is therefore revealed independently with probability $\kappa(t)$ at time t, so

$$\mathbb{P}(X_t^y = x) = \prod_{i=1}^{L} \left[(1 - \kappa(t)) \mathbf{1}\{x_i = \mathsf{m}\} + \kappa(t) \mathbf{1}\{x_i = y_i\} \right]. \tag{1}$$

Since $\kappa(1) = 1$, $X_1^y = y$ almost surely. Hence the mixture $p_t = \mathcal{L}(X_t^Y)$ satisfies $p_1 = p_{\text{data}}$. This conditional process reveals y_i at rate $\kappa'(t)/(1 - \kappa(t))$ while position i is masked and then leaves it fixed. Gat et al. (2024) obtain the jump rate kernel for the marginal process X_t with law p_t through the per-position rates

$$r_{i,t}(x_i \to a \mid x) = \mathbf{1}\{x_i = \mathsf{m}\} \frac{\kappa'(t)}{1 - \kappa(t)} p_{i,t}(a \mid x), \qquad a \in \mathcal{V}. \tag{2}$$

Thus generation requires $p_{i,t}(a \mid x)$, a categorical endpoint posterior at each position, conditioned on the entire current sequence. These posteriors can be learned by cross-entropy on samples of (1) (Gat et al., 2024; Sahoo et al., 2024). At an masked position, the local mask provides no evidence about its target token.

2.2 CONTINUOUS EMBEDDINGS AND ENDPOINT POSTERIORS

We briefly review continuous diffusion for categorical data using endpoint-posterior prediction (Dieleman et al., 2022) and posterior averaging of conditional generators (Holderrieth et al., 2025). We illustrate these established ideas with Brownian bridges (Liu et al., 2022), which also provide the continuous component of our later construction. For each endpoint $y \in \mathcal{V}^L$, consider Brownian bridges $X_t^y = (X_{1,t}^y, \dots, X_{L,t}^y)$ satisfying, for $0 \le t < 1$,

$$dX_{i,t}^y = \frac{e(y_i) - X_{i,t}^y}{1 - t}\, dt + \sigma_d\, dW_{i,t}, \qquad X_{i,0}^y \sim \mathcal{N}(0, \sigma_s^2 I_d), \quad \sigma_d > 0, \sigma_s \ge 0$$

These paths extend continuously to $t = 1$ with $X_1^y = E(y)$ almost surely. Let $p_t = \mathcal{L}(X_t^Y)$ for $Y \sim p_{\text{data}}$. Posterior averaging gives the marginal SDE $dX_{i,t} = b_{i,t}(X_t)\, dt + \sigma_d\, dW_{i,t}$, with $X_0 \sim \mathcal{N}(0, \sigma_s^2 I_{dL})$, $X_1 \sim \mathcal{L}(E(Y))$ and

$$b_{i,t}(x) = \frac{\mathbb{E}[e(Y_i) \mid X_t^Y = x] - x_i}{1 - t} = \frac{\sum_{a \in \mathcal{V}} p_{i,t}(a \mid x) e(a) - x_i}{1 - t}.$$

Under the usual existence and uniqueness assumptions for the forward equation, the resulting process has law p_t at each time t. We can learn $p_{i,t}(a \mid x)$ with the loss

$$\mathcal{L}_{\text{CE}}(\theta) = \mathbb{E}_{t, Y, X_t^Y} \left[-\sum_i \log p_{\theta, i, t}(Y_i \mid X_t^Y) \right].$$

2.3 AN ENDPOINT-DIRECTED BRIDGE WITH REPEATED RESETS

For a fixed endpoint y, consider the conditional jump diffusion $X_t^y = (X_{1,t}^y, \dots, X_{L,t}^y)$ given by

$$dX_{i,t}^y = \frac{e(y_i) - X_{i,t-}^y}{1-t}\,dt + \sigma_d\,dW_{i,t} + (e(y_i) - X_{i,t-}^y)\,dN_{i,t}, \qquad X_{i,0}^y \sim \mathcal{N}(0, \sigma_s^2 I_d). \tag{3}$$

Each coordinate follows Brownian bridge dynamics between jumps, resets exactly to $e(y_i)$ at every jump, and resumes diffusing immediately afterwards. Here N_i are independent Poisson counting processes with a nonnegative deterministic rate $\lambda(t)$ satisfying $\int_0^t \lambda(s)\,ds < \infty$ for every $t < 1$. The notation $X_{i,t-}^y$ denotes the state just before time t. For example, $\lambda_\gamma(t) = \gamma t^{\gamma-1}/(1-t^\gamma)$, $\gamma \in \{1, 2, 4\}$, gives first-reset probability $\mathbb{P}(N_{i,t} \geq 1) = t^\gamma$. Figure 1 illustrates one choice; Appendix A.2 gives the conditional transition laws.

Remark 2.1 (Bridge clock). We write (3) and the derivations below for $\kappa(t) = t$. With a continuous bridge clock $u = \kappa(t)$, the drift factor becomes $\kappa'(t)/(1-\kappa(t))$ and the diffusion coefficient becomes $\sigma_d\sqrt{\kappa'(t)}$. The reset rate $\lambda(t)$ is specified independently in physical time. The Text8 and ZINC250k conditional-path comparisons use $\kappa(t) = t^2$, giving drift factor $2t/(1-t^2)$ and diffusion coefficient $\sigma_d\sqrt{2t}$; the jump-schedule comparisons vary only $\lambda_\gamma(t)$. Training states and analytic likelihoods use the conditional laws for the same clock.

Sampling $Y \sim p_{\text{data}}$ defines the mixture X_t^Y and its marginal curve $p_t = \mathcal{L}(X_t^Y)$, with $p_0 = \mathcal{N}(0, \sigma_s^2 I_{dL})$. Posterior averaging (Holderrieth et al., 2025, Proposition 1) gives the following operator, whose weak forward equation is satisfied by p_t (proved directly in Appendix B.2):

$$\begin{aligned}\mathcal{A}_t f(x) = \sum_i \left[\frac{\sum_{a\in\mathcal{V}} \mathbb{P}(Y_i = a \mid X_t^Y = x)e(a) - x_i}{1-t} \cdot \nabla_i f(x) + \frac{\sigma_d^2}{2}\Delta_i f(x)\right] \\ + \lambda(t)\sum_{i,a} \mathbb{P}(Y_i = a \mid X_t^Y = x)\left[f(x^{i\leftarrow e(a)}) - f(x)\right].\end{aligned} \tag{4}$$

Here ∇_i and Δ_i are the gradient and Laplacian with respect to x_i. The conditional paths concentrate on their endpoints as $t \uparrow 1$, so the marginal curve extends to $p_1 = \mathcal{L}(E(Y))$, or $p_1 = p_{\text{data}}$ when identifying tokens with their embeddings.

The only unknown quantities in (4) are $p_{i,t}(a \mid x)$, which determine both the drift and the reset probabilities. We therefore learn $p_{\theta,i,t}$ from conditional-path samples using

$$\mathcal{L}_{\text{full}}(\theta) = \mathbb{E}_{t,Y,X_t^Y}\left[-\sum_i \log p_{\theta,i,t}(Y_i \mid X_t^Y)\right]. \tag{5}$$

Its unrestricted minimizer is $p_{i,t}$. Appendices B.1 and C give the posterior-learning argument and sampling details, respectively.

Remark 2.2 (Sampling training states). For $y_i = a$, write $p_{i,t}(\cdot \mid a) = \mathcal{L}(X_{i,t}^y)$ for the conditional state law; the argument order distinguishes it from the endpoint posterior $p_{i,t}(a \mid x)$. To estimate (5), draw t and $Y \sim p_{\text{data}}$, then sample the coordinates independently from $p_{i,t}(\cdot \mid Y_i)$. Each coordinate requires only the last reset time (or no reset) and a Gaussian draw. Appendix A.2 gives the mixture law, and Proposition A.2 gives its closed form when $\lambda(t) = \kappa'(t)/(1-\kappa(t))$.

Remark 2.3 (Local endpoint information). For uniform four-token even-parity data, Figure 2 shows that the current local state already reveals much of its endpoint: nearest-embedding decoding from $X_{i,t}$ alone is 93.9% accurate at $t = 0.2$ and 99.9% at $t = 0.5$. Thus, endpoint prediction can be easy without learning the dependence between positions required for joint generation.

3 LEARNING CONTEXTUAL RESETS

Figure 2 shows that the current local state can already reveal most of its endpoint token, making endpoint prediction easy without learning dependencies between positions. To remove this shortcut from the jump predictor, we exclude $X_{i,t}$ when predicting the reset destination at position i. This restriction changes the posterior learned by cross-entropy, so using it in the dynamics requires care. We show that it can be used while still preserving p_t, provided the continuous drift retains $p_{i,t}(\cdot \mid X_t)$.

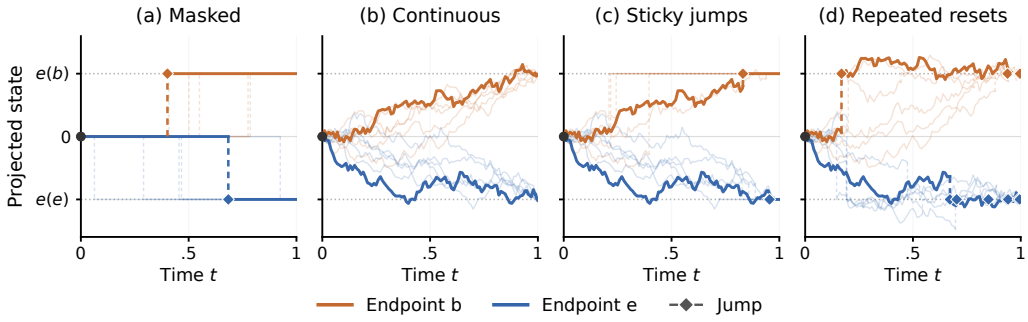

Figure 1: **Four paths, shown against time.** States are projected onto $e(b)$; colors distinguish the antipodal unit-circle endpoints $e(b)$ and $e(e)$. Dashed segments mark jumps. Masked diffusion reveals once; the schematic sticky jump diffusion (Jutras-Dubé et al., 2026) stays fixed after commitment. The illustrated repeated-reset trajectories use independent clocks with $\lambda(t) = (1-t)^{-1}$ and continue diffusing after each reset.

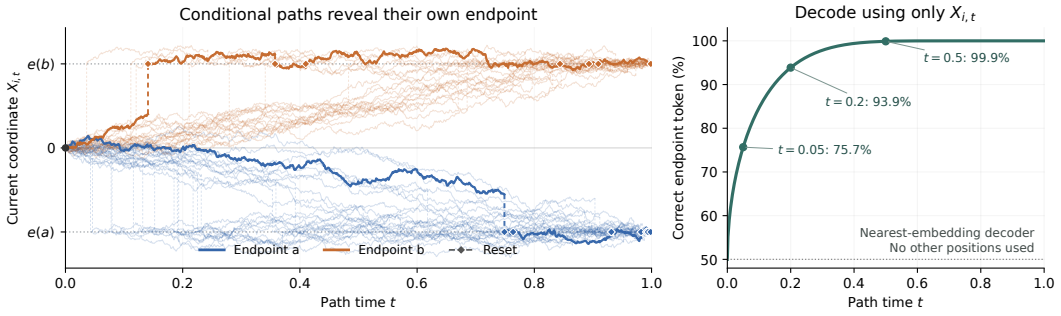

Figure 2: **Local information in conditional paths.** Repeated-reset bridges with $e(a) = -1$, $e(b) = 1$, $X_0 = 0$, $\sigma_d = 0.35$, and $\lambda(t) = (1-t)^{-1}$. Left: 48 coordinate paths, colored by endpoint; dashed segments indicate jumps, after which diffusion continues. Right: endpoint recovery using only the current local state.

3.1 Endpoint prediction without local information

Following contextual token prediction (Shin et al., 2020) and leave-one-out denoising in discrete diffusion (Gourevitch et al., 2026), we consider a leave-one-out (LOO) predictor at position i: a categorical kernel $q_{\theta,i,t}(a \mid x_{-i})$ invariant to every change of x_i. Given samples of the conditional path, we minimize

$$\mathcal{L}_{\mathrm{LOO}}(\theta) = \mathbb{E}_{t,Y,X_t}\left[-\sum_{i=1}^{L} \log q_{\theta,i,t}(Y_i \mid X_{-i,t})\right]. \tag{6}$$

Equation (12) gives their law for the chosen reset schedule.

Over all measurable categorical kernels, the minimizers of (6) are versions of

$$q^{\star}_{i,t}(a \mid x_{-i}) = \mathbb{P}(Y_i = a \mid X_{-i,t} = x_{-i}) \tag{7}$$

In general, $q^{\star}_{i,t}(\cdot \mid x_{-i}) \neq p_{i,t}(\cdot \mid x)$.

3.2 Preserving the marginal path

Remark 3.1. Replacing $p_{i,t}(\cdot \mid x)$ by $q^{\star}_{i,t}(\cdot \mid x_{-i})$ changes the bridge drift by

$$b^{\mathrm{LOO}}_{i,t}(x) - b_{i,t}(x) = \frac{\mathbb{E}[e(Y_i) \mid X_{-i,t} = x_{-i}] - \mathbb{E}[e(Y_i) \mid X_t = x]}{1-t}.$$

This substitution does not in general preserve the marginal path, even if the jump probabilities are left unchanged. In Example B.2 in Appendix B.3, the jump rate kernel is unchanged, but the second-moment equation for $X_{1,t}$ fails. Thus the discrepancy is caused by the drift alone.

Keeping the drift $b_{i,t}$ while using $q^\star_{i,t}$ for the jump part preserves p_t, although the resulting jump rate kernel differs from that obtained by generator matching. The trajectory laws can differ while the one-time laws remain p_t.

Proposition 3.2 (Drift $b_{i,t}$ and LOO jumps)**.** *For the conditional path* (3) *with any rate* $\lambda(t)$ *satisfying the conditions above, consider the generator*

$$\mathcal{A}^{\mathrm{ctx}}_t f(x) = \sum_i \left[\frac{\sum_a p_{i,t}(a \mid x) e(a) - x_i}{1-t} \cdot \nabla_i f(x) + \frac{\sigma_d^2}{2} \Delta_i f(x) \right] + \lambda(t) \sum_{i,a} q^\star_{i,t}(a \mid x_{-i}) \left[f(x^{i \leftarrow e(a)}) - f(x) \right] \tag{8}$$

Under the assumptions stated in Appendix B.2, the resulting process started from p_0 *has law* p_t *for every* $t < 1$*. Moreover,*

$$p_t \Rightarrow \mathcal{L}(E(Y)) \qquad \text{as } t \uparrow 1.$$

The precise statement and proof are given in Appendix B.2.

3.3 Input-isolated contextual prediction

We adapt the attention construction of T-TA (Shin et al., 2020) to noisy embeddings and path time. Each position has a source feature $s_j = \phi(x_j, j, t)$ depending only on its own input, position, and time. The sources are fixed across layers. A query $h_i^{(0)} = \psi(i,t)$ has no state input and is updated by

$$h_i^{(\ell+1)} = F_\ell\Big(h_i^{(\ell)}, \mathrm{Attn}_\ell(h_i^{(\ell)}, \{s_j : j \neq i\}, t), t\Big), \qquad q_{\theta,i,t} = \mathrm{softmax}(W_q h_i^{(D)} + b_q), \tag{9}$$

Here F_ℓ acts separately at each position; its residual and feed-forward structure is specified in Appendix G.1. Evolving query features never supply source keys or values.

For fixed parameters, $q_{\theta,i,t}$ in (9) is invariant by construction to changes of x_i, so $\partial q_{\theta,i,t}/\partial x_i = 0$ wherever differentiable.

Diagonal masking alone in ordinary self-attention does not ensure this property: another position can receive x_i and return it in a later layer. Fixed token-local sources prevent that route; SwinIA uses the same principle for blind-spot image denoising (Papkov et al., 2025). All L predictions are computed in one forward pass, with $O(L^2)$ attention cost per layer at fixed width. Our unmasked control uses the same fixed-source architecture and allows the additional source s_i. Another possibility is to target invariance to x_i through the loss: for uniform discrete diffusion, Gourevitch et al. (2026, Section 3.2) give objectives whose expected loss is minimized over unrestricted predictors by the LOO posterior, even when the predictor receives X_t. This invariance need not hold after finite training (Gourevitch et al., 2026, Appendix H).

3.4 Recovering the posterior for the drift

Proposition 3.2 justifies LOO reset probabilities. To obtain the posterior $p_{i,t}(a \mid x)$ required by the continuous drift, we apply analytic likelihood completion to the LOO predictions. For token-factorizing discrete corruption, the analogous cavity-to-denoiser conversion is given by Casado Noguerales et al. (2026, Eq. (42)).

The conditional paths (3) split into products i.e. $p_t(dx \mid y) = \prod_i p_{i,t}(dx_i \mid y_i)$. Let $\mu_{i,t}$ be such that $p_{i,t}(\cdot \mid a) \ll \mu_{i,t}$ is absolutely continuous for every a and set $\ell_{i,t}(x_i \mid a) = dp_{i,t}(\cdot \mid a)/d\mu_{i,t}(x_i)$. For example, $\mu_{i,t} = \sum_a p_{i,t}(\cdot \mid a)$ is a valid choice.

Proposition 3.3 (Likelihood completion)**.** *For conditional paths as in* (3)*,* $p_{i,t}$ *satisfies*

$$p_{i,t}(a \mid x) = \frac{q^\star_{i,t}(a \mid x_{-i}) \ell_{i,t}(x_i \mid a)}{\sum_b q^\star_{i,t}(b \mid x_{-i}) \ell_{i,t}(x_i \mid b)} \tag{10}$$

for p_t*-almost every* x *where the denominator is positive.*

The proof is given in Appendix B.4. Changing λ changes the training law and its local likelihood; the learned posteriors and likelihood completion must use that same law.

We insert the completed probabilities only into the drift of (8). The reset term continues to use $q_{\theta,i,t}$, so a single network supplies both prediction roles. Algorithms 1 and 2 in Appendix C summarize simulation-free training and numerical generation. An alternative with a learned drift predictor is described in Appendix H.2.

Remark 3.4 (Learning dependence between positions). On uniform four-token even-parity data, each token is marginally uniform but determined by the other three. After 250 updates, the full-state and local-only predictors have nearly identical loss and approximately 50% invalid generations (Table 1). LOO prediction with analytic drift has a larger loss but no invalid strings in the evaluated samples, indicating earlier learning of this dependence. The full-state predictor also learns the correct distribution by 3,000 updates, so the observed separation concerns learning speed. Appendix F gives the setup.

Table 1: Cross-entropy loss (CE) and percentage of invalid generated strings at step 250 on the four-token even-parity task. Results average three seeds, with 8,192 generated strings per seed.

Prediction target / method	Evaluation CE	Invalid strings
$p_{i,t}(\cdot \mid X_t)$	0.0876	49.1%
$\mathbb{P}(Y_i \in \cdot \mid X_{i,t})$	0.0876	49.5%
$q^{\star}_{i,t}(\cdot \mid X_{-i,t})$ + analytic drift	0.1677	0% observed
Masked flow matching	0.6286	1.2%

4 Numerical experiments

We study two forms of discrete generative modeling. Text8 (Mahoney, 2006) contains cleaned English text and tests generation of ordered character sequences, including word formation. ZINC250k (Sterling & Irwin, 2015; Jo et al., 2022) contains molecules represented as graphs with categorical atom and bond labels. Generating these graphs requires modeling atom–bond compatibility and connectivity while respecting permutation symmetry. Across both datasets, we compare unrestricted, LOO, and analytically completed predictions to assess how local information affects learning dependencies. Metric definitions and experimental protocols are in Appendices E, G, and I. Throughout the main comparisons, "unrestricted" denotes access to the target coordinate within the same fixed-source architecture used by the LOO predictor. These matched controls isolate the restriction on input access within that architecture.

4.1 Text8 experiments

Repeated resets: unrestricted, LOO, and completed predictions. Repeated-reset jump diffusion (RJD) follows (3) with $(\sigma_s, \sigma_d) = (0, 1)$ and bridge clock $\kappa(t) = t^2$. Its physical-time reset rate is $\lambda(t) = \kappa'(t)/(1 - \kappa(t)) = 2t/(1 - t^2)$. Continuous bridge motion resumes after every reset to a token embedding, and further resets remain possible.

We compare three prediction choices. RJD uses the unrestricted predictor $p_{\theta,i,t}$ for both drift and reset probabilities. RJD-LOO uses $q_{\theta,i,t}$ for both. RJD-BC-D applies Bayesian completion (BC), (10), only to the drift and retains $q_{\theta,i,t}$ for resets. The last two variants reuse the same LOO checkpoints, with one backbone evaluation per interval.

RJD-LOO improves all four metrics over unrestricted prediction (Table 2). Compared with completion, raw LOO drift gives higher validity and lower JSD_4, while completion gives higher D_5 and E_5. Each ordering holds in all three seeds. RJD-BC-D implements the construction justified by Proposition 3.2 under exact predictions and dynamics.

Pure diffusion: LOO variants and unrestricted control. Our pure-diffusion variants use RJD's time-changed Brownian bridge without jumps. PD-LOO uses $q_{\theta,i,t}$ for the drift, and PD-BC uses its likelihood-completed prediction. PD is the unrestricted control, with drift prediction $p_{\theta,i,t}$.

For pure diffusion, LOO improves all four metrics over both unrestricted prediction and completion. PD-LOO has the highest validity; RJD-LOO has higher D_5 and E_5 and lower JSD_4 than PD-LOO.

Comparison methods. Our comparison methods are masked discrete flow matching (DFM) (Gat et al., 2024) and embedded pure jumps (EPJ). DFM reveals a token through $\mathsf{m} \to A_i$, whereas EPJ uses $0 \to e(A_i)$; both use $q_{\theta,i,t}$ and the same backbone after the input encoder. Each position jumps once and then remains fixed, with reveal probability $\kappa(t) = t^2$.

DFM and EPJ give similar results in Table 2. DFM has the lowest mean JSD_4 despite much lower validity, so four-gram agreement alone does not capture lexical quality. A paired seed-0 control improves EPJ performance with ordinary self-attention, indicating that the input-isolated architecture can restrict this baseline even though unresolved positions contain no local token information. The AJD prediction comparison and the control's full sampling sweep use 1,024 samples per seed and are reported in Appendices G.4 and H.1.

Table 2: Text8 at 128 NFE and 4,096 samples per seed (3 seeds per experiment). Mean $\pm$ sample standard deviation over three training seeds; D_5 and E_5 are rounded to integers. All paths use $\kappa(t) = t^2$ and the cosine grid. Jumps use CAT except in the TOPP row; drift predictions have temperature one. All rows share the validation reference in Appendix E. Bold marks the best mean in each column.

Method	V_5 (%) ↑	D_5 ↑	E_5 ↑	JSD_4 ↓
Repeated resets: our variants and unrestricted control				
RJD	28.96 ± 0.62	3051 ± 43	911 ± 21	0.0657 ± 0.0007
RJD-LOO	75.60 ± 0.89	6596 ± 66	2411 ± 48	0.0580 ± 0.0011
RJD-BC-D	60.16 ± 1.18	$\mathbf{6783} \pm 13$	$\mathbf{2843} \pm 54$	0.0672 ± 0.0015
RJD-LOO, TOPP	79.70 ± 0.68	6611 ± 68	2368 ± 29	0.0545 ± 0.0009
Pure diffusion: our variants and unrestricted control				
PD	29.47 ± 1.49	3137 ± 191	964 ± 83	0.0664 ± 0.0005
PD-LOO	$\mathbf{85.49} \pm 0.67$	5696 ± 58	1944 ± 36	0.0653 ± 0.0012
PD-BC	36.34 ± 0.51	4499 ± 7	1637 ± 23	0.0691 ± 0.0004
Comparison methods				
DFM	44.62 ± 0.56	4798 ± 159	1501 ± 150	$\mathbf{0.0536} \pm 0.0002$
EPJ	44.96 ± 0.47	4971 ± 75	1703 ± 73	0.0540 ± 0.0002

Sampling rules and network budgets. For fixed RJD-LOO checkpoints, we compare categorical sampling $A_i \sim q_{\theta,i,t}$ (CAT), $A_i = \arg\max_a q_{\theta,i,t}(a)$ (MAX), probabilities proportional to $q_{\theta,i,t}(a)^{1/0.7}$ (TEMP), and categorical sampling restricted to the smallest set of highest-probability tokens with total mass at least 0.9 (TOPP, or nucleus sampling) (Holtzman et al., 2020, Sections 3.1 and 3.3). Confidence selection (CONF) reassigns the positions selected by the independent clocks to those with largest $\max_a q_{\theta,i,t}(a)$, then samples destinations categorically. This adapts confidence-based position selection from Mask-Predict (Ghazvininejad et al., 2019) and MaskGIT (Chang et al., 2022). All rules retain the temperature-one raw LOO drift. The modified rules change the generator; their implementation and full comparisons are in Appendix G.5.

At 4,096 samples per seed, TOPP raises RJD-LOO validity from 75.60% to 79.70% and lowers JSD_4 from 0.0580 to 0.0545, while D_5 stays similar and E_5 decreases slightly (Table 2). Compared with PD-LOO, it has lower validity but higher D_5 and E_5 and lower divergence.

The full five-rule study uses 1,024 samples per seed (Appendix G.5). Within that study, TEMP and TOPP improve validity and four-gram agreement over CAT while retaining similar D_5; MAX and CONF increase validity at a cost in word diversity. Appendix G.6 reports additional NFE budgets; Appendix G.8 evaluates other samplers with 4,096 samples per seed.

CADD-derived training. Using the LOO architecture with a CADD-derived training recipe (Zheng et al., 2026), one token predictor supplies both continuous and discrete updates. In a paired seed-0 comparison, control and LOO differ only in the attention mask. With posterior-mean (soft) endpoints, LOO improves word validity, distinct valid-word counts, and four-gram agreement at 128 NFE with CAT; with argmax (hard) endpoints, it increases validity but worsens divergence. Appendix H.4 gives the configurations and complete sweep, see Table 14.

4.2 ZINC250K EXPERIMENTS

We evaluate ZINC250k (Sterling & Irwin, 2015) using the GDSS representation and test split (Jo et al., 2022). A shared permutation-equivariant predictor with 6.92M parameters processes atom and bond coordinates. We compare EPJ, PD, and absorbing jump diffusion (AJD). Each AJD coordinate follows the same continuous bridge as RJD until its first jump to an embedding, then remains fixed (Appendix D). For EPJ we train the unrestricted predictor $p_{\theta,i,t}$ and for PD and AJD both, $p_{\theta,i,t}$ and the LOO predictor $q_{\theta,i,t}$. All five models receive 40,000 updates for each of two training seeds. Table 3 evaluates EMA weights with 10,000 generated graphs per seed, 128 NFE, and all 24,887 test references. Appendix I gives the protocol and other NFE budgets. The two-seed results show that the effect of raw LOO drift depends on the conditional path. For AJD, raw LOO drift increases the fraction of valid and connected outputs over unrestricted prediction and achieves the lowest mean raw and corrected FCD. Completing its drift reduces connected validity but improves NSPDK agreement. For PD, completion improves raw validity, connected validity, raw FCD, and NSPDK over raw LOO drift; corrected FCD improves only slightly on average. PD-BC has the highest mean raw validity, whereas AJD-LOO has the highest mean fraction of valid and connected molecules.

Table 3: ZINC250k test results at 40,000 updates and 128 NFE: mean $\pm$ sample standard deviation over training seeds 0 and 1, with 10,000 generated graphs per seed. All jumps use CAT; BC changes only the drift. Validity percentages use all generated graphs before correction. Raw distributional metrics include valid disconnected outputs; corrected FCD uses bond repair and largest-component selection. NSPDK is squared MMD (Appendix I.3). Bold marks the best mean per column.

Method	Raw valid (%) ↑	Raw valid & connected (%) ↑	Raw FCD ↓	Corrected FCD ↓	Raw NSPDK ↓
EPJ	69.02 ± 0.22	41.81 ± 0.11	21.190 ± 0.039	21.305 ± 0.002	0.04882 ± 0.00088
PD	89.98 ± 0.33	49.97 ± 0.37	21.919 ± 0.031	21.740 ± 0.003	0.05154 ± 0.00015
PD-LOO	55.38 ± 5.42	20.07 ± 3.91	24.966 ± 1.164	21.670 ± 1.250	0.15202 ± 0.02161
PD-BC	$\mathbf{91.84} \pm 0.40$	54.19 ± 2.21	21.642 ± 0.066	21.535 ± 0.090	0.05149 ± 0.00088
AJD	87.41 ± 0.95	47.82 ± 2.49	22.065 ± 0.088	21.943 ± 0.082	0.05054 ± 0.00043
AJD-LOO	83.96 ± 0.52	$\mathbf{71.07} \pm 0.61$	$\mathbf{20.460} \pm 0.019$	$\mathbf{20.968} \pm 0.015$	0.05141 ± 0.00011
AJD-BC-D	78.53 ± 0.16	59.11 ± 0.38	20.677 ± 0.090	20.990 ± 0.053	$\mathbf{0.04629} \pm 0.00003$

5 DISCUSSION

In repeated-reset jump diffusion, reset destinations can be predicted from $X_{-i,t}$ while the drift uses the posterior conditioned on X_t. Replacing the reset probabilities by $q^{\star}_{i,t}$ preserves the prescribed marginals when the drift retains $p_{i,t}$. Input-isolated attention excludes $X_{i,t}$ from the prediction at position i. Analytic likelihood completion then combines this prediction with the conditional likelihood of $X_{i,t}$ to estimate $p_{i,t}$. A single network can therefore supply both predictions without an additional network evaluation.

The experiments show that excluding a coordinate from its own prediction can improve generation under matched training budgets. LOO prediction improves lexical statistics on Text8 and increases the fraction of valid, connected molecules for absorbing jump diffusion on ZINC250k. The raw and completed variants share the same trained LOO predictor and differ only in whether the drift uses $q_{\theta,i,t}$ directly or its likelihood-completed distribution. Completion improves molecular pure diffusion, whereas raw LOO drift gives higher Text8 validity and better four-gram agreement. For repeated resets, completion also produces greater valid-word diversity. Thus the same trained predictor can yield different generation tradeoffs depending on the probabilities used in the drift.

An open question is why using $q_{\theta,i,t}$ directly in the drift improves some generation metrics, although the marginal-preservation result uses $p_{i,t}$. Addressing this question requires understanding how prediction errors, finite sampling intervals, and terminal decoding affect the generated distribution. Comparing exact and learned predictions on distributions whose probabilities can be computed explicitly would help separate these effects. Evaluating longer-range text dependencies and repeated resets on molecular graphs would further test whether the observed improvements extend beyond lexical statistics and absorbing-jump generation.

AI USE STATEMENT

In this work, we used generative AI tools to assist with data preparation, interpretation of experimental results, elaboration of author-provided research ideas, and writing and completing proofs based on author-provided proof ideas. Additionally, we used these tools to assist with preparing figures and drafting and revising manuscript text. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies generative modeling through theoretical analysis, experiments on the existing Text8 and ZINC250k benchmarks, and synthetic stochastic-process diagnostics. It does not involve human-subject experiments. As with other text-generation methods, models developed using this approach may reproduce biases or memorized content from their training data and could be used to generate misleading or harmful text.

REPRODUCIBILITY STATEMENT

Sections 2 and 3 specify the conditional paths, learning objectives, and theoretical assumptions, with supporting proofs and derivations in Appendices A and B.1. Appendix G.1 describes the LOO architecture, and Appendix C gives the numerical interval updates. Section 4 presents the main numerical comparisons. Appendix G gives the Text8 training and sampling protocol, Appendix H describes the additional predictor configurations, and Appendix I gives the molecular protocol. Appendix E defines the Text8 metrics. The main Text8 comparisons report three training seeds. The 40,000-update ZINC250k tables report means and sample standard deviations over two training seeds at matched training and NFE budgets. Molecular test results and the separate development sampling sweep use their stated sample counts and references. Section 5 discusses the limitations of the empirical evidence. We provide an anonymized code archive, including experiment configurations and reproduction instructions, in the supplementary material.

REFERENCES

Joshua Batson and Loic Royer. Noise2Self: Blind denoising by self-supervision. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 524–533, 2019. URL https://proceedings.mlr.press/v97/batson19a.html.

Rodrigo Casado Noguerales, Bernhard Schölkopf, Thomas Hofmann, and Aran Raoufi. What does a discrete diffusion model learn? *arXiv preprint arXiv:2607.05381*, 2026. URL https://arxiv.org/abs/2607.05381.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11315–11325, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Chang_MaskGIT_Masked_Generative_Image_Transformer_CVPR_2022_paper.html.

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

Giovanni Conforti and Christian Léonard. Time reversal of Markov processes with jumps under a finite entropy condition. *Stochastic Processes and their Applications*, 144:85–124, 2022. doi: 10.1016/j.spa.2021.10.002. URL https://doi.org/10.1016/j.spa.2021.10.002.

Fabrizio Costa and Kurt De Grave. Fast neighborhood subgraph pairwise distance kernel. In *Proceedings of the 27th International Conference on Machine Learning*, 2010. URL https://icml.cc/Conferences/2010/papers/347.pdf.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H. Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, Curtis Hawthorne, Rémi Leblond, Will Grathwohl, and Jonas Adler. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022. URL https://arxiv.org/abs/2211.15089.

Daniil Dmitriev, Zhihan Huang, and Yuting Wei. Provably adaptive sampling with uniform and remasking discrete diffusion models. *arXiv preprint arXiv:2608.23554*, 2026. URL https://arxiv.org/abs/2608.23554.

Martin R. Evans, Satya N. Majumdar, and Grégory Schehr. Stochastic resetting and applications. *Journal of Physics A: Mathematical and Theoretical*, 53(19):193001, 2020. doi: 10.1088/1751-8121/ab7cfe. URL https://arxiv.org/abs/1910.07993.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4239. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/f0d629a734b56a642701bba7bc8bb3ed-Abstract-Conference.html.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-Predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6112–6121. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1633. URL https://aclanthology.org/D19-1633/.

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. *arXiv preprint arXiv:2605.22765*, 2026. URL https://arxiv.org/abs/2605.22765.

Ishaan Gulrajani and Tatsunori B. Hashimoto. Likelihood-based diffusion language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper_files/paper/2023/file/35b5c175e139bff5f22a5361270fce87-Paper-Conference.pdf.

Peter Holderrieth, Marton Havasi, Jason Yim, Neta Shaul, Itai Gat, Tommi Jaakkola, Brian Karrer, Ricky T. Q. Chen, and Yaron Lipman. Generator matching: Generative modeling with arbitrary Markov processes. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.20587.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1904.09751.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10362–10383, 2022. URL https://proceedings.mlr.press/v162/jo22a.html.

Pascal Jutras-Dubé, Patrick Pynadath, Jeremy Lu, Yuan Gao, and Ruqi Zhang. Sticky jump diffusions: A unifying view of masked, continuous, and hybrid diffusion. *arXiv preprint arXiv:2607.10951*, 2026. URL https://arxiv.org/abs/2607.10951.

Hiroshi Kunita. *Stochastic Flows and Jump-Diffusions*, volume 92 of *Probability Theory and Stochastic Modelling*. Springer, Singapore, 2019. ISBN 978-981-13-3800-7. doi: 10.1007/978-981-13-3801-4. URL https://doi.org/10.1007/978-981-13-3801-4.

Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.neurips.cc/paper/8920-high-quality-self-supervised-deep-image-denoising.pdf.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 4328–4343, 2022. URL https://arxiv.org/abs/2205.14217.

Xingchao Liu, Lemeng Wu, Mao Ye, and Qiang Liu. Let us build bridges: Understanding and extending diffusion generative models. *arXiv preprint arXiv:2208.14699*, 2022. URL https://arxiv.org/abs/2208.14699.

Matt Mahoney. About the test data, 2006. URL https://mattmahoney.net/dc/textdata.html. Text8 dataset; section "Relationship of Wikipedia Text to Clean Text".

Mikhail Papkov, Pavel Chizhov, and Leopold Parts. SwinIA: Self-supervised blind-spot image denoising without convolutions. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 7071–7080, 2025. URL https://openaccess.thecvf.com/content/WACV2025/html/Papkov_SwinIA_Self-Supervised_Blind-Spot_Image_Denoising_without_Convolutions_WACV_2025_paper.html.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://arxiv.org/abs/2212.09748.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet ChemNet distance: A metric for generative models for molecules in drug discovery. *Journal of Chemical Information and Modeling*, 58(9):1736–1741, 2018. doi: 10.1021/acs.jcim.8b00234. URL https://doi.org/10.1021/acs.jcim.8b00234.

Patrick Pynadath, Jiaxin Shi, and Ruqi Zhang. CANDI: Hybrid discrete-continuous diffusion models. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.22510.

Subham Sekhar 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 *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4135. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. URL https://arxiv.org/abs/2002.05202.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/bad233b9849f019aead5e5cc60cef70f-Abstract-Conference.html.

Joongbo Shin, Yoonhyung Lee, Seunghyun Yoon, and Kyomin Jung. Fast and accurate deep bidirectional language representations for unsupervised learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 823–835. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.76. URL https://aclanthology.org/2020.acl-main.76/.

Teague Sterling and John J. Irwin. ZINC 15—ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 55(11):2324–2337, 2015. doi: 10.1021/acs.jcim.5b00559. URL https://doi.org/10.1021/acs.jcim.5b00559.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.16750.

Zhihan Yang, Wei Guo, Shuibai Zhang, Subham Sekhar Sahoo, Yongxin Chen, Arash Vahdat, Morteza Mardani, and John Thickstun. Continuous diffusion scales competitively with discrete diffusion for language. *arXiv preprint arXiv:2605.18530*, 2026. URL https://arxiv.org/abs/2605.18530.

Yixiu Zhao, Jiaxin Shi, Feng Chen, Shaul Druckmann, Lester Mackey, and Scott Linderman. Informed correctors for discrete diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4183. URL https://papers.nips.cc/paper_files/paper/2025/hash/b67d9d07faf071532c420ed5b140756b-Abstract-Conference.html.

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 *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.01329.

Cai Zhou, Chenxiao Yang, Yi Hu, Chenyu Wang, Chubin Zhang, Muhan Zhang, Lester Mackey, Tommi Jaakkola, Stephen Bates, and Dinghuai Zhang. Coevolutionary continuous discrete diffusion: Make your diffusion language model a latent reasoner. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.03206.

A Conditional law and analytic likelihood

A.1 Jump rate kernels and Markov generators

Jump rate kernels and destination probabilities. Let S denote the state space. A *jump rate kernel* $K_t(x, dz)$ is a nonnegative measure in the destination dz and measurable in x. It specifies the instantaneous rates of jump events into subsets of S, rather than probabilities over a fixed time interval (Conforti & Léonard, 2022). For the processes considered here, its total mass $\lambda_t(x) = K_t(x, S)$ is finite at each $t < 1$ and gives the total jump intensity. When $\lambda_t(x) > 0$, $J_t(x, dz) = K_t(x, dz)/\lambda_t(x)$ is the destination probability kernel conditional on an event. When $\lambda_t(x) = 0$, the instantaneous event rate is zero and this destination law may be chosen arbitrarily.

For a finite-state pure-jump process with rates continuous in time, let N_t count all jump events, including self-resets. As $h \downarrow 0$,

$$\mathbb{P}(N_{t+h} - N_t = 0 \mid X_t = x) = 1 - \lambda_t(x)h + o(h),$$
$$\mathbb{P}(N_{t+h} - N_t \geq 1 \mid X_t = x) = \lambda_t(x)h + o(h).$$

Thus $\lambda_t(x)$ is a rate, and $\lambda_t(x)h$ approximates the probability of an event over a short interval of length h. Self-resets may be included in K_t; the rate of actual state changes is then $K_t(x, S \setminus \{x\}) = \lambda_t(x) - K_t(x, \{x\})$. Accordingly,

$$\mathbb{P}(X_{t+h} = x \mid X_t = x) = 1 - h\big[\lambda_t(x) - K_t(x, \{x\})\big] + o(h),$$
$$\mathbb{P}(X_{t+h} = z \mid X_t = x) = hK_t(x, \{z\}) + o(h), \qquad z \neq x.$$

Without self-resets, the first probability is $1 - \lambda_t(x)h + o(h)$; the second is $h\lambda_t(x)J_t(x, \{z\}) + o(h)$ when $\lambda_t(x) > 0$. For a jump diffusion, no reset does not imply an unchanged state, since the drift and Brownian motion continue between resets. We use *jump rate kernel* for K_t and *destination probabilities* for normalized laws such as J_t.

Discrete transitions and embedding resets. On a discrete state space, the masses $K_t(x, \{z\})$, $z \neq x$, are also called transition rates or transition intensities. In the masked model, the per-position rates in (2) specify

$$K_t(x, dz) = \sum_{i=1}^{L} \sum_{a \in \mathcal{V}} r_{i,t}(x_i \to a \mid x)\delta_{x^{i \leftarrow a}}(dz),$$

where δ_z is the point mass at z, and coordinate replacement is understood here for token sequences. For the embedding-space resets, the kernels in (4) and (8) are, respectively,

$$K_t(x, dz) = \lambda(t) \sum_{i,a} p_{i,t}(a \mid x)\delta_{x^{i \leftarrow e(a)}}(dz),$$

$$K_t^{\mathrm{ctx}}(x, dz) = \lambda(t) \sum_{i,a} q_{i,t}^{\star}(a \mid x_{-i})\delta_{x^{i \leftarrow e(a)}}(dz).$$

Both have total reset intensity $\lambda_t(x) = L\lambda(t)$; $\lambda(t)$ is the per-position clock intensity. The categorical laws $p_{i,t}$ and $q_{i,t}^{\star}$ specify destination tokens conditional on resetting position i; multiplying by the clock intensity gives the corresponding contribution to the jump rate kernel.

Markov generators. The *infinitesimal generator*, or *Markov generator*, describes the instantaneous change in expected test functions (Holderrieth et al., 2025, Section 4). For our jump diffusions,

$$\mathcal{A}_t f(x) = \sum_i \left[b_{i,t}(x) \cdot \nabla_i f(x) + \frac{\sigma_d^2}{2} \Delta_i f(x) \right] + \int_S [f(z) - f(x)] K_t(x, dz).$$

For a pure-jump process, the continuous terms are absent. On a finite state space, the generator matrix has off-diagonal entries $K_t(x, \{z\})$ and diagonal entry $-K_t(x, S \setminus \{x\})$; its rows sum to zero. Self-resets contribute zero to the generator.

A.2 Conditional jump-diffusion SDE

For background on stochastic differential equations with jumps and their generators, see Kunita (2019, Chapters 3–4). We derive the conditional state law used in Remark 2.2 by conditioning on the last reset time, as in last-renewal representations of stochastic resetting (Evans et al., 2020).

As in the main construction, let $\lambda(t) \geq 0$ satisfy $\int_0^t \lambda(s)\, ds < \infty$ for every $t < 1$. Fix $y \in \mathcal{V}^L$. Let W_i be independent d-dimensional Brownian motions and N_i independent inhomogeneous Poisson counting processes with intensity measure $\lambda(t)\, dt$. Recall the Jump SDE from (3)

$$dX_{i,t}^y = \frac{e(y_i) - X_{i,t-}^y}{1-t}\, dt + \sigma_d\, dW_{i,t} + (e(y_i) - X_{i,t-}^y)\, dN_{i,t}, \qquad X_{i,0}^y \sim \mathcal{N}(0, \sigma_s^2 I_d). \tag{11}$$

We take $\sigma_d > 0$ and $\sigma_s \geq 0$, with a base law $\mathcal{N}(0, \sigma_s^2 I_d)$ independent of y. Here $X_{i,t-}^y$ is the left limit. At a jump of N_i, the increment is $e(y_i) - X_{i,t-}^y$, so $X_{i,t}^y = e(y_i)$ exactly. The Brownian motion and counting process continue after this event. On each $[0, T]$, $T < 1$, there are almost surely finitely many resets, and the between-reset linear SDE has a unique solution. Write $p_t(\cdot \mid y) = \mathcal{L}(X_t^y)$.

Lemma A.1 (Conditional state law)**.** *Consider* (11) *with deterministic λ and mutually independent initial states, Brownian motions, and Poisson processes. For every $0 < t < 1$ and $y_i = a$, the law $p_{i,t}(\cdot \mid a) = \mathcal{L}(X_{i,t}^y)$ is the Gaussian mixture*

$$p_{i,t}(dx \mid a) = S(0,t)\mathcal{N}(dx; te(a), v_{0,t} I_d) + \int_0^t \lambda(r) S(r,t) \mathcal{N}(dx; e(a), v_{r,t} I_d)\, dr, \tag{12}$$

where $S(r,t) = \exp(-\int_r^t \lambda(u)\, du)$, $v_{0,t} = (1-t)^2\sigma_s^2 + \sigma_d^2 t(1-t)$, and $v_{r,t} = \sigma_d^2(t-r)(1-t)/(1-r)$ for $0 < r < t$. The coordinates are conditionally independent, so $p_t(dx \mid y) = \prod_i p_{i,t}(dx_i \mid y_i)$.

Proof. For $0 \leq s < t < 1$, if no reset occurs during $(s, t]$, solving the linear SDE gives

$$X_{i,t}^y = e(y_i) + \frac{1-t}{1-s}(X_{i,s}^y - e(y_i)) + \sigma_d(1-t) \int_s^t \frac{dW_{i,u}}{1-u}.$$

The last term is centered Gaussian with covariance $\sigma_d^2(t-s)(1-t)/(1-s) I_d$. If there is no reset before t, set $s = 0$ and use the independent Gaussian initial state. This gives mean $te(a)$ and covariance $v_{0,t} I_d$, where $v_{0,t} = (1-t)^2\sigma_s^2 + \sigma_d^2 t(1-t)$. If the last reset is at $r \in (0,t)$, then $X_{i,r}^y = e(a)$. Setting $s = r$ therefore gives mean $e(a)$ and covariance $v_{r,t} I_d$, where $v_{r,t} = \sigma_d^2(t-r)(1-t)/(1-r)$.

It remains to weight these Gaussian laws by the reset probabilities. Let $\tau_{i,t}$ be the last reset time of coordinate i up to time t, with $\tau_{i,t} = 0$ if no reset has occurred. For $0 \leq r < t$,

$$\mathbb{P}(\tau_{i,t} \leq r) = \mathbb{P}(N_{i,t} - N_{i,r} = 0) = \exp\left(- \int_r^t \lambda(u)\, du \right) = S(r,t).$$

Thus $\tau_{i,t}$ has an atom of mass $S(0,t)$ at zero and density $\partial_r S(r,t) = \lambda(r) S(r,t)$ for almost every $r \in (0,t)$. Independence of the Poisson process from the Brownian motion and initial state ensures that conditioning on these reset times leaves the Gaussian laws above unchanged. Weighting the no-reset law by $S(0,t)$ and integrating the last-reset law against $\lambda(r)S(r,t)\, dr$ proves (12). The weights sum to one because $S(0,t) + \int_0^t \lambda(r) S(r,t)\, dr = 1$. Finally, each coordinate depends only on its own initial state and drivers, which proves conditional coordinate independence. □

Since $(t-r)/(1-r) \le t$ for $0 \le r < t < 1$, the Gaussian mixture above gives

$$\mathbb{E}\big[\|X^y_{i,t} - e(y_i)\|^2\big] \le (1-t)^2\big(\|e(y_i)\|^2 + d\sigma_s^2\big) + d\sigma_d^2 t(1-t) \longrightarrow 0.$$

Summing over positions and averaging over Y yields $p_t \Rightarrow \mathcal{L}(E(Y))$ as $t \uparrow 1$. For a bridge clock κ, replace t by $\kappa(t)$ in this bound.

A.3 Closed form for a shared bridge and reset schedule

The closed form extends beyond the linear clock: it applies whenever the bridge clock and the first-reset CDF are both $\kappa(t)$. For the baseline reset schedule, we use it for the repeated-reset training law and for likelihood completion in RJD-BC-D, with $\kappa(t) = t^2$.

Proposition A.2 (Conditional law and likelihood)**.** *Let κ satisfy the clock conditions in Section 2.1. Consider the repeated-reset bridge under the independence and noise assumptions of Lemma A.1, with drift factor $\kappa'(t)/(1-\kappa(t))$, diffusion coefficient $\sigma_d\sqrt{\kappa'(t)}$, and reset rate $\lambda(t) = \kappa'(t)/(1-\kappa(t))$. For $y_i = a$ and $u = \kappa(t) \in (0,1)$, put $v_u = \sigma_d^2 u(1-u)$ and $v_{0,u} = (1-u)^2\sigma_s^2 + v_u$. Then*

$$p_{i,t}(dx \mid a) = (1-u)\mathcal{N}(dx; ue(a), v_{0,u} I_d) + u\int_0^1 \mathcal{N}(dx; e(a), v_u z I_d)\, dz. \tag{13}$$

Its moments satisfy

$$\begin{aligned} \mathbb{E}[X_{i,t} \mid Y_i = a] &= u(2-u)e(a), \\ \mathbb{E}[\|X_{i,t} - e(a)\|^2 \mid Y_i = a] &= (1-u)^3(\|e(a)\|^2 + d\sigma_s^2) + d\sigma_d^2 u(1-u)(1-u/2). \end{aligned} \tag{14}$$

For even $d \ge 4$ and $x \ne e(a)$, let $n = d/2 - 1$ and $\beta = \|x - e(a)\|^2/(2v_u) > 0$. Conditional on at least one reset, the density is

$$\begin{aligned} \int_0^1 \mathcal{N}(x; e(a), v_u z I_d)\, dz &= (2\pi v_u)^{-d/2}\beta^{-n}\Gamma(n,\beta), \\ \Gamma(n,\beta) &= (n-1)!e^{-\beta}\sum_{k=0}^{n-1}\frac{\beta^k}{k!}. \end{aligned} \tag{15}$$

Here $\Gamma(n,\beta) = \int_\beta^\infty w^{n-1}e^{-w}\,dw$ is the upper incomplete gamma function.

Proof. The process is the deterministic time change $u = \kappa(t)$ of the linear-clock bridge with reset rate $(1-u)^{-1}$. Lemma A.1 therefore gives no-reset weight $1-u$ and last-reset density $(1-u)/(1-\tau)^2$, for $0 < \tau < u$. Following a last reset at τ, the mean is $e(a)$ and the variance is $\sigma_d^2(u-\tau)(1-u)/(1-\tau)$. The substitution $z = (u-\tau)/(u(1-\tau))$ transforms this variance into $v_u z$ and the last-reset measure into $u\,dz$ on $[0,1]$, with reversed limits. Together with the no-reset Gaussian, this proves (13).

The component means give $(1-u)ue(a) + ue(a) = u(2-u)e(a)$. The expected squared distance from $e(a)$ is $(1-u)[(1-u)^2\|e(a)\|^2 + dv_{0,u}] + udv_u/2$, since $\int_0^1 z\,dz = 1/2$. Expanding gives (14). For the density, substituting $w = \beta/z$ yields

$$\int_0^1 z^{-d/2}e^{-\beta/z}\,dz = \beta^{-n}\int_\beta^\infty w^{n-1}e^{-w}\,dw.$$

Repeated integration by parts for the positive integer n gives the finite sum in (15). □

To sample (13), use the first Gaussian with probability $1-u$; otherwise draw $z \sim \mathrm{Unif}(0,1)$ and $\xi \sim \mathcal{N}(0, I_d)$ independently and set $x = e(a) + \sqrt{v_u z}\,\xi$. In dimension 64, the density formula uses $n = 31$ and is evaluated in log space without reset-time quadrature. For even $d \ge 4$ and $0 < u < 1$, the density is singular at an anchor, but the law has no atom there; a posterior version may be chosen on these zero-measure points. At $u = 0$, including $\sigma_s = 0$, the common base law is handled directly. The mean squared distance in (14) tends to zero at the data endpoint.

Other jump schedules. The proposition covers $\lambda_\gamma(t) = \gamma t^{\gamma-1}/(1-t^\gamma)$ when the bridge also uses $\kappa(t) = t^\gamma$. If the bridge remains at $\kappa(t) = t^2$ while γ changes, the repeated-reset process has no-reset weight $1-t^\gamma$ and last-reset density $\gamma r^{\gamma-1}(1-t^\gamma)/(1-r^\gamma)^2$ on $(0,t)$. For repeated resets, these weights must be combined with the Gaussian laws in bridge time: the no-reset mean and variance are $\kappa(t)e(a)$ and $v_{0,\kappa(t)}$, and a last reset at physical time r gives mean $e(a)$ and variance $\sigma_d^2(\kappa(t)-\kappa(r))(1-\kappa(t))/(1-\kappa(r))$. This is the last-reset construction of Lemma A.1; the uniform-mixture formula above does not generally apply. Our non-absorbing schedule experiments also use $\gamma = 1, 4$ with the bridge fixed at $\kappa(t) = t^2$. Training draws from this schedule-specific last-reset mixture. For completed drift, the local likelihood is evaluated by numerical quadrature; $\gamma = 2$ retains the closed form above. The quadrature uses 256 generalized Gauss–Laguerre nodes and checks the completed posterior against 128 nodes at every sampled position, aborting if the maximum ℓ_1 difference exceeds 0.002. This is a numerical convergence check, not an exact likelihood evaluation. The raw LOO and unmasked sampling arms do not evaluate the local likelihood.

For analytic completion with a learned predictor, we add $\log \ell_{i,t}(x_i \mid a)$ to contextual logits and normalize over a. The local likelihood is computed from the conditional training path using the learned token embeddings, which also define reset destinations and nearest-embedding decoding.

B Proofs and a counterexample

B.1 Minimizers of the training losses

All expectations below use the training pair (Y, X_t^Y). Logarithms are natural.

Full-posterior prediction. Recall the loss (5),

$$\mathcal{L}_{\text{full}}(\theta) = \mathbb{E}_{t,Y,X_t^Y}\left[-\sum_{i=1}^{L} \log p_{\theta,i,t}(Y_i \mid X_t^Y)\right].$$

Condition on $(t, X_t^Y) = (t, x)$. At position i, the conditional loss from predicting $p_{\theta,i,t}(\cdot \mid x)$ instead of $p_{i,t}(\cdot \mid x)$ is

$$\mathrm{KL}(p_{i,t}(\cdot \mid x) \| p_{\theta,i,t}(\cdot \mid x)) \geq 0.$$

Averaging over training times and states and summing over positions therefore shows that the unrestricted minimizers equal $p_{i,t}$ almost everywhere. The learned drift is

$$b_{\theta,i,t}(x) = \frac{\sum_{a\in\mathcal{V}} p_{\theta,i,t}(a \mid x)e(a) - x_i}{1-t},$$

and the reset probabilities are $p_{\theta,i,t}$. When $p_{\theta,i,t} = p_{i,t}$, these coefficients and the corresponding reset clock recover the posterior-averaged generator for the chosen path.

Leave-one-out prediction. Write $X_{-i,t}^Y$ for X_t^Y with position i omitted. The loss (6) is

$$\mathcal{L}_{\text{LOO}}(\theta) = \mathbb{E}_{t,Y,X_t^Y}\left[-\sum_{i=1}^{L} \log q_{\theta,i,t}(Y_i \mid X_{-i,t}^Y)\right].$$

Conditioning on $(t, X_{-i,t}^Y)$ gives the conditional loss

$$\mathrm{KL}\big(q_{i,t}^\star(\cdot \mid X_{-i,t}^Y) \| q_{\theta,i,t}(\cdot \mid X_{-i,t}^Y)\big).$$

The same argument shows that the unrestricted minimizers equal $q_{i,t}^\star$ almost everywhere under the training law, establishing (7).

B.2 Assumptions and proof of Proposition 3.2

Proposition B.1. *Consider the conditional path* (3) *under the assumptions of Appendix A.2, with any nonnegative deterministic reset rate $\lambda(t)$ satisfying $\int_0^t \lambda(s)\,ds < \infty$ for every $t < 1$. Let $p_t(dx) = \sum_y p_{\text{data}}(y) p_t(dx \mid y)$, and let $p_{i,t}$ and $q_{i,t}^\star$ be versions of $\mathbb{P}(Y_i \in \cdot \mid X_t^Y)$ and $\mathbb{P}(Y_i \in \cdot \mid X_{-i,t}^Y)$*

under this mixture. On every $[0, T]$*,* $T < 1$*, assume that* (8) *admits a Markov process started from* p_0 *whose laws satisfy its integrated weak forward equation against smooth, compactly supported test functions, and that this equation has a unique probability-law solution from* p_0*. Then this Markov process has law* p_t *for every* $t < 1$*, and* $p_t \Rightarrow \mathcal{L}(E(Y))$ *as* $t \uparrow 1$*.*

Proof. Fix $T < 1$. We establish two claims for every smooth, compactly supported test function $f : \mathbb{R}^{dL} \to \mathbb{R}$.

Claim 1. The mixture p_t satisfies the integrated weak forward equation for the averaged operator $\mathcal{A}_t$:

$$\int f\, dp_t - \int f\, dp_0 = \int_0^t \int \mathcal{A}_s f(x)\, p_s(dx)\, ds, \qquad 0 \le t \le T.$$

Proof of Claim 1. For a fixed endpoint $y \in \mathcal{V}^L$, the conditional generator of (3) is

$$\mathcal{A}_s^y f(x) = \sum_{i=1}^{L} \left[\frac{e(y_i) - x_i}{1 - s} \cdot \nabla_i f(x) + \frac{\sigma_d^2}{2} \Delta_i f(x) \right] + \lambda(s) \sum_{i=1}^{L} \left[f(x^{i \leftarrow e(y_i)}) - f(x) \right].$$

Here ∇_i and Δ_i are the gradient and Laplacian with respect to the block $x_i \in \mathbb{R}^d$, and $x^{i \leftarrow e(y_i)}$ replaces that block by $e(y_i)$. The continuous terms are uniformly bounded over $s \le T$, x, and y: the derivatives of f have compact support, $1 - s \ge 1 - T$, and the finite vocabulary makes the embeddings bounded. The jump term is bounded in absolute value by $2L\lambda(s)\|f\|_\infty$, where $\|f\|_\infty = \sup_x |f(x)|$. These bounds and $\int_0^T \lambda(s)\, ds < \infty$ justify taking expectations in the jump Itô formula and interchanging the time integral with expectation; the Brownian and compensated jump integrals have zero expectation. There are no jumps at any fixed deterministic time almost surely, since the Poisson intensity measures are $\lambda(s)\, ds$, so the left limits in that formula can be replaced by X_s^y under expectation.

Now draw $Y \sim p_{\mathrm{data}}$ and write $\mathcal{A}_s^Y$ for the conditional generator with endpoint Y. Since $\mathbb{P}(Y_i = a \mid X_s^Y = x) = p_{i,s}(a \mid x)$, conditioning $\mathcal{A}_s^Y f(X_s^Y)$ on $X_s^Y = x$ gives exactly $\mathcal{A}_s f(x)$ in (4). Consequently, for $0 \le t \le T$, taking expectation over Y and its conditional process gives

$$\begin{aligned} \int f\, dp_t - \int f\, dp_0 &= \int_0^t \mathbb{E}[\mathcal{A}_s^Y f(X_s^Y)]\, ds \\ &= \int_0^t \int \mathcal{A}_s f(x)\, p_s(dx)\, ds. \end{aligned}$$

This proves Claim 1 and establishes posterior averaging directly, following the conditional-expectation argument of Holderrieth et al. (2025, Proposition 1).

Claim 2. Replacing the full-posterior reset probabilities by the LOO probabilities leaves the generator integral under p_t unchanged:

$$\int \mathcal{A}_t f\, dp_t = \int \mathcal{A}_t^{\mathrm{ctx}} f\, dp_t, \qquad 0 \le t \le T.$$

Proof of Claim 2. From (4) and (8), using $\sum_a p_{i,t}(a \mid x) = \sum_a q_{i,t}^\star(a \mid x_{-i}) = 1$,

$$(\mathcal{A}_t - \mathcal{A}_t^{\mathrm{ctx}}) f(x) = \lambda(t) \sum_{i,a} \left[p_{i,t}(a \mid x) - q_{i,t}^\star(a \mid x_{-i}) \right] f(x^{i \leftarrow e(a)}).$$

The tower property gives

$$\int p_{i,t}(a \mid x)\, p_t(dx_i \mid x_{-i}) = q_{i,t}^\star(a \mid x_{-i}).$$

Since $f(x^{i\leftarrow e(a)})$ does not depend on x_i,

$$\int (\mathcal{A}_t - \mathcal{A}_t^{\mathrm{ctx}}) f \, dp_t$$
$$= \lambda(t) \sum_{i,a} \int f(x^{i\leftarrow e(a)}) \left[\int p_{i,t}(a \mid x) \, p_t(dx_i \mid x_{-i}) - q^{\star}_{i,t}(a \mid x_{-i}) \right] p_t(dx_{-i}) = 0,$$

which proves Claim 2.

Combining Claims 1 and 2, p_t solves the integrated weak forward equation for $\mathcal{A}_t^{\mathrm{ctx}}$. By the assumed uniqueness, the Markov process started from p_0 therefore has law p_t on $[0, T]$. Since $T < 1$ was arbitrary, this holds for every $t < 1$. Endpoint convergence follows from the bound in Appendix A.2. □

B.3 A COUNTEREXAMPLE FOR LOO DRIFT

Example B.2 (LOO drift with informative context)**.** Take $L = 2$, $d = 1$, and $Y = (B, B)$, where B is uniform on $\{-1, 1\}$; let $e(a) = a$. Conditional on B, let the coordinates follow independent copies of the repeated-reset bridge (3), with $X_{1,0} = X_{2,0} = 0$, $\sigma_d = 1$, and $\lambda(t) = (1-t)^{-1}$. Both positions therefore carry noisy information about the same endpoint. Define, under this prescribed conditional-path mixture,

$$m_t(x) = \mathbb{E}[B \mid X_{2,t} = x], \qquad M_t(x_1, x_2) = \mathbb{E}[B \mid X_{1,t} = x_1, X_{2,t} = x_2].$$

By symmetry, m_t is also the posterior mean given the first coordinate alone. The drift $b_{i,t}$ at position i is $(M_t(x_1, x_2) - x_i)/(1-t)$; its LOO replacement is $(m_t(x_{3-i}) - x_i)/(1-t)$. Let $\widetilde{\mathcal{A}}_t$ change only these drifts in (4), retaining the exact jump probabilities $p_{i,t}$, reset clocks, and diffusion coefficients. The original process with generator $\mathcal{A}_t$ has marginal law p_t. Write $\widetilde{p}_t$ for the marginal law of the process with generator $\widetilde{\mathcal{A}}_t$, started from the same initial law $\widetilde{p}_0 = p_0 = \delta_{(0,0)}$.

Claim. Replacing the drift in this way does not preserve the prescribed marginal path $(p_t)_{0 \le t < 1}$, even though the jump rate kernel remains exact.

Proof. Take $f(x) = x_1^2$. All second moments are finite on compact time intervals. Each generator determines the second-moment derivative under its own marginal law:

$$\frac{d}{dt} \mathbb{E}_{p_t}[X_1^2] = \int \mathcal{A}_t f \, dp_t,$$
$$\frac{d}{dt} \mathbb{E}_{\widetilde{p}_t}[X_1^2] = \int \widetilde{\mathcal{A}}_t f \, d\widetilde{p}_t.$$

Suppose, for contradiction, that $\widetilde{p}_t = p_t$ for every $t < 1$. Then these second moments, and hence their derivatives, must agree.

To compare these derivatives, we first compute the generator difference. The continuous diffusion and jump terms of the two generators coincide. For $f(x) = x_1^2$, we have $\partial_1 f = 2x_1$ and $\partial_2 f = 0$, so only the difference between the first-coordinate drifts contributes:

$$(\widetilde{\mathcal{A}}_t - \mathcal{A}_t) f(x) = \frac{2x_1}{1-t} \big(m_t(x_2) - M_t(x_1, x_2) \big).$$

Integrating against p_t gives

$$\int (\widetilde{\mathcal{A}}_t - \mathcal{A}_t) f \, dp_t = \frac{2}{1-t} \left(\mathbb{E}[X_{1,t} m_t(X_{2,t})] - \mathbb{E}[X_{1,t} M_t(X_{1,t}, X_{2,t})] \right).$$

It remains to evaluate the two expectations on the right. Both are under the prescribed joint law of (B, X_t), whose state marginal is p_t. With $\kappa(t) = t$, (14) gives $\mathbb{E}[X_{1,t} \mid B] = t(2-t)B$.

For the first expectation, involving the LOO term, condition on both B and $X_{2,t}$. The coordinates are independent conditional on B, so observing $X_{2,t}$ provides no further information about $X_{1,t}$ once B is known. Thus

$$\mathbb{E}[X_{1,t} \mid B, X_{2,t}] = \mathbb{E}[X_{1,t} \mid B] = t(2-t)B.$$

Since $m_t(X_{2,t})$ is determined by $X_{2,t}$, it can be taken outside this conditional expectation. The tower property therefore gives

$$\begin{aligned}\mathbb{E}[X_{1,t}m_t(X_{2,t})] &= \mathbb{E}[\mathbb{E}[X_{1,t}m_t(X_{2,t}) \mid B, X_{2,t}]] \\ &= \mathbb{E}[m_t(X_{2,t})\mathbb{E}[X_{1,t} \mid B, X_{2,t}]] \\ &= t(2-t)\mathbb{E}[Bm_t(X_{2,t})].\end{aligned}$$

Next condition on $X_{2,t}$ alone and use the definition $m_t(X_{2,t}) = \mathbb{E}[B \mid X_{2,t}]$:

$$\begin{aligned}\mathbb{E}[Bm_t(X_{2,t})] &= \mathbb{E}[\mathbb{E}[Bm_t(X_{2,t}) \mid X_{2,t}]] \\ &= \mathbb{E}[m_t(X_{2,t})\mathbb{E}[B \mid X_{2,t}]] \\ &= \mathbb{E}[m_t(X_{2,t})^2].\end{aligned}$$

Combining these two calculations yields

$$\mathbb{E}[X_{1,t}m_t(X_{2,t})] = t(2-t)\mathbb{E}[m_t(X_{2,t})^2].$$

For the second expectation, involving the full-posterior term, recall that $M_t(X_{1,t}, X_{2,t}) = \mathbb{E}[B \mid X_{1,t}, X_{2,t}]$. Here $X_{1,t}$ is known when conditioning on both coordinates, so

$$\begin{aligned}\mathbb{E}[X_{1,t}M_t(X_{1,t}, X_{2,t})] &= \mathbb{E}[X_{1,t}\mathbb{E}[B \mid X_{1,t}, X_{2,t}]] \\ &= \mathbb{E}[\mathbb{E}[X_{1,t}B \mid X_{1,t}, X_{2,t}]] \\ &= \mathbb{E}[X_{1,t}B].\end{aligned}$$

Finally, conditioning on B and using $B^2 = 1$ gives

$$\begin{aligned}\mathbb{E}[X_{1,t}B] &= \mathbb{E}[B\mathbb{E}[X_{1,t} \mid B]] \\ &= t(2-t)\mathbb{E}[B^2] = t(2-t).\end{aligned}$$

Substituting these two expectations into the generator difference gives

$$\begin{aligned}\int (\widetilde{\mathcal{A}}_t - \mathcal{A}_t) f \, dp_t &= \frac{2t(2-t)}{1-t}\left(\mathbb{E}[m_t(X_{2,t})^2] - 1\right) \\ &= -\frac{2t(2-t)}{1-t}\mathbb{E}\big[1 - m_t(X_{2,t})^2\big] < 0, \qquad 0 < t < 1. \end{aligned} \tag{16}$$

To justify the strict inequality in (16), we must show that $\mathbb{E}[1 - m_t(X_{2,t})^2] > 0$. For either endpoint $B = \pm 1$, the local law of $X_{2,t}$ contains the no-reset component $\mathcal{N}(\pm t, t(1-t))$ with weight $1-t > 0$. For $0 < t < 1$, both Gaussian densities are positive everywhere. Hence both values of B retain positive conditional probability after observing $X_{2,t}$, so $|m_t(X_{2,t})| < 1$ almost surely. Since $B^2 = 1$, this gives

$$1 - m_t(X_{2,t})^2 = \mathrm{Var}(B \mid X_{2,t}) > 0 \quad \text{almost surely.}$$

Its expectation is therefore strictly positive, while the prefactor $-2t(2-t)/(1-t)$ is strictly negative. This proves the strict negativity of the residual. Figure 3 plots this residual as a function of time.

Under the assumed equality $\widetilde{p}_t = p_t$, this yields

$$\frac{d}{dt}\left(\mathbb{E}_{\widetilde{p}_t}[X_1^2] - \mathbb{E}_{p_t}[X_1^2]\right) = \int (\widetilde{\mathcal{A}}_t - \mathcal{A}_t) f \, dp_t < 0, \qquad 0 < t < 1,$$

contradicting equality of the derivatives. Thus the two marginal paths cannot coincide. □

B.4 Proof of Proposition 3.3

Proof. Fix i and t. Given Y, the coordinates are independent, and the law of $X_{i,t}$ depends on Y only through Y_i. Averaging over the other endpoint tokens therefore gives $X_{i,t} \perp X_{-i,t} \mid Y_i$. Consequently, for almost every context x_{-i},

$$\begin{aligned}\mathbb{P}(Y_i = a, X_{i,t} \in dx_i \mid X_{-i,t} = x_{-i}) &= q^\star_{i,t}(a \mid x_{-i}) p_{i,t}(dx_i \mid a) \\ &= q^\star_{i,t}(a \mid x_{-i}) \ell_{i,t}(x_i \mid a) \mu_{i,t}(dx_i).\end{aligned}$$

Summing over a shows that the conditional density of $X_{i,t}$ given $X_{-i,t} = x_{-i}$, with respect to $\mu_{i,t}$, is

$$\sum_b q^\star_{i,t}(b \mid x_{-i}) \ell_{i,t}(x_i \mid b).$$

Bayes' rule divides the joint density by this sum, yielding (10) for p_t-almost every x where the denominator is positive. □

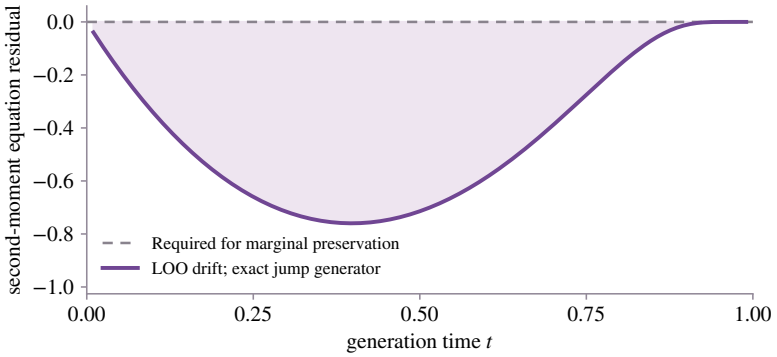

Figure 3: **LOO drift violates the moment equation with informative context.** The residual in (16) for Example B.2, with $Y = (B, B)$ and $\sigma_d = 1$. Preserving the prescribed marginal path requires a zero residual; replacing only the drift posterior by LOO gives a negative residual. The expectation is evaluated by deterministic quadrature of the exact local conditional law, not by trajectory simulation.

C Training and sampling

Algorithms 1 and 2 summarize training and generation for the repeated-reset model with bridge clock κ and deterministic physical-time reset rate λ. The numerical sampler freezes predictions on each grid interval and uses the no-reset probability and last-reset distribution described below. Sampling stops before $t = 1$, where the chosen reset schedules have infinite cumulative intensity.

C.1 Frozen-coefficient interval update

For the linear bridge clock $\kappa(t) = t$, consider an interval $[a, b] \subset [0, 1)$ and freeze the contextual reset distribution q_i and drift endpoint mean m_i at its left endpoint. With $S(a, b) = \exp(-\int_a^b \lambda(r)\, dr)$, the probability of no reset is $S(a, b)$, and the unconditional density of the last reset time $\tau \in (a, b)$ is $\lambda(\tau)S(\tau, b)$. Its destination is sampled from q_i, independently of the clock. Earlier resets can be integrated out because the last reset replaces the state completely.

Set $\tau = a$ and $z_i = X_{i,a}$ if no reset occurs; otherwise let τ be the last reset time and z_i its destination embedding. The exact update for this frozen-coefficient process is

$$X_{i,b} = m_i + \alpha(z_i - m_i) + \sigma_d\sqrt{(b-\tau)\alpha}\,\xi_i, \qquad \alpha = \frac{1-b}{1-\tau}, \quad \xi_i \sim \mathcal{N}(0, I_d). \tag{17}$$

For a general bridge clock κ, keep a, b, τ in physical time when computing S and sampling resets, and apply (17) with b, τ replaced by $\kappa(b), \kappa(\tau)$ in α and the Gaussian variance. This is the construction used in Algorithm 2 and does not require κ to be invertible. Alternatively, to sample resets directly in bridge time $u = \kappa(t)$, suppose κ is strictly increasing with $\kappa'(t) > 0$ for $0 < t < 1$. The reset rate per unit bridge time is then

$$\lambda_{\text{bridge}}(u) = \frac{\lambda(\kappa^{-1}(u))}{\kappa'(\kappa^{-1}(u))}, \qquad 0 < u < 1,$$

where λ remains the physical-time rate. In particular, $\int_{\kappa(a)}^{\kappa(b)} \lambda_{\text{bridge}}(u)\, du = \int_a^b \lambda(t)\, dt$, so both clocks give the same no-reset probability. For $\kappa(t) = t^2$, the baseline physical rate $2t/(1-t^2)$ becomes $\lambda_{\text{bridge}}(u) = 1/(1-u)$, whereas the physical rate $1/(1-t)$ becomes $1/[2\sqrt{u}(1-\sqrt{u})]$.

The post-reset drift uses m_i, not the last sampled embedding as an additional hidden endpoint. Predictions are recomputed at the next grid point. This uses one contextual backbone call per interval and no terminal prediction call. Freezing state-dependent predictions, truncating before the endpoint, and decoding at the cutoff are numerical approximations.

C.2 SIMULATION-FREE TRAINING

Algorithm 1 samples training states directly from the conditional law in Lemma A.1, using bridge time $\kappa(t)$. Conditional on Y and t, reset times and Gaussian draws are independent across positions; examples are sampled independently. The atom $\tau_i = 0$ represents no reset; the predictor observes only x and t, without reset times or indicators. Gradients pass through the sampled states to the embedding parameters.

Algorithm 1 Simulation-free training of the RJD LOO predictor

Require: Data law p_{data}; trainable predictor $q_{\theta,i,t}$ and embedding e;
bridge clock κ, reset rate λ, noise scales σ_s, σ_d;
batch size B and training cutoff $0 < t_{\max} < 1$.

while training budget remains **do**
  for $b = 1, \dots, B$ **do**
    Draw $Y \sim p_{\mathrm{data}}$, $t \sim \mathrm{Unif}(0, t_{\max})$; set $u \leftarrow \kappa(t)$.
    for $i = 1, \dots, L$ **do**
      Draw $\tau_i \in [0, t]$ with $\mathbb{P}(\tau_i \le h) = S(h, t)$ for $0 \le h \le t$.
      Draw $\xi_i \sim \mathcal{N}(0, I_d)$ independently.
      if $\tau_i = 0$ **then**
        $x_i \leftarrow ue(Y_i) + \sqrt{(1-u)^2\sigma_s^2 + \sigma_d^2 u(1-u)}\,\xi_i$.
      else
        $x_i \leftarrow e(Y_i) + \sigma_d \sqrt{\dfrac{(u - \kappa(\tau_i))(1-u)}{1 - \kappa(\tau_i)}}\,\xi_i$.
      end if
    end for
    One forward pass gives $q_{\theta,i,t}(\cdot \mid x_{-i})$ for all i.
    $\mathcal{L}_b \leftarrow -\dfrac{1}{L}\sum_{i=1}^{L} \log q_{\theta,i,t}(Y_i \mid x_{-i})$.
  end for
  Update θ and the embedding parameters using $B^{-1}\sum_{b=1}^{B} \mathcal{L}_b$.
end while

The Text8 experiments use $t_{\max} = 0.999$. For $\lambda(t) = \kappa'(t)/(1 - \kappa(t))$, the equivalent mixture (13) can also be sampled directly. The unrestricted control uses the same state sampler and replaces $q_{\theta,i,t}$ by $p_{\theta,i,t}$ in the loss. Pure diffusion uses $\lambda = 0$ in both training and sampling.

C.3 REPEATED-RESET GENERATION

Algorithm 2 uses physical reset times and applies (17) in bridge time. The local likelihood $\ell_{i,t}$ is defined in Proposition 3.3 and must match the training path and embedding. Appendices A.2 and A.3 give its Gaussian-mixture form and evaluation. All predictions are computed from the state at the interval's left endpoint and held fixed throughout that interval. Reset times, destination draws, and Gaussian noises are independent across positions conditional on those predictions.

Algorithm 2 RJD generation with LOO resets and completed or raw LOO drift

Require: Trained $q_{\theta,i,t}$ and e; $\kappa, \lambda, \sigma_s, \sigma_d$;
grid $0 = t_0 < \cdots < t_M < 1$; drift choice; local likelihood $\ell_{i,t}$ if completed.

Draw $x_i \sim \mathcal{N}(0, \sigma_s^2 I_d)$ independently for $i = 1, \ldots, L$.
for $j = 0, \ldots, M-1$ **do**
  Set $s \leftarrow t_j$, $r \leftarrow t_{j+1}$.
  One forward pass gives $q_i(a) \leftarrow q_{\theta,i,s}(a \mid x_{-i})$ for all i, a.
  if using completed drift (RJD-BC-D) **then**
    $m_i \leftarrow \dfrac{\sum_{a \in \mathcal{V}} q_i(a) \ell_{i,s}(x_i \mid a) e(a)}{\sum_{a \in \mathcal{V}} q_i(a) \ell_{i,s}(x_i \mid a)}$ for all i.
  else ▷ raw LOO drift (RJD-LOO)
    $m_i \leftarrow \sum_{a \in \mathcal{V}} q_i(a) e(a)$ for all i.
  end if
  for $i = 1, \ldots, L$ **do**
    Draw $\tau_i \in [s, r]$ with $\mathbb{P}(\tau_i \le h) = S(h, r)$ for $s \le h \le r$.
    if $\tau_i = s$ **then** ▷ no reset
      $z_i \leftarrow x_i$.
    else
      Draw $A_i \sim q_i$; set $z_i \leftarrow e(A_i)$.
    end if
    Set $\alpha_i \leftarrow \dfrac{1 - \kappa(r)}{1 - \kappa(\tau_i)}$; draw $\xi_i \sim \mathcal{N}(0, I_d)$ independently.
    $x_i' \leftarrow m_i + \alpha_i (z_i - m_i) + \sigma_d \sqrt{(\kappa(r) - \kappa(\tau_i))\alpha_i}\, \xi_i$.
  end for
  $x \leftarrow x'$.
end for
Decode $\widehat{Y}_i \in \arg\min_{a \in \mathcal{V}} \|x_i - e(a)\|^2$ for each i; **return** $\widehat{Y}$.

The law of τ_i has an atom of mass $S(s, r)$ at s and unconditional density $\lambda(h) S(h, r)$ on (s, r). Thus each interval includes any number of resets through its last-reset law, and every coordinate remains eligible for later resets. The algorithm uses M network evaluations and no terminal prediction call. Raw LOO drift is the empirical variant and is not covered by Proposition 3.2.

D Baseline paths and information access

The pure-jump reference starts with every coordinate unresolved, commits to $e(Y_i)$ at rate $\kappa'(t)/(1 - \kappa(t))$, and remains fixed after commitment. The baseline experiments use $\kappa(t) = t^2$. Its augmented state records commitment. At an unresolved position the local state carries no token information, so $p_{i,t}$ and $q^\star_{i,t}$ agree there in the idealized model. Their learned architectures and losses can nevertheless behave differently.

Pure diffusion follows (3) with $\lambda = 0$. Its conditional law is $\mathcal{N}(te(Y_i), [(1-t)^2 \sigma_s^2 + \sigma_d^2 t(1-t)] I_d)$. A predictor of $p_{i,t}$ or exact likelihood-completed LOO supplies its drift mean; raw LOO is an empirical ablation without that justification. Absorbing jump diffusion follows this bridge until its first reset, then holds both the anchor and an observed commitment indicator fixed. Completion for that control includes the observed status. Proposition 3.2 concerns repeated resets without such an indicator and is not its proof.

The numerical experiments use the time change $\kappa(t) = t^2$ for the continuous bridge.

E Evaluation details

E.1 Metrics

Word validity and diversity. Words are sequences of lowercase letters a–z, separated by spaces or other non-letter characters. Word fragments at sample boundaries are retained. For each seed, let $\mathcal{W}_\ell$

be the multiset of generated words of length at least $\ell \in \{5, 6\}$, and $\mathcal{D}$ the fixed dictionary from the Text8 test split. Define

$$V_\ell = \frac{\sum_{w \in \mathcal{W}_\ell} \mathbf{1}\{w \in \mathcal{D}\}}{|\mathcal{W}_\ell|},$$
$$D_\ell = \big|\{w : w \in \mathcal{W}_\ell,\ w \in \mathcal{D}\}\big|.$$

Compare D_ℓ only at matched sample counts and sequence lengths.

Effective valid-word count. If $f_\ell(w)$ is the relative frequency of w among valid words in $\mathcal{W}_\ell$, define

$$E_\ell = \exp\left(- \sum_{w \in \mathcal{D}:\, f_\ell(w) > 0} f_\ell(w) \log f_\ell(w) \right),$$

setting $E_\ell = 0$ when no valid words occur.

Exact duplicates and training overlap. The duplicate fraction is one minus the number of distinct complete sequences divided by the sample count. The exact training-match fraction is the fraction of generated sequences occurring as contiguous length-128 segments anywhere in the full training text, including across stored row boundaries. It does not test partial or approximate copying.

Character 4-gram divergence. Let $C_{\text{gen}}(g)$ and $C_{\text{ref}}(g)$ count overlapping character 4-grams g (including spaces) in generated samples and the validation reference. With

$$\mathcal{S} = \{g : C_{\text{gen}}(g) + C_{\text{ref}}(g) > 0\}, \qquad \varepsilon = 10^{-8},$$

use the smoothed distributions

$$P(g) = \frac{C_{\text{gen}}(g) + \varepsilon}{\sum_{h \in \mathcal{S}} C_{\text{gen}}(h) + \varepsilon|\mathcal{S}|}, \qquad Q(g) = \frac{C_{\text{ref}}(g) + \varepsilon}{\sum_{h \in \mathcal{S}} C_{\text{ref}}(h) + \varepsilon|\mathcal{S}|}$$

and report

$$\text{JSD}_4 = \frac{1}{2} \sum_{g \in \mathcal{S}} P(g) \log \frac{P(g)}{M(g)} + \frac{1}{2} \sum_{g \in \mathcal{S}} Q(g) \log \frac{Q(g)}{M(g)}, \qquad M(g) = \frac{P(g) + Q(g)}{2}.$$

All logarithms are natural, so $0 \le \text{JSD}_4 \le \log 2$. Counts never cross generated-sample boundaries; the reference is contiguous across stored chunks. The main 4,096-sample comparison and Appendix G.8 use validation characters $[3{,}000{,}000 : 4{,}000{,}000]$. Unless stated otherwise, the separate 1,024-sample ablations use the first 10^6 validation characters. Compare scores only at matched reference slices and sample counts.

Aggregation across seeds. Compute each metric on each seed's complete sample set. For values r_s from training seeds $s = 0, 1, 2$, report $\bar{r} \pm \text{SD}(r)$, where

$$\bar{r} = \frac{1}{3} \sum_{s=0}^{2} r_s, \qquad \text{SD}(r) = \sqrt{\frac{1}{2} \sum_{s=0}^{2} (r_s - \bar{r})^2}.$$

All transformations, including the exponential defining E_ℓ, precede averaging. Report V_ℓ as percentages and round count statistics only for display.

F FOUR-TOKEN PARITY DIAGNOSTIC

For Table 1, we use $\mathcal{V} = \{0, 1\}$, $L = 4$, scalar embeddings $e(a) = 2a - 1$, and p_{data} uniform on $\{y \in \mathcal{V}^4 : \sum_i y_i \equiv 0 \pmod 2\}$. Each token is marginally uniform, but the other three determine it exactly. A generated string is invalid if it has odd parity; independently sampling the four marginal distributions gives 50% invalid strings. We use $\lambda(t) = (1 - t)^{-1}$ and train equally sized MLPs for 250 updates. Table 1 reports cross-entropy and the percentage of invalid strings, averaged over three seeds with 8,192 generated strings per seed.

G TEXT8 EXPERIMENTAL DETAILS AND ADDITIONAL RESULTS

G.1 INPUT-ISOLATED BLOCK DETAILS

The positionwise update in (9) is

$$F_\ell(h, a, t) = h + a + \mathrm{FF}_\ell(\mathrm{LN}_\ell(h + a), t)\,.$$

Here LN_ℓ is positionwise layer normalization and FF_ℓ is a positionwise feed-forward network. Time conditioning and residual gates are included in Attn_ℓ and FF_ℓ; attention also includes positionwise normalization of its queries and sources.

G.2 TRAINING AND SAMPLING PROTOCOL

We use 90 million training characters and a nominal five-million-character validation split of Text8 (Mahoney, 2006), divided into length-128 sequences with incomplete chunks discarded. The alphabet consists of the 26 lower-case letters and space. Each model is trained for 60,000 updates with batch size 512 and uniform training times on $[0, .999]$. The endpoint cross-entropy includes all positions. AdamW uses learning rate 10^{-3}, no warmup or weight decay, coefficients $(0.9, 0.999)$, numerical constant 10^{-8}, and gradient-norm clipping at one. Evaluation uses the network and embedding EMA with decay 0.9999. Training and evaluation use FP32 and the math attention backend.

The shared backbone has eight blocks of width 256, eight attention heads, 32 time features, AdaLN-Zero conditioning (Peebles & Xie, 2023), SwiGLU width 704 (Shazeer, 2020), and dropout 0.1. The LOO and unrestricted variants have the same 6,878,880 trainable parameters, including the normalized 27×64 embedding matrix. The LOO construction excludes the local input through the fixed-source keys and values in (9); it is not obtained by masking only the diagonal of ordinary multilayer self-attention. DFM replaces the geometric input encoder with a token-plus-mask lookup and has 6,867,616 parameters, approximately 0.164% fewer. Its backbone after the input encoder and its token-level conditional path match EPJ.

The continuous bridge uses the fixed schedule $\kappa(t) = t^2$. During continuous motion, the conditional drift in physical time is $2t(e(y_i) - x_i)/(1 - t^2)$ and the diffusion coefficient is $\sigma_d\sqrt{2t}$. The jump-schedule comparisons keep this bridge fixed and vary only the physical-time reset rate $\lambda_\gamma(t) = \gamma t^{\gamma-1}/(1 - t^\gamma)$, $\gamma \in \{1, 2, 4\}$.

Unless specified otherwise, we use the cosine grid

$$t_j = .999\left[1 - \cos\left(\frac{\pi j}{2M}\right)\right], \qquad j = 0, \ldots, M,$$

where M is the number of predictor intervals. NFE counts actual backbone evaluations, including terminal predictions for unabsorbed coordinates. Pure diffusion and RJD are decoded to the nearest token embedding at the cutoff; absorbing jump models use the stated token rule for remaining unabsorbed positions. The conditional-path comparisons use temperature one, CAT for jumps, and no corrector; the separately labelled TOPP variant changes only reset destinations. AJD-BC-D and AJD-BC reuse the AJD-LOO checkpoints; they change posterior routing without additional training or backbone evaluations. Their local likelihood is Gaussian at an unabsorbed coordinate and includes the token atom and observed status after absorption. PD-BC similarly reuses PD-LOO.

Repeated resets. The RJD baseline uses (3) with reset rate $1/(1 - u)$ in path time $u = \kappa(t) = t^2$, so each coordinate has an independent reset clock with physical-time intensity $\kappa'(t)/(1 - \kappa(t)) = 2t/(1 - t^2)$. Training samples follow (13) with $u = \kappa(t)$; no absorption indicator is observed. RJD-LOO and RJD-BC-D reuse the same repeated-reset LOO checkpoints. For RJD-BC-D, the local likelihood is evaluated using (15), without reset-time quadrature. Sampling freezes the predictions on each grid interval and uses the last-reset update (17) between $\kappa(t_j)$ and $\kappa(t_{j+1})$. The last-reset law integrates out any earlier resets within that interval; diffusion continues afterwards. Each interval costs one network evaluation, and nearest-embedding decoding costs none. RJD-BC-D has the prescribed marginals with exact posteriors and dynamics; learned prediction and finite-step sampling introduce approximations. RJD-LOO additionally replaces the drift posterior by $q_{\theta,i,t}$ and is an empirical ablation. RJD and AJD use path-specific training, different observed states, and different terminal rules, so their comparison does not isolate the effect of absorption alone.

Main evaluation. The nine settings in Table 2 use 4,096 sequences per training seed at 128 NFE, generated on A100 GPUs in batches of 64 with TF32 disabled. Batch $b \in \{0, \ldots, 63\}$ for training seed s uses generation seed $625000 + 1000s + b$. RJD and PD use 128 predictor intervals; DFM and EPJ use 127 intervals and one terminal prediction. Actual backbone calls are checked in every batch. The checkpoints are unchanged across sampling variants; fresh generation adds no training replicates.

The main comparison uses validation characters $[3{,}000{,}000 : 4{,}000{,}000]$ as the 4-gram reference. The separate 1,024-sample ablations use the first million validation characters unless stated otherwise. The fixed word dictionary is derived from the test split and has been reused during development; this is not an untouched test evaluation. Dictionary membership is a lexical metric, not a measure of sentence grammaticality. We compute each metric separately for each seed, then report its arithmetic mean and sample standard deviation across seeds 0, 1, and 2. Repeated evaluations of one checkpoint are not additional training seeds. The path, decoding, and clock comparisons have separate evaluation records, which accounts for small differences between their CAT results.

G.3 Real-text calibration and supplementary diagnostics

Table 4 adds V_6 and real-text controls to the main 4,096-sample comparison. The model rows use the same generated sequences as Table 2. For real text, we take three disjoint blocks of 128×4096 validation characters, starting at offsets $1{,}100{,}000$, $1{,}700{,}000$, and $2{,}300{,}000$, and divide each into length-128 passages. These blocks are disjoint from the common 4-gram reference $[3{,}000{,}000 : 4{,}000{,}000]$. We report mean and sample standard deviation across training seeds for models and across text blocks for the real controls.

Table 4: Calibration on 4,096 length-128 sequences per replicate. Model rows use three training seeds at 128 NFE; real text uses three validation blocks. All rows share the dictionary and validation reference. Counts are rounded to integers. Model rows repeat the main comparison and add V_6.

Method	V_5 (%)	V_6 (%)	D_5	E_5	JSD_4
RJD	28.96 ± 0.62	17.54 ± 0.68	3051 ± 43	911 ± 21	0.0657 ± 0.0007
RJD-LOO	75.60 ± 0.89	70.26 ± 1.08	6596 ± 66	2411 ± 48	0.0580 ± 0.0011
RJD-BC-D	60.16 ± 1.18	51.59 ± 1.43	6783 ± 13	2843 ± 54	0.0672 ± 0.0015
DFM	44.62 ± 0.56	34.34 ± 0.96	4798 ± 159	1501 ± 150	0.0536 ± 0.0002
EPJ	44.96 ± 0.47	35.11 ± 0.56	4971 ± 75	1703 ± 73	0.0540 ± 0.0002
PD	29.47 ± 1.49	18.29 ± 1.74	3137 ± 191	964 ± 83	0.0664 ± 0.0005
PD-LOO	85.49 ± 0.67	80.55 ± 0.81	5696 ± 58	1944 ± 36	0.0653 ± 0.0012
PD-BC	36.34 ± 0.51	24.27 ± 0.65	4499 ± 7	1637 ± 23	0.0691 ± 0.0004
RJD-LOO, TOPP	79.70 ± 0.68	74.22 ± 0.88	6611 ± 68	2368 ± 29	0.0545 ± 0.0009
Real Text8	88.28 ± 0.85	86.83 ± 1.35	7204 ± 1055	2380 ± 415	0.0585 ± 0.0219

RJD-BC-D has higher D_5 and E_5 than RJD-LOO, while RJD-LOO has higher validity and lower four-gram divergence. TOPP improves RJD-LOO's validity and divergence with similar D_5 but slightly lower E_5. Real-text scores vary across blocks, and their four-gram divergence can exceed that of generated sets; they do not define a universal lower bound.

Within each of the 27 generated sets of 4,096 samples, all complete sequences are distinct, and none exactly matches a contiguous 128-character segment of the training text. This does not exclude partial or approximate copying. Real-text controls have an exact training-match fraction of $0.285\% \pm 0.322\%$ across blocks. These diagnostics concern lexical statistics, short-range character agreement, and exact repetition, rather than grammaticality or coherence.

G.4 Absorbing jump diffusion

Absorbing jump diffusion (AJD) follows the time-changed Brownian bridge until its first jump, then remains at the sampled embedding. Its rate is $2t/(1-t^2)$ while unabsorbed. We compare unrestricted prediction in both drift and jump probabilities (AJD), LOO prediction in both (AJD-LOO), completion only in the drift (AJD-BC-D), and completion in both terms (AJD-BC). The full predictor observes the absorption indicators; $q_{\theta,i,t}$ excludes both $X_{i,t}$ and its indicator. Completion uses the local likelihood of this absorbing path.

Table 5 compares the four AJD variants at 128 NFE, using 1,024 samples per seed and the first million validation characters as reference. LOO improves all four reported metrics over unrestricted AJD and both completion variants. This is a separate prediction ablation; its D_5 and JSD_4 values should not be compared directly with the 4,096-sample main table.

Table 5: AJD prediction ablation at 128 NFE and 1,024 samples per seed. We report mean $\pm$ sample standard deviation over three training seeds; D_5 counts are rounded to the nearest integer. All paths use $\kappa(t) = t^2$, jumps use CAT, and predictions have temperature one.

Method	V_5 (%) $\uparrow$	V_6 (%) $\uparrow$	$D_5 \uparrow$	$\mathrm{JSD}_4 \downarrow$
AJD	29.37 ± 2.36	18.38 ± 2.70	1339 ± 97	0.0919 ± 0.0020
AJD-LOO	$\mathbf{68.62} \pm 0.34$	$\mathbf{58.75} \pm 0.43$	$\mathbf{2910} \pm 77$	$\mathbf{0.0761} \pm 0.0014$
AJD-BC-D	53.68 ± 0.89	41.71 ± 1.08	2722 ± 42	0.0879 ± 0.0009
AJD-BC	35.86 ± 0.08	23.70 ± 0.46	1816 ± 16	0.0947 ± 0.0003

G.5 Jump sampling rules and complete comparisons

The five jump rules reuse each path's 60,000-update EMA checkpoints without retraining. TEMP applies temperature 0.7 only to jump destinations; TOPP truncates and renormalizes the destination probabilities at cumulative mass 0.9. MAX takes their argmax. CAT and CONF sample destinations from the original $q_{\theta,i,t}$. AJD-LOO and RJD-LOO retain the temperature-one mean $\sum_a q_{\theta,i,t}(a)e(a)$ in their drift for every rule. All comparisons in this subsection use 1,024 length-128 sequences per seed, the same dictionary, the first million validation characters as reference, the cosine grid, and no corrector. The 4,096-sample CAT/TOPP comparison is reported separately in Table 2. EPJ and AJD-LOO include terminal prediction calls for unabsorbed positions in NFE; RJD-LOO uses nearest-embedding decoding at $t = .999$ with no extra call. Their training distributions, observed states, and terminal rules differ, so cross-path comparisons do not isolate absorption alone.

Confidence selection. For EPJ and AJD-LOO, CONF draws the number of jumps from the original independent clocks and assigns them to the most confident unabsorbed positions; committed positions remain fixed. For RJD-LOO, all positions remain eligible on every interval. On the path-time interval $[u, v]$, we first sample the independent reset indicators and last-reset times from Appendix C. Their count satisfies

$$K \sim \mathrm{Binomial}\left(L, \frac{v-u}{1-u}\right).$$

Here K counts positions with at least one reset, not the total number of Poisson events. CONF chooses the K positions with largest $\max_a q_{\theta,i,t}(a)$, breaking ties with fresh independent uniforms, and reassigns the sampled last-reset times to them. The multiset of times is preserved, and their assignment does not depend on the time values. The interval update (17) then includes diffusion after each reset. The drift uses the unchanged LOO mean, rather than the last sampled token as a hidden endpoint. This finite-grid rule preserves the sampled count but changes the position-specific reset probabilities; it is not a marginal-preserving modification.

At 128 NFE, CAT and CONF both produce about 4.19 reset-containing intervals per position. The fraction of positions reset in multiple intervals is 96.5% for CAT and 48.9% for CONF; CONF's selected positions have mean predicted confidence 0.994. Thus CONF concentrates resets on fewer positions. This is consistent with, but does not establish the cause of, the lower word diversity. These diagnostics count intervals with resets, not individual Poisson events within an interval.

Complete sampling results. Tables 6–8 report all five rules, including the effective word count E_5. At 128 NFE, AJD-LOO with TOPP has lower JSD_4 than RJD-LOO with TOPP, but lower validity and diversity; neither path improves every metric. EPJ and AJD-LOO have all three training seeds at every budget. For the RJD-LOO sampling study, 32 and 128 NFE have seeds 0, 1, and 2; the 512-NFE results below are preliminary and use only seeds 0 and 1. Seed 2 at 512 NFE is not available in this snapshot. The original CAT path comparison in Table 9 has all three seeds; it is distinct from this incomplete five-rule comparison.

At 128 NFE, MAX raises RJD-LOO validity over CAT but reduces E_5 by about 25%; CONF reduces it by about 19%. None of the completed RJD-LOO evaluations contains duplicate full sequences; the reduction concerns word diversity.

For budget comparisons involving 512 NFE, we match seeds 0 and 1 at every budget. RJD-LOO with TOPP has $V_5 = 70.27\%, 79.96\%, 82.31\%$ and $\mathrm{JSD}_4 = 0.0874, 0.0771, 0.0750$ at 32, 128, and 512 NFE. At 512 NFE, PD-LOO on these same two seeds has $V_5 = 85.37\%$, $D_5 = 2904$, $E_5 = 1432$, and $\mathrm{JSD}_4 = 0.0811$, compared with $D_5 = 3210$ and $E_5 = 1654$ for RJD-LOO with TOPP. The tradeoff remains; these two-seed estimates should not be compared directly with the three-seed means as if the seed sets were identical.

Table 6: EPJ sampling at 32, 128, and 512 NFE with 1,024 sequences per seed. Mean ± sample standard deviation; count metrics are rounded to integers. Best means within each budget are bold. All budgets use three training seeds.

Rule	V_5 (%) ↑	V_6 (%) ↑	D_5 ↑	E_5 ↑	JSD_4 ↓
32 NFE, 3 seeds					
CAT	35.22 ± 0.50	26.01 ± 0.32	1804 ± 20	967 ± 8	0.0996 ± 0.0007
TEMP	53.05 ± 0.47	41.59 ± 0.66	1915 ± 60	695 ± 46	0.0875 ± 0.0015
TOPP	44.19 ± 0.45	34.60 ± 0.59	$\mathbf{2021} \pm 29$	$\mathbf{1020} \pm 47$	$\mathbf{0.0856} \pm 0.0005$
MAX	$\mathbf{90.58} \pm 0.74$	$\mathbf{76.16} \pm 1.47$	363 ± 14	13 ± 1	0.3708 ± 0.0034
CONF	9.78 ± 0.15	5.86 ± 0.29	665 ± 37	452 ± 42	0.2223 ± 0.0016
128 NFE, 3 seeds					
CAT	44.75 ± 0.41	34.85 ± 0.59	2139 ± 47	$\mathbf{1098} \pm 46$	0.0817 ± 0.0006
TEMP	64.52 ± 0.49	53.17 ± 0.64	2117 ± 25	712 ± 42	0.0825 ± 0.0009
TOPP	55.95 ± 0.91	45.92 ± 1.18	$\mathbf{2323} \pm 42$	1092 ± 31	$\mathbf{0.0722} \pm 0.0006$
MAX	$\mathbf{96.87} \pm 0.29$	$\mathbf{89.50} \pm 1.53$	329 ± 36	11 ± 2	0.3932 ± 0.0104
CONF	26.24 ± 0.82	19.23 ± 0.63	1393 ± 45	720 ± 25	0.1397 ± 0.0002
512 NFE, 3 seeds					
CAT	47.58 ± 0.57	37.56 ± 0.88	2220 ± 42	1131 ± 86	0.0776 ± 0.0005
TEMP	68.25 ± 0.37	57.23 ± 0.69	2192 ± 45	718 ± 54	0.0817 ± 0.0024
TOPP	58.78 ± 0.14	49.03 ± 0.34	$\mathbf{2427} \pm 36$	$\mathbf{1149} \pm 32$	$\mathbf{0.0705} \pm 0.0005$
MAX	$\mathbf{98.22} \pm 0.23$	$\mathbf{93.09} \pm 0.26$	309 ± 3	10 ± 0	0.4025 ± 0.0058
CONF	49.74 ± 0.11	41.54 ± 0.93	2046 ± 48	842 ± 95	0.0899 ± 0.0018

Table 7: AJD-LOO sampling at 32, 128, and 512 NFE with 1,024 sequences per seed. Mean ± sample standard deviation; count metrics are rounded to integers. Best means within each budget are bold. All budgets use three training seeds.

Rule	V_5 (%) ↑	V_6 (%) ↑	D_5 ↑	E_5 ↑	JSD_4 ↓
32 NFE, 3 seeds					
CAT	58.94 ± 0.68	48.19 ± 0.95	2796 ± 29	1567 ± 23	0.0833 ± 0.0013
TEMP	66.04 ± 1.12	55.44 ± 1.82	2859 ± 90	1481 ± 70	$\mathbf{0.0774} \pm 0.0015$
TOPP	63.48 ± 0.41	52.89 ± 1.08	$\mathbf{2904} \pm 29$	$\mathbf{1578} \pm 43$	0.0776 ± 0.0011
MAX	74.52 ± 0.21	64.19 ± 0.31	2640 ± 68	1162 ± 27	0.0845 ± 0.0006
CONF	$\mathbf{77.41} \pm 0.93$	$\mathbf{70.26} \pm 1.01$	2828 ± 65	1300 ± 81	0.0813 ± 0.0013
128 NFE, 3 seeds					
CAT	68.34 ± 0.42	58.47 ± 0.39	2924 ± 42	$\mathbf{1508} \pm 31$	0.0758 ± 0.0003
TEMP	73.41 ± 0.52	63.87 ± 0.42	2896 ± 23	1392 ± 20	0.0743 ± 0.0012
TOPP	71.75 ± 0.73	62.17 ± 0.75	$\mathbf{2942} \pm 41$	1484 ± 24	$\mathbf{0.0729} \pm 0.0007$
MAX	80.26 ± 0.17	71.41 ± 0.67	2650 ± 36	1078 ± 29	0.0830 ± 0.0021
CONF	$\mathbf{82.75} \pm 0.20$	$\mathbf{76.12} \pm 0.28$	2882 ± 55	1264 ± 52	0.0807 ± 0.0025
512 NFE, 3 seeds					
CAT	69.80 ± 1.26	60.05 ± 1.27	2929 ± 28	$\mathbf{1443} \pm 30$	0.0745 ± 0.0006
TEMP	75.17 ± 0.98	65.90 ± 1.16	2869 ± 5	1339 ± 20	0.0740 ± 0.0007
TOPP	73.35 ± 0.95	63.87 ± 1.19	$\mathbf{2958} \pm 8$	1424 ± 37	$\mathbf{0.0720} \pm 0.0009$
MAX	81.75 ± 1.10	73.18 ± 1.29	2625 ± 35	1052 ± 44	0.0845 ± 0.0017
CONF	$\mathbf{83.69} \pm 0.60$	$\mathbf{76.83} \pm 0.68$	2886 ± 61	1216 ± 59	0.0818 ± 0.0022

Table 8: RJD-LOO sampling at 32, 128, and 512 NFE with 1,024 sequences per seed. Mean ± sample standard deviation; count metrics are rounded to integers. Best means within each budget are bold. The 32- and 128-NFE results use three seeds; 512 NFE uses only seeds 0 and 1. The nonzero CAT standard deviation for JSD_4 at 512 NFE is shown in scientific notation.

Rule	V_5 (%) ↑	V_6 (%) ↑	D_5 ↑	E_5 ↑	JSD_4 ↓
32 NFE, 3 seeds					
CAT	64.39 ± 1.34	58.06 ± 1.28	2819 ± 29	1597 ± 48	0.0979 ± 0.0024
TEMP	72.70 ± 1.00	65.82 ± 1.12	$\mathbf{2963} \pm 33$	1601 ± 34	0.0857 ± 0.0013
TOPP	69.90 ± 0.86	63.34 ± 1.06	2960 ± 36	$\mathbf{1654} \pm 58$	0.0882 ± 0.0013
MAX	$\mathbf{80.64} \pm 0.71$	$\mathbf{73.55} \pm 0.96$	2734 ± 29	1282 ± 65	0.0865 ± 0.0017
CONF	77.20 ± 0.91	71.03 ± 1.64	2718 ± 8	1312 ± 42	$\mathbf{0.0847} \pm 0.0009$
128 NFE, 3 seeds					
CAT	75.76 ± 1.04	70.61 ± 1.26	3118 ± 41	$\mathbf{1669} \pm 42$	0.0824 ± 0.0018
TEMP	81.23 ± 0.38	75.63 ± 0.60	3100 ± 43	1576 ± 40	$\mathbf{0.0775} \pm 0.0016$
TOPP	79.52 ± 0.89	74.10 ± 0.77	$\mathbf{3149} \pm 13$	1659 ± 21	0.0776 ± 0.0010
MAX	$\mathbf{86.29} \pm 0.38$	$\mathbf{80.20} \pm 0.40$	2814 ± 66	1250 ± 103	0.0834 ± 0.0010
CONF	84.21 ± 0.41	78.23 ± 0.65	2876 ± 12	1352 ± 26	0.0801 ± 0.0005
512 NFE, 2 seeds					
CAT	79.22 ± 0.10	74.47 ± 0.22	3203 ± 1	$\mathbf{1689} \pm 4$	$0.0781 \pm 4.19 \times 10^{-6}$
TEMP	83.06 ± 0.37	77.82 ± 0.27	3143 ± 18	1558 ± 22	0.0755 ± 0.0011
TOPP	82.31 ± 0.76	77.29 ± 0.85	$\mathbf{3210} \pm 4$	1654 ± 9	$\mathbf{0.0750} \pm 0.0004$
MAX	$\mathbf{86.85} \pm 0.54$	$\mathbf{80.94} \pm 0.38$	2776 ± 83	1209 ± 116	0.0826 ± 0.0016
CONF	85.92 ± 0.28	80.36 ± 0.85	2852 ± 62	1330 ± 88	0.0793 ± 0.0016

G.6 Additional network budgets

Table 9 gives the complete comparison at 512 NFE and the available 2,048-NFE results. DFM, AJD-BC-D, and RJD-LOO were not evaluated at 2,048 NFE. There is no complete comparison at 32 NFE: PD, PD-BC, AJD, AJD-BC, AJD-BC-D, RJD, and RJD-BC-D have no such evaluation in this set; the separate RJD-LOO sampling results are in Table 8. All values below use 1,024 samples per seed. At 512 NFE, RJD-LOO has $D_5 = 3193 \pm 17$, compared with 2939 ± 23 for AJD-LOO and 3209 ± 23 for RJD-BC-D. Thus the validity advantage of RJD-LOO over RJD-BC-D does not imply higher word diversity: RJD-BC-D has slightly higher D_5 and higher E_5 at this budget (1911 ± 22 versus 1686 ± 6).

At 128 NFE, EPJ with MAX gives 96.87% word validity but only about 11 effective valid-word types, compared with about 1,098 for CAT. Here the effective count is $\exp(H)$, where H is the empirical entropy of valid words of length at least five; it is calculated per seed before averaging. This concentration explains why a high V_5 need not imply a close match to the reference text distribution.

Table 9: Additional budgets for the conditional-path comparison. Mean ± sample standard deviation over three seeds, with 1,024 samples per seed, temperature one, CAT, and $\kappa(t) = t^2$. All checkpoints are evaluated after 60,000 training updates.

NFE	Method	V_5 (%) ↑	V_6 (%) ↑	JSD_4 ↓
512	DFM	47.09 ± 0.37	36.47 ± 0.83	0.0787 ± 0.0018
512	EPJ	47.52 ± 0.92	37.42 ± 1.27	0.0778 ± 0.0002
512	PD	31.48 ± 1.30	20.10 ± 1.71	0.0898 ± 0.0016
512	PD-LOO	$\mathbf{86.10} \pm 1.29$	$\mathbf{81.12} \pm 1.70$	0.0817 ± 0.0012
512	PD-BC	37.59 ± 0.34	25.78 ± 0.40	0.0938 ± 0.0003
512	AJD	31.51 ± 1.21	20.06 ± 1.84	0.0885 ± 0.0023
512	AJD-LOO	70.06 ± 1.04	60.35 ± 1.04	$\mathbf{0.0750} \pm 0.0005$
512	AJD-BC-D	54.96 ± 0.49	43.26 ± 0.20	0.0856 ± 0.0004
512	AJD-BC	37.62 ± 0.82	25.47 ± 1.04	0.0929 ± 0.0007
512	RJD	29.89 ± 0.66	18.76 ± 1.21	0.0897 ± 0.0003
512	RJD-LOO	78.57 ± 1.12	73.56 ± 1.59	0.0793 ± 0.0021
512	RJD-BC-D	65.31 ± 1.59	57.24 ± 1.84	0.0909 ± 0.0027
2048	EPJ	48.57 ± 0.14	38.50 ± 0.61	0.0764 ± 0.0002
2048	PD	31.24 ± 1.67	19.63 ± 2.09	0.0891 ± 0.0014
2048	PD-LOO	$\mathbf{85.70} \pm 0.82$	$\mathbf{80.88} \pm 1.07$	0.0808 ± 0.0021
2048	PD-BC	38.58 ± 0.60	26.53 ± 0.19	0.0934 ± 0.0005
2048	AJD	32.39 ± 2.10	20.44 ± 2.31	0.0891 ± 0.0025
2048	AJD-LOO	70.21 ± 0.52	60.76 ± 0.75	$\mathbf{0.0739} \pm 0.0010$
2048	AJD-BC	38.26 ± 0.38	25.61 ± 0.43	0.0929 ± 0.0003
2048	RJD	30.70 ± 0.58	19.05 ± 0.64	0.0891 ± 0.0006
2048	RJD-BC-D	68.77 ± 0.91	60.97 ± 1.58	0.0864 ± 0.0026

G.7 Jump-time schedules for absorbing jump diffusion

Keeping the continuous schedule $\kappa(t) = t^2$ fixed, we compare independent first-jump times T_i with

$$\mathbb{P}(T_i \le t) = t^\gamma, \qquad \lambda_\gamma(t) = \frac{\gamma t^{\gamma-1}}{1-t^\gamma}, \qquad \gamma \in \{1, 2, 4\}.$$

The baseline is $\gamma = 2$; larger γ delays jumps. For absorbing jump diffusion, each model is trained and sampled with the same γ. An unabsorbed coordinate jumps over $[s, t]$ with probability $(t^\gamma - s^\gamma)/(1 - s^\gamma)$. We keep the cosine time grid and network budget fixed, stop at $t = .999$, and apply the chosen token rule to any remaining unabsorbed positions, counting terminal network evaluations.

At 128 NFE, delaying absorption increases validity and diversity: V_5 rises from 58.07% to 77.10% between $\gamma = 1$ and 4. The lowest mean JSD_4 occurs at $\gamma = 2$, close to $\gamma = 4$ (Table 10).

Table 10: AJD-LOO with CAT, 128 NFE, and 1,024 samples per seed (mean ± sample standard deviation over three seeds). D_5 counts are rounded to the nearest integer. The continuous schedule is fixed at $\kappa(t) = t^2$; the $\gamma = 1, 4$ models are trained with their respective jump clocks.

γ	V_5 (%) ↑	V_6 (%) ↑	D_5 ↑	JSD_4 ↓
1	58.07 ± 1.15	46.86 ± 0.71	2749 ± 64	0.0792 ± 0.0013
2	68.62 ± 0.34	58.75 ± 0.43	2910 ± 77	$\mathbf{0.0761} \pm 0.0014$
4	$\mathbf{77.10} \pm 0.83$	$\mathbf{69.44} \pm 1.08$	$\mathbf{3064} \pm 13$	0.0766 ± 0.0007

Table 11 supplements the 128-NFE clock comparison in Table 10. Increasing γ improves V_5 at both additional budgets. At 32 NFE, $\gamma = 4$ gives a slightly lower mean JSD_4 than $\gamma = 2$; at 512 NFE, $\gamma = 2$ gives the lowest mean. The continuous schedule and numerical grid remain fixed. At the common cutoff, the theoretical unabsorbed fractions are $1 - .999^\gamma$, approximately 0.001, 0.002, and 0.004 for $\gamma = 1, 2, 4$; terminal decoding therefore affects different fractions.

Table 11: Additional budgets for AJD-LOO with CAT and trained jump CDF t^γ. Mean $\pm$ sample standard deviation over three seeds, with 1,024 samples per seed and fixed continuous schedule $\kappa(t) = t^2$.

NFE	γ	V_5 (%) $\uparrow$	V_6 (%) $\uparrow$	$\mathrm{JSD}_4 \downarrow$
32	1	50.72 ± 0.41	39.07 ± 0.70	0.0887 ± 0.0012
32	2	59.05 ± 0.62	48.39 ± 0.77	0.0833 ± 0.0013
32	4	$\mathbf{67.85} \pm 1.35$	$\mathbf{59.32} \pm 1.85$	$\mathbf{0.0830} \pm 0.0010$
512	1	60.32 ± 0.10	49.04 ± 0.17	0.0781 ± 0.0002
512	2	70.06 ± 1.04	60.35 ± 1.04	$\mathbf{0.0750} \pm 0.0005$
512	4	$\mathbf{78.53} \pm 0.72$	$\mathbf{71.09} \pm 1.11$	0.0765 ± 0.0007

G.8 Additional samplers with 4,096 samples per seed

This separate evaluation covers five sampler settings, the same three training seeds, and budgets of 128 and 512 NFE. All checkpoints are the 60,000-update EMA checkpoints. We generate 4,096 sequences per seed with generation seeds $51000 + s$, where $s \in \{0, 1, 2\}$ is the training seed. The dictionary and validation reference $[3{,}000{,}000 : 4{,}000{,}000]$ match the main comparison, but the grids, jump rules, and correction steps below differ. These are complete-sampler comparisons using existing checkpoints, not additional training seeds.

PD-LOO uses a grid uniform in $\kappa(t) = t^2$; the other settings use the cosine grid. All AJD variants retain raw LOO drift. The final setting adds 16 LOO updates at times in $[2/3, 1]$: each update chooses one position per sequence, draws its token from $q_{\theta,i,t}$, and redraws its complete conditional state, including its absorption indicator. These are 16 single-position updates, not 16 sweeps over the sequence. At total budget B, this setting uses $B - 17$ predictor intervals, 16 correction calls, and one terminal call. The selected settings are fixed before this confirmation; they compare complete samplers rather than isolating one change to the conditional path.

Table 12: Additional samplers with 4,096 generated sequences per seed. Mean $\pm$ sample standard deviation over three training seeds; the dictionary and 4-gram reference match the main comparison. All correction and terminal calls count toward NFE.

NFE	Sampler	V_5 (%) $\uparrow$	$\mathrm{JSD}_4 \downarrow$
128	PD-LOO, uniform κ	$\mathbf{85.48} \pm 0.75$	0.0657 ± 0.0011
128	EPJ, TOPP	55.72 ± 0.58	0.0508 ± 0.0009
128	AJD-LOO, $\gamma = 2$, TOPP	72.05 ± 0.35	$\mathbf{0.0487} \pm 0.0002$
128	AJD-LOO, $\gamma = 4$, TOPP	79.00 ± 0.93	0.0543 ± 0.0008
128	AJD-LOO, $\gamma = 4$, CONF + 16 updates	82.87 ± 0.78	0.0582 ± 0.0012
512	PD-LOO, uniform κ	$\mathbf{85.86} \pm 0.72$	0.0646 ± 0.0019
512	EPJ, TOPP	58.81 ± 0.56	0.0487 ± 0.0001
512	AJD-LOO, $\gamma = 2$, TOPP	73.82 ± 0.60	$\mathbf{0.0486} \pm 0.0005$
512	AJD-LOO, $\gamma = 4$, TOPP	80.33 ± 0.67	0.0542 ± 0.0007
512	AJD-LOO, $\gamma = 4$, CONF + 16 updates	83.54 ± 0.95	0.0576 ± 0.0015

G.9 GENERATED TEXT8 SAMPLES

The panels below show the first four generated sequences from training seed 0 for each model in Table 4, using the 60,000-update EMA checkpoints and 128 NFE. They come from the first batch, with generation seed 625000, of the main 4,096-sample evaluations. The real-text panel shows the first four passages from the first calibration block. Every sequence has 128 characters; spelling and spacing are retained, with line wrapping only for display. No sample is selected by quality. Text8 chunks may begin or end inside a word and need not contain complete sentences.

RJD

```
1 e dumb and viner moders in londom ishue eight seven true tradicated in
      rather from the condect one nine isles that citigating an
2  sex mperet copics miles the hzrio seven with e glor out and afforms
      what pertihally obstylor dissogovernation has game millions
3 ore s for scends are and agoin profound whone in the work irs and it
      christ of alkind with the pus assi son the moduls awice sho
4 s soon an ive a rind th for diale orseural our eight freed basc time
      samier unite forcrection was indual numes two zerological w
```

RJD-LOO

```
1 lm an increased at never until one nine zero zero s missed into the
      from the germans seven the size with their but conditions in
2 ne nine from the similar mey lije finding in treaty are federatorial
      from master others leading system and and so two spirit in
3 c wicked as they three poland beyond lanes to with city award and
      stand its black ards s hulm premiere of kelrey could arrive wo
4 lt at the bbc s case of diamond one four three singer had a links
      situationes the bhm trishment both saturated in twins in the t
```

RJD-BC-D

```
1 dm and the six nine microbably the much five zero s protestants the
      from the one one depart on mars with their dint eduction sym
2 ne five root the similar taxation finding of traits are need italian
      crime mo orname ballistine system appsara a yff cld nations
3 nglabour is that finally have a sack antident with no times terrain
      him one which own a coast julians cpu of continues refer com
4 t wae the end a core or ted airlinear used to exclaim nsiek manuse
      igration scales greenish will tried segwod war complete the r
```

DFM

```
1 the geoglopes have cpublic grand children cummon from any alternet tee
      also goes a mitchem abablical hugaleth three of covement
2 lly even products while is olderchears and dandry minstur five last
      invaserabolations und novel tamp artial and theory prejertiv
3 ked place the icon unsoruence a like sotrague are excell museumus very
      proper j five zero fivr kx three eighteen agree mcass as
4 gly class for no fish race humans batant tu merolesc french chances of
      inderstand s s movonecy but estment of the acid and as e
```

EPJ

```
1 ch proootypes whom the diccs and children cummon lock and alternet the
      also goes a discreas biblical quesleth three of constant
2 lly john products would be kide cietre and landon the tur five last
      invoker word it sound constant pourthel in its of t rkjers a
3 ked electrolysi of unborusics a like sourage earloes allo usedmus very
      proper s live articuiar exported either aka meen carious
4 g include for no fish race humans balant toe epollic speech chances of
      inderstand s complexacy bettes term of the acid and acle
```

PD

```
ellow willing printers used an a host qz o rge work wood the
    suveriations in use at mouke the devells web or the taffairable pun
ines technical uperators as a justicially electrbst delphan s also
    litylcha tustman pathinal his most zero six somes posite diag
n to athmic as long a can carediamed in waftgaivil that all ardent of
    whichment of comleptitiea buduals targe base it beespresen
rm nately of brone for a i to st author of bell or or terrable or a
    masteri s anotherain of two time construttle golgia of aomon
```

PD-LOO

```
ther way if and is required and his three in the much lead the unable
    two zero with fresco known explants botany where bible by
in combinatings and known incomes with more volume turkiest south the
    faced back and from a pictures functive in a zero six foll
 family concerned with led the handpoint and of break train later and
    german is in nine door is violence west for three the part
 selling roman holding deputies the threats the root of british
    unificialized in the official companion islanding man and four f
```

PD-BC

```
ry arvasion god be nasourew as whichborre with robbonder external
    planing were highening of sean explated but onothers libdiness
ided mmigh mary and brown in one white clocter over rarring years
    howevered contrast s am ibrans intemulceptic to where firstwir
eft day converses logist diated a insult and required for foltwerna
    decamillies on the amate native abcoaster togethrew the land
mportational civil its families one three e well of hopes uzes it
    inviding to in the old that cormits is standard but ael founli
```

RJD-LOO, TOPP

```
cd and spreaded an generatical one nine zero zero somerged into the
    from the germans seven the life with their all conditions in
ne nine from the similar may life finding in treaty are federatorial
    from master others leading system and and so two shend cond
d passed as they three poland beyond lined to with city award and
    stand its black ards a hulm premiere on the us could argued wo
temit the abdul work os studies one four three loved from a livks in
    party by these largestment both saturated in twins in the t
```

Real Text8

```
n nine seven five ad also malay sailors knew of mauritius as early as
    the one zero th century portuguese sailors one five zero s
even one five one three mauritius was discovered and visited by the
    portuguese between one five zero seven and one five one thre
e an official world map by diego ribeiro described from west to east
    the first island mascarenhas the second santa apolonia and
the third domingo friz one the three islands r union mauritius and
    rodrigues were discovered some years earlier by chance during
```

H ADDITIONAL PREDICTOR CONFIGURATIONS ON TEXT8

H.1 PURE-JUMP CONTROL WITH ORDINARY SELF-ATTENTION

We additionally evaluate a pure-jump predictor with ordinary self-attention, using the matched-layer DDiT control. Queries, keys, and values are recomputed from the evolving hidden states at every layer, and attention includes the diagonal. This differs from both the LOO and unmasked fixed-source predictors in the main comparison. The remaining layers match our backbone: eight blocks of width 256, eight heads of dimension 32, SwiGLU width 704, time conditioning, rotary positional embeddings, normalization, dropout, and learned 64-dimensional token embeddings. The control has 6,878,624 trainable parameters, versus 6,878,880 for structural LOO; the difference is the latter's learned 256-dimensional initial query.

The run uses the same embedded pure-jump path as EPJ: coordinates start at zero, jump with CDF t^2, and remain at their sampled token embeddings. It is trained from scratch on an A100 for 60,000 updates with seed 0, length-128 sequences, batch size 512, uniform training times on $[0, .999]$, and ordinary cross-entropy over all positions. AdamW, learning rate 10^{-3}, FP32/math attention, and EMA decay 0.9999 match the main protocol. No likelihood completion or LOO loss correction is used. The mean training loss over the final 1,000 updates is 1.5089; final EMA validation CE is 1.4885 over all positions, 1.8254 for the per-sequence average over unabsorbed positions, and 5.20×10^{-5} over absorbed positions. These are denoising losses, not language-model likelihoods or generation-quality scores.

Table 13 compares this control with the original EPJ structural-LOO checkpoint for the same training seed. Each evaluation uses the 60,000-update EMA, sampling seed 0, 1,024 sequences, the cosine grid, and categorical destinations at temperature one or 0.7. NFE includes terminal classification of any remaining unabsorbed positions. The validation-reference, lexicon, and character-map hashes agree across all twelve evaluations. These evaluations reuse two trained checkpoints; there is no training-seed standard deviation.

Ordinary self-attention gives higher V_5 and V_6 and lower JSD_4 in all six matched sampling settings, while the word-diversity comparisons are mixed. At 128 NFE and temperature one, V_5 increases from 44.35% to 54.08% and JSD_4 decreases from 0.08215 to 0.07284; both D_5 and E_5 also increase. Lowering the self-attention predictor's sampling temperature to 0.7 raises V_5 to 73.41%, but reduces diversity and increases divergence. At temperature one, increasing the budget from 512 to 2,048 NFE gives little additional validity and reduces D_5 and E_5. No evaluation contains duplicate full sequences.

This provides a stronger pure-jump reference than the structural-LOO EPJ predictor in this seed. At an unabsorbed pure-jump position, the local zero state carries no endpoint information, so the local denoising shortcut that motivates LOO for continuous paths is absent. Structural LOO therefore need not improve this baseline. The result is specific to this single-seed comparison and does not establish an ordering across seeds or continuous paths.

Table 13: Pure-jump predictors with matched layers, 60,000-update EMA checkpoints, training and sampling seed 0, and 1,024 length-128 sequences per setting. LOO is the original fixed-source EPJ predictor; SA is ordinary self-attention. All six sampling settings are shown. Counts D_5 are exact and E_5 is rounded to one decimal.

NFE	Temp.	Model	V_5 (%) ↑	V_6 (%) ↑	D_5 ↑	E_5 ↑	JSD_4 ↓
128	1.0	LOO	44.35	34.49	2135	1101.4	0.08215
128	1.0	SA	54.08	46.45	2266	1126.0	0.07284
128	0.7	LOO	64.67	52.91	2121	707.6	0.08250
128	0.7	SA	73.41	65.40	2162	726.5	0.07988
512	1.0	LOO	47.57	37.03	2145	1031.1	0.07773
512	1.0	SA	58.17	50.40	2367	1078.1	0.06881
512	0.7	LOO	67.82	56.48	2155	706.5	0.08246
512	0.7	SA	74.61	66.62	2140	704.0	0.08188
2048	1.0	LOO	48.72	39.05	2252	1137.3	0.07654
2048	1.0	SA	58.64	50.74	2280	1071.3	0.06884
2048	0.7	LOO	68.24	57.32	2183	743.9	0.08051
2048	0.7	SA	75.81	68.53	2151	740.2	0.08026

H.2 A SEPARATE HEAD FOR PREDICTING $p_{i,t}$

An alternative to analytic likelihood completion is to learn the drift posterior. Let $h_i^{(D)}(x_{-i}, t)$ denote the contextual feature defined by the input-isolated architecture in (9). A positionwise head receives this feature together with the local observation,

$$r_{\theta,i}(\cdot \mid x, t) = \mathrm{softmax}\big(g_\theta(h_i^{(D)}(x_{-i}, t), x_i, t)\big),$$
$$\mathcal{L}_{\mathrm{dual}} = \mathcal{L}_{\mathrm{LOO}} + \mathbb{E}\left[-\sum_i \log r_{\theta,i}(Y_i \mid X_t, t)\right].$$

For a fixed conditional path, including fixed embeddings and corruption schedules, assume that the model class jointly represents both posteriors. Every global minimizer of the expected loss then satisfies $q_{\theta,i,t} = q^\star_{i,t}$ and $r_{\theta,i} = p_{i,t}$ almost everywhere under the training law of (t, X_t^Y), recovering the coefficients of (8) under that law. This statement does not constrain predictions outside the support of the training law, in particular beyond the training-time cutoff. Marginal preservation over a given time interval still requires exact posterior coefficients on that interval and the existence and uniqueness assumptions of Proposition B.1. The local input and the drift-head activations never enter the jump branch. Both losses may update the shared contextual parameters.

H.3 CADD-DERIVED CORRUPTION

We adapt the masked categorical supervision and Gaussian interpolation of CADD (Zheng et al., 2026, Section 4.3 and Appendix B), with detached hints and 10% whole-hint dropout from its released training guide. Our architecture and normalized embeddings are adaptations; the control is not a reproduction of the original CADD backbone. Using clean-progress time t (the paper's noise time is $1 - t$), draw $t \sim \mathrm{Unif}[0, .999)$ per sequence and independent $R_{i,t} \sim \mathrm{Bernoulli}(t)$, where $R_{i,t} = 1$ means resolved. For independent $\varepsilon_i \sim \mathcal{N}(0, I_{64})$, set

$$X_{i,t} = R_{i,t} e(Y_i) + (1 - R_{i,t})\,\mathrm{sg}\big[H\big(te(Y_i) + (1-t)\varepsilon_i\big)\big], \qquad H \sim \mathrm{Bernoulli}(0.9).$$

Here sg stops gradients and one H is shared by the entire minibatch. Resolved inputs still supply embedding gradients. Before hint dropout, the unresolved conditional covariance is $(1-t)^2 I_{64}$, with a Gaussian initial state.

H.4 CADD-DERIVED TRAINING WITH ONE SHARED PREDICTOR

This comparison uses a matched pair trained for 60,000 updates with seed 0. One transformer stream and one categorical head supply the predictions for both continuous and discrete updates. Control and LOO share initialization, data order, parameter count, corruption, optimizer, and training budget;

only the attention diagonal differs. Both use fixed position-local sources for keys and values at every layer and queries initialized independently of the inputs. LOO excludes the position's own input and status from both predictions, including indirect routes within one network call. Resolved positions receive no loss and remain fixed during sampling; their embeddings remain visible to other positions.

We use the 90/5/5-million-character Text8 split, $|\mathcal{V}| = 27$, $L = 128$, and normalized embeddings in $\mathbb{R}^{64}$ with norm 8. The network has eight blocks of width 256, eight attention heads, SwiGLU width 704, rotary positions, AdaLN-Zero conditioning, and feature/attention dropout 0.1. There are 6,878,880 trainable parameters including the embeddings. Training uses batch size 512, AdamW with constant learning rate 10^{-3}, coefficients $(0.9, 0.999)$, numerical constant 10^{-8}, no warmup or weight decay, gradient clipping at one, and EMA decay 0.9999 for network and embeddings. The corruption is the CADD-derived Gaussian interpolation, linear masking, detached hints, and 10% minibatch hint dropout described in Appendix H.3. The loss is a single categorical cross-entropy term at unresolved positions, divided by BL for a batch of B sequences, using the one shared prediction. Our fixed-source architecture and normalized embeddings are adaptations of the CADD recipe (Zheng et al., 2026); this control does not reproduce the published model.

Continuous estimates and discrete rules. Sampling starts with independent Gaussian hints and no resolved positions. On a uniform grid from zero to .999, one call at time t gives logits $\ell_{i,t}$ for both updates. An unresolved position commits over $[t, r]$ with probability $(r - t)/(1 - t)$, except that the final interval commits all survivors. Uncommitted positions receive a fresh hint with independent $\varepsilon'_i \sim \mathcal{N}(0, I_{64})$,

$$X_{i,r} = r m_{i,t} + (1 - r)\varepsilon'_i, \qquad m_{i,t} = \begin{cases} e(\arg\max_a \ell_{i,t}(a)), & \text{hard,} \\ \sum_a \mathrm{softmax}(\ell_{i,t})_a e(a), & \text{soft.} \end{cases}$$

These are fresh-noise updates rather than Brownian-bridge increments. The endpoint choice is independent of the discrete rule: CAT uses raw probabilities, TEMP uses temperature 0.7, TOPP uses mass 0.9, and MAX uses argmax only for the committed token. In particular, TEMP and TOPP never alter the soft continuous mean. CONF preserves each sequence's independently drawn commitment count and selects that many unresolved positions with largest raw predicted confidence, with exchangeable random tie-breaking; destinations remain categorical. Positions already resolved stay at their committed embeddings. These sampling changes, and the LOO continuous prediction without Bayesian completion, carry no guarantee of preserving the prescribed marginals.

One NFE counts one transformer call, including the last commitment call; there is no additional terminal call. All 60 combinations of two predictors, two continuous estimates, five discrete rules, and 32/128/512 NFE are complete for training seed 0. Each generates 1,024 length-128 sequences with sampling seed 0, batch size 64, EMA network and embeddings, FP32 and math attention on a V100. The dictionary, validation reference, and metric implementation match Appendix E. Different decoding cells reuse checkpoints and do not supply independent training seeds. CAT at 128 NFE is the primary comparison; the sweep is reported in full below, rather than selecting the best observed setting.

Results. LOO improves V_5 in all 30 matched comparisons, V_6 in 29, D_5 in 21, E_5 in eight, and JSD_4 in nine. At 128 NFE with CAT, soft LOO improves all five metrics over its control. Hard LOO increases V_5 and D_5 but reduces E_5 from 417.2 to 379.5 and worsens JSD_4 from 0.10837 to 0.12347. For LOO, soft endpoints have larger E_5 and lower divergence than hard endpoints for every tested rule and budget. TEMP and TOPP increase soft LOO validity over CAT at 128 NFE, but reduce its diversity and worsen its divergence. Increasing NFE from 128 to 512 raises V_5, reduces E_5, and worsens divergence for all ten LOO settings. For soft LOO with CAT, V_5 increases from 67.55% to 71.27%, while E_5 falls from 761.8 to 542.3 and divergence rises from 0.08610 to 0.09606. These are paired observations for one training seed.

Table 14: CADD-derived shared-predictor decoding at 128 NFE: 60,000-update EMA, training and sampling seed 0, and 1,024 sequences per cell. Hard and soft specify the continuous endpoint; the discrete rule is varied independently. Bold marks the better value in each matched control/LOO pair. No training-seed standard deviation is available.

Rule	Predictor	V_5 (%) ↑	V_6 (%) ↑	D_5 ↑	E_5 ↑	JSD_4 ↓
Hard continuous endpoint						
CAT	Control	45.95	33.59	1316	**417.2**	**0.10837**
CAT	LOO	**70.62**	**58.63**	**1865**	379.5	0.12347
TEMP	Control	53.01	38.82	1346	**336.8**	**0.11475**
TEMP	LOO	**79.55**	**66.52**	**1541**	189.0	0.17239
TOPP	Control	49.58	37.71	1380	**436.4**	**0.10492**
TOPP	LOO	**75.35**	**64.57**	**1802**	334.6	0.13314
MAX	Control	73.68	52.33	**927**	**88.6**	**0.20184**
MAX	LOO	**97.40**	**85.93**	308	7.7	0.42072
CONF	Control	42.96	27.54	**978**	**188.5**	**0.14950**
CONF	LOO	**70.64**	**35.86**	588	20.9	0.32908
Soft continuous endpoint						
CAT	Control	43.93	33.07	1496	615.6	0.10799
CAT	LOO	**67.55**	**58.79**	**2291**	**761.8**	**0.08610**
TEMP	Control	51.01	38.98	1547	**528.9**	**0.10414**
TEMP	LOO	**77.58**	**68.17**	**1986**	462.6	0.10927
TOPP	Control	47.23	36.01	1544	616.4	0.10132
TOPP	LOO	**72.78**	**64.38**	**2244**	**693.1**	**0.09075**
MAX	Control	69.01	52.52	**1219**	**225.6**	**0.15387**
MAX	LOO	**95.64**	**88.81**	712	27.7	0.31796
CONF	Control	41.16	29.95	**1189**	**377.7**	**0.13903**
CONF	LOO	**74.02**	**56.04**	1024	86.0	0.22006

Table 15: CADD-derived shared-predictor decoding at 32 NFE: 60,000-update EMA, training and sampling seed 0, and 1,024 sequences per cell. Hard and soft specify the continuous endpoint; the discrete rule is varied independently. Bold marks the better value in each matched control/LOO pair. No training-seed standard deviation is available.

Rule	Predictor	V_5 (%) ↑	V_6 (%) ↑	D_5 ↑	E_5 ↑	JSD_4 ↓
Hard continuous endpoint						
CAT	Control	30.92	19.51	1089	366.2	0.11044
CAT	LOO	**56.29**	**43.03**	**1849**	**467.4**	**0.10689**
TEMP	Control	37.57	23.75	1105	**294.8**	**0.11213**
TEMP	LOO	**67.52**	**51.60**	**1661**	252.7	0.13686
TOPP	Control	33.65	22.15	1156	384.3	**0.10560**
TOPP	LOO	**61.58**	**48.24**	**1858**	**420.4**	0.10889
MAX	Control	55.28	29.88	**835**	**93.2**	**0.17755**
MAX	LOO	**92.16**	**70.29**	466	12.1	0.36684
CONF	Control	27.48	15.38	945	**245.6**	**0.13181**
CONF	LOO	**53.93**	**32.06**	**1016**	84.6	0.21833
Soft continuous endpoint						
CAT	Control	27.55	17.80	1166	506.9	0.12304
CAT	LOO	**53.05**	**43.00**	**2224**	**943.4**	**0.08325**
TEMP	Control	34.09	22.90	1245	473.2	0.11310
TEMP	LOO	**65.13**	**54.62**	**2245**	**726.1**	**0.09050**
TOPP	Control	31.10	21.06	1270	572.1	0.11238
TOPP	LOO	**59.20**	**49.50**	**2331**	**958.3**	**0.07979**
MAX	Control	46.92	29.35	1062	**233.0**	**0.14755**
MAX	LOO	**88.46**	**75.38**	**1083**	61.3	0.25569
CONF	Control	23.23	14.32	996	**421.8**	0.15064
CONF	LOO	**58.38**	**45.14**	**1499**	275.5	**0.14467**

Table 16: CADD-derived shared-predictor decoding at 512 NFE: 60,000-update EMA, training and sampling seed 0, and 1,024 sequences per cell. Hard and soft specify the continuous endpoint; the discrete rule is varied independently. Bold marks the better value in each matched control/LOO pair. No training-seed standard deviation is available.

Rule	Predictor	V_5 (%) ↑	V_6 (%) ↑	D_5 ↑	E_5 ↑	JSD_4 ↓
Hard continuous endpoint						
CAT	Control	54.24	42.33	1450	**425.9**	**0.11701**
CAT	LOO	**74.56**	**62.65**	**1701**	286.1	0.13670
TEMP	Control	61.98	47.86	1343	**304.6**	**0.12622**
TEMP	LOO	**83.25**	**72.33**	**1400**	158.3	0.18997
TOPP	Control	58.74	46.96	1478	**430.7**	**0.11372**
TOPP	LOO	**80.04**	**69.98**	**1675**	271.1	0.15121
MAX	Control	80.61	62.54	**898**	**77.2**	**0.22200**
MAX	LOO	**98.05**	**90.40**	263	6.9	0.43330
CONF	Control	55.90	**38.45**	**903**	**140.4**	**0.18721**
CONF	LOO	**72.81**	34.28	439	15.8	0.37798
Soft continuous endpoint						
CAT	Control	53.55	43.24	1513	539.4	0.11474
CAT	LOO	**71.27**	**61.85**	**2049**	**542.3**	**0.09606**
TEMP	Control	60.19	48.52	1536	**460.0**	**0.11230**
TEMP	LOO	**80.62**	**71.82**	**1792**	340.2	0.12702
TOPP	Control	57.41	47.12	1569	**554.7**	0.10915
TOPP	LOO	**76.90**	**68.48**	**2019**	507.2	**0.10370**
MAX	Control	77.99	64.13	**1099**	**185.5**	**0.17410**
MAX	LOO	**96.86**	**91.58**	533	18.2	0.34971
CONF	Control	55.82	42.74	**1043**	**244.8**	**0.16683**
CONF	LOO	**76.37**	**53.38**	749	39.8	0.27811

Word concentration. MAX and CONF concentrate the word distribution. In every matched setting, CONF has lower E_5 and higher divergence than CAT. Table 17 reports the share of valid word occurrences accounted for by the three and ten most frequent words. The three most frequent are `three`, `seven`, and `eight`, in varying order. With hard LOO and MAX, they account for 86.92% of valid word occurrences of length at least five, despite $V_5 = 97.40\%$. These are lexical concentration measurements, not fractions of identical generated sequences.

Table 17: Word concentration for shared-predictor LOO at 128 NFE. Percentages use dictionary-valid occurrences of words of at least five letters as the denominator. The checkpoints and samples are those in Table 14.

Endpoint	Rule	Top-three share (%)	Top-ten share (%)	E_5
Hard	CAT	28.24	32.57	379.5
Hard	TEMP	39.16	43.31	189.0
Hard	TOPP	29.65	34.41	334.6
Hard	MAX	86.92	89.23	7.7
Hard	CONF	72.61	75.27	20.9
Soft	CAT	16.58	21.37	761.8
Soft	TEMP	23.84	29.06	462.6
Soft	TOPP	17.83	22.41	693.1
Soft	MAX	67.55	71.79	27.7
Soft	CONF	49.49	54.29	86.0

I ZINC250K EXPERIMENTAL DETAILS

I.1 DATA, REPRESENTATION, AND PREDICTOR

We use the GDSS benchmark CSV and test indices at commit `24cc490e0c5b` (Jo et al., 2022). From its 224,568 training examples, we reserve 5,000 for development using split seed 314159 and grouping identical non-isomeric canonical SMILES, leaving 219,568 fitting examples. Together with the 24,887 official test examples, these comprise 249,455 molecules. The input molecules are connected. Fitting and development identities do not overlap. The official test split is unchanged; 295 test entries share fitting identities after non-isomeric canonicalization. This inherited overlap is retained and recorded. Test examples supply neither training batches nor the atom-count distribution. Holding out development examples reduces the fitting set relative to methods using the entire published training pool.

Graphs have at most 38 heavy atoms and are represented by $L = 38 + \binom{38}{2} = 741$ coordinates. Atom counts are sampled from the fitting-set distribution. The nine atom types are $\{\mathrm{C}, \mathrm{N}, \mathrm{O}, \mathrm{F}, \mathrm{P}, \mathrm{S}, \mathrm{Cl}, \mathrm{Br}, \mathrm{I}\}$; bond labels are no bond, single, double, and triple. Kekulization encodes aromatic bonds through single and double bonds. Hydrogens are implicit; the model predicts neither stereochemistry nor three-dimensional coordinates nor separate formal-charge labels. Decoding follows the GDSS positive-charge convention where applicable. Given the sampled atom count n, the first n atom slots and all $\binom{n}{2}$ corresponding pair slots are active, including pairs labeled "no bond." Padding is an external mask, not an atom category; inactive slots are excluded from the loss and attention sources.

The atom and bond embeddings are learned vectors in $\mathbb{R}^{64}$, each normalized to norm 8.

The predictor has eight blocks, width 256, eight attention heads, 32 time features, 32 context channels, dropout 0.1, AdaLN conditioning, and SwiGLU layers. The atom and bond classifiers share this backbone; they are not separate drift and jump heads. Its 6,923,552 parameters and the $(9 + 4) \times 64 = 832$ embedding parameters total 6,924,384 for each of the five trained models. Parameters have matched shapes and initializations within each seed.

An atom query aggregates features of other atoms and their incident bonds; a bond query uses symmetric endpoint features and pairs of bonds through a third atom. Each such context excludes the target coordinate's state and absorption indicator. All features use noisy observations, without access to the clean adjacency. Attention keys and values are fixed functions of individual input coordinates, as in (9); updated queries never serve as sources. LOO excludes the target source in every layer. Allowing this source is the only architectural change for predictors of $p_{i,t}(\cdot \mid X_t)$. Shared features and the absence of positional embeddings give equivariance under joint permutations of atom slots and their induced pair slots.

I.2 TRAINING AND SAMPLING

With $u = t^2$, a continuous coordinate before absorption has conditional law $\mathcal{N}(ue(y_i), u(1-u)I_{64})$ and starts at zero. Jump coordinates have first-jump CDF t^2 and hazard $2t/(1-t^2)$; after jumping they remain at the selected anchor. An observed indicator records absorption. In EPJ, the coordinate remains zero before its jump. For unabsorbed continuous coordinates, Bayesian completion uses the Gaussian likelihood above. PD-BC completes the LOO prediction for the drift, and AJD-BC-D does so only for the drift, retaining LOO jump probabilities. EPJ uses $p_{\theta,i,t}$. Table 18 maps the seven sampling variants to the five trained models. PD-LOO and PD-BC share one checkpoint, as do AJD-LOO and AJD-BC-D; completion is a sampling-time operation. The absorbing process differs from RJD: continuous motion stops after a jump, and Proposition 3.2 is not a marginal-preservation result for this augmented absorbing process.

Training samples t uniformly from $[0, 0.999)$ and minimizes $\frac{1}{2}\mathcal{L}_{\mathrm{atom}} + \frac{1}{2}\mathcal{L}_{\mathrm{bond}}$, where each cross-entropy is normalized over its eligible coordinates. These are all active coordinates for PD and only unabsorbed active coordinates for jump paths. No additional bond-class weights are used. AdamW uses batch size 512, accumulated in microbatches of 128, learning rate 10^{-3} after 1,000 warmup updates, cosine decay to 10^{-4} at 40,000 updates, zero weight decay, and gradient-norm clipping at one. EMA includes predictor and anchors, with decay $\min(0.999, (1+s)/(10+s))$ at update s. The

Table 18: ZINC250k methods. All predictors have the same architecture and parameter count. Unrestricted and LOO prediction differ only in allowing or excluding the target input as an attention source. BC denotes analytic likelihood completion; all jump destinations use CAT in the test comparison.

Method	Conditional path	Predictor	Drift	Jump probabilities
EPJ	Pure jump	Unrestricted	–	Unrestricted
PD	Diffusion	Unrestricted	Unrestricted	–
PD-LOO	Diffusion	LOO	Raw LOO	–
PD-BC	Diffusion	LOO	BC	–
AJD	Absorbing jump diffusion	Unrestricted	Unrestricted	Unrestricted
AJD-LOO	Absorbing jump diffusion	LOO	Raw LOO	Raw LOO
AJD-BC-D	Absorbing jump diffusion	LOO	BC	Raw LOO

test comparison uses the EMA weights at exactly 40,000 updates, rather than selecting checkpoints by validation loss. Computation uses FP32 with TF32 disabled.

For M NFE, the sampling grid is $t_j = 1 - \cos(\pi j/(2M))$, $j = 0, \ldots, M$, with $u_j = t_j^2$. There is one predictor call at each left endpoint. Given its predicted anchor mean m, the continuous update is

$$x' = m + \alpha(x - m) + \sqrt{(u_{j+1} - u_j)\alpha}\,\varepsilon, \qquad \alpha = \frac{1 - u_{j+1}}{1 - u_j}, \qquad \varepsilon \sim \mathcal{N}(0, I_{64}).$$

An eligible coordinate jumps with probability $(u_{j+1} - u_j)/(1 - u_j)$. Thus the final interval reaches $u = 1$ without a network call at $t = 1$, and M NFE counts exactly M backbone calls. Pure diffusion is decoded by nearest anchor, equivalently maximum anchor inner product. Test jump destinations use CAT at temperature one, without correction steps. Except for the evaluation specified below, atom counts and generation seeds are paired across methods, although different samplers consume random numbers differently. Evaluation uses batches of 32 graphs and sampling seed base 1729. For AJD-LOO at 512 NFE with training seed 1, we use an independent H100 evaluation with batch size 256 and sampling seed base 271828, keeping the same 40,000-update EMA checkpoint, strict LOO predictor, and CAT sampler without BC. This evaluation has the same sample count and test reference but does not share the paired atom-count and sampling-seed realization. The original five test variants run on H100 for both seeds. Additional raw-LOO drift evaluations use V100 for seed 0 and H100 for seed 1 at 64 and 128 NFE; some 512-NFE evaluations resume on A100. Table 20 reports generation time and hardware separately for each seed, excluding scoring and queue waits. These times are not a comparison of sampler speed on a common GPU and batch size.

I.3 Molecular evaluation metrics

Validity (no correction) is the number of outputs passing full RDKit sanitization before repair divided by the number generated. Valid & connected additionally requires exactly one connected component, with the same denominator; it is not connectivity conditional on validity. For comparison with GDSS (Jo et al., 2022), we separately report the fraction passing its valence check before bond correction, labeled GDSS valence (no correction). This uses the GDSS `check_valency` routine rather than full sanitization; the two checks need not agree, and neither requires a connected graph. Corrected outputs use GDSS bond downgrading followed by largest-component selection; we compute their metrics separately from raw outputs.

FCD compares the empirical means and covariances of pretrained ChemNet features (Preuer et al., 2018), using `fcd-torch` 1.0.7. For NSPDK, we use normalized GDSS/EDeN graph features at complexity four (Costa & De Grave, 2010; Jo et al., 2022). If $\phi(G)$ denotes these features, the reported statistic is

$$\widehat{\mathrm{MMD}}^2 = \left\| \frac{1}{N} \sum_{r=1}^{N} \phi(G_r) - \frac{1}{N_{\mathrm{ref}}} \sum_{r=1}^{N_{\mathrm{ref}}} \phi(G_r^{\mathrm{ref}}) \right\|_2^2,$$

the biased empirical squared MMD for the corresponding linear kernel. Here N counts valid raw outputs or valid corrected outputs, respectively; invalid outputs do not enter either distributional metric. Their frequency is therefore reported separately by validity. The FCD reference retains the benchmark SMILES, including stereochemical information where present.

Uniqueness is the number of distinct valid non-isomeric canonical SMILES divided by the number of valid outputs. Novelty is the fraction of distinct valid identities absent from the training reference, reported against both the fitting set and the full published training pool. The fitting-copy fraction instead counts repetitions and divides the number of valid samples matching fitting identities by the valid sample count. Raw identities retain all components; corrected identities retain only the selected component. Component pruning is the fraction of generated graphs for which decoding discards components after bond repair.

The test evaluation uses 10,000 attempted graphs per method, NFE budget, and training seed, with all 24,887 test molecules as reference. Invalid outputs are not replaced to reach a target number of valid molecules. The tables report training seeds 0 and 1. Each metric is computed separately per seed and summarized by its mean and sample standard deviation; generated collections are not pooled for FCD. The development sweep instead uses seed 0, 1,000 graphs, and all 5,000 development references. Absolute FCD values across these protocols are not a matched measure of training progress. Evaluation uses RDKit 2025.03.6, NumPy 2.2.6, NetworkX 3.4.2, and a fixed Python hash seed of zero for the NSPDK feature map.

I.4 Test results at 40,000 updates

Table 19 extends the main comparison to 64 and 512 NFE and includes corrected NSPDK. At 128 NFE, raw LOO drift gives AJD the lowest mean FCD, while completion gives it the lowest mean NSPDK; these metrics therefore rank the drift choices differently. Increasing NFE improves EPJ and AJD-BC-D raw validity and connected validity, whereas PD and PD-BC change less. At 512 NFE, AJD-LOO has the highest mean connected validity and lowest mean raw FCD, while AJD-BC-D has the lowest mean corrected FCD and NSPDK. These rankings describe means over two seeds.

Secondary metrics and generation time. All 42 test evaluations have 100% corrected validity and no decoding errors. Novelty against both training references is 100%, with one exception: PD-LOO at 128 NFE for seed 1 contains one fitting-set copy, giving raw and corrected novelty of 99.9825% and 99.9867%, respectively. Its raw and corrected fitting-copy fractions are $1/5921$ and $1/10000$; all other evaluations have zero observed fitting copies. Raw uniqueness is 100% except for PD-LOO; corrected uniqueness is also 100% except for PD-LOO and AJD at 128 NFE for seed 0 and 512 NFE for seed 1 (99.99% for each of these two AJD evaluations). Table 20 gives the nonconstant secondary metrics. The low uniqueness of PD-LOO shows an additional limitation of its raw LOO drift. High novelty or corrected validity alone does not establish good raw validity, connectivity, or distributional agreement.

Table 19: ZINC250k test results: 40,000-update EMA weights, training seeds 0 and 1, 10,000 attempted graphs per seed, and all 24,887 test references. Each cell shows the mean above ± sample standard deviation. All jumps use CAT and $\kappa(t) = t^2$. The validity percentages use all generated graphs as denominator. NSPDK is squared MMD. Corrected metrics use bond repair and largest-component selection. The seed-1 AJD-LOO evaluation at 512 NFE uses the independent sampling realization described above.

		Raw				Corrected	
Method	NFE	Valid (%) ↑	Valid & conn. (%) ↑	FCD ↓	NSPDK ↓	FCD ↓	NSPDK ↓
EPJ	64	59.88 ±0.45	34.25 ±0.62	21.375 ±0.159	0.04826 ±0.00035	21.534 ±0.028	0.05192 ±0.00037
PD	64	88.98 ±0.52	45.36 ±0.01	22.178 ±0.022	0.05229 ±0.00021	21.933 ±0.015	0.05197 ±0.00014
PD-LOO	64	52.56 ±5.35	12.03 ±3.87	28.285 ±2.367	0.27730 ±0.07328	25.727 ±2.735	0.18862 ±0.04773
PD-BC	64	91.33 ±0.05	50.61 ±2.76	21.694 ±0.025	0.05189 ±0.00098	21.544 ±0.069	0.05159 ±0.00091
AJD	64	86.54 ±1.03	43.80 ±0.97	22.309 ±0.029	0.05135 ±0.00066	22.051 ±0.047	0.05129 ±0.00053
AJD-LOO	64	72.11 ±0.95	61.27 ±1.32	20.827 ±0.078	0.05063 ±0.00081	21.506 ±0.113	0.05254 ±0.00039
AJD-BC-D	64	65.82 ±1.27	48.27 ±1.38	20.786 ±0.110	0.04420 ±0.00079	21.304 ±0.086	0.04657 ±0.00038
EPJ	128	69.02 ±0.22	41.81 ±0.11	21.190 ±0.039	0.04882 ±0.00088	21.305 ±0.002	0.05142 ±0.00048
PD	128	89.98 ±0.33	49.97 ±0.37	21.919 ±0.031	0.05154 ±0.00015	21.740 ±0.003	0.05127 ±0.00009
PD-LOO	128	55.38 ±5.42	20.07 ±3.91	24.966 ±1.164	0.15202 ±0.02161	21.670 ±1.250	0.10993 ±0.00862
PD-BC	128	91.84 ±0.40	54.19 ±2.21	21.642 ±0.066	0.05149 ±0.00088	21.535 ±0.090	0.05134 ±0.00057
AJD	128	87.41 ±0.95	47.82 ±2.49	22.065 ±0.088	0.05054 ±0.00043	21.943 ±0.082	0.05062 ±0.00011
AJD-LOO	128	83.96 ±0.52	71.07 ±0.61	20.460 ±0.019	0.05141 ±0.00011	20.968 ±0.015	0.05242 ±0.00019
AJD-BC-D	128	78.53 ±0.16	59.11 ±0.38	20.677 ±0.090	0.04629 ±0.00003	20.990 ±0.053	0.04715 ±0.00024
EPJ	512	77.12 ±0.15	49.34 ±0.25	20.921 ±0.041	0.04978 ±0.00014	21.119 ±0.056	0.05157 ±0.00013
PD	512	89.94 ±0.35	51.38 ±0.73	21.693 ±0.097	0.05081 ±0.00059	21.636 ±0.093	0.05079 ±0.00052
PD-LOO	512	59.42 ±5.88	24.91 ±4.55	24.185 ±1.237	0.13591 ±0.01791	21.281 ±1.161	0.10386 ±0.00698
PD-BC	512	92.55 ±0.08	55.77 ±2.77	21.264 ±0.117	0.05061 ±0.00046	21.235 ±0.176	0.05073 ±0.00028
AJD	512	87.95 ±0.95	49.94 ±2.74	21.929 ±0.001	0.04994 ±0.00054	21.858 ±0.154	0.05013 ±0.00015
AJD-LOO	512	91.08 ±0.06	76.21 ±0.25	20.575 ±0.150	0.05357 ±0.00006	20.908 ±0.133	0.05375 ±0.00017
AJD-BC-D	512	86.84 ±0.28	65.91 ±0.80	20.615 ±0.045	0.04663 ±0.00030	20.843 ±0.056	0.04697 ±0.00035

Table 20: Secondary ZINC250k metrics and generation time for the same two-seed test samples as Table 19. GDSS valence is checked before correction. Uniqueness uses valid outputs as denominator; pruning uses all attempted graphs. All four metric columns are percentages, with the mean above ± sample standard deviation. Each time column shows generation minutes for 10,000 graphs above the GPU used, separately for each seed; scoring and queue waits are excluded. A slash indicates generation resumed on different hardware. Batch size is 32 except for seed-1 AJD-LOO at 512 NFE (256).

Method	NFE	GDSS valence	Raw unique	Corrected unique	Component pruned	Seed 0 min. / GPU	Seed 1 min. / GPU
EPJ	64	59.86 ±0.45	100.00 ±0.00	100.00 ±0.00	41.56 ±0.59	15.5 H100	15.6 H100
PD	64	88.95 ±0.53	100.00 ±0.00	100.00 ±0.00	48.44 ±0.16	15.4 H100	15.6 H100
PD-LOO	64	52.56 ±5.35	67.85 ±10.75	49.96 ±9.07	79.54 ±5.61	54.1 V100	15.4 H100
PD-BC	64	91.26 ±0.10	100.00 ±0.00	100.00 ±0.00	44.30 ±3.05	15.6 H100	15.5 H100
AJD	64	86.49 ±1.02	100.00 ±0.00	100.00 ±0.00	48.73 ±1.73	15.5 H100	15.5 H100
AJD-LOO	64	72.11 ±0.95	100.00 ±0.00	100.00 ±0.00	14.91 ±0.74	54.1 V100	15.5 H100
AJD-BC-D	64	65.81 ±1.29	100.00 ±0.00	100.00 ±0.00	26.46 ±0.70	15.6 H100	15.7 H100
EPJ	128	69.00 ±0.19	100.00 ±0.00	100.00 ±0.00	38.32 ±0.07	30.9 H100	30.8 H100
PD	128	89.94 ±0.34	100.00 ±0.00	100.00 ±0.00	44.15 ±0.06	30.9 H100	31.0 H100
PD-LOO	128	55.37 ±5.43	92.78 ±5.10	71.76 ±5.01	67.68 ±4.50	108.2 V100	31.0 H100
PD-BC	128	91.81 ±0.39	100.00 ±0.00	100.00 ±0.00	40.61 ±2.84	31.1 H100	31.1 H100
AJD	128	87.36 ±0.96	100.00 ±0.00	100.00 ±0.01	44.72 ±3.34	31.0 H100	30.9 H100
AJD-LOO	128	83.95 ±0.52	100.00 ±0.00	100.00 ±0.00	15.32 ±0.50	108.4 V100	32.8 H100
AJD-BC-D	128	78.50 ±0.15	100.00 ±0.00	100.00 ±0.00	24.75 ±0.62	31.1 H100	31.1 H100
EPJ	512	77.11 ±0.15	100.00 ±0.00	100.00 ±0.00	35.61 ±0.16	123.7 H100	125.7 H100
PD	512	89.88 ±0.33	100.00 ±0.00	100.00 ±0.00	42.37 ±0.70	123.7 H100	123.6 H100
PD-LOO	512	59.41 ±5.87	95.54 ±4.17	76.41 ±4.17	63.73 ±4.50	258.8 A100/V100	138.3 A100/H100
PD-BC	512	92.53 ±0.11	100.00 ±0.00	100.00 ±0.00	39.38 ±2.97	124.4 H100	124.3 H100
AJD	512	87.91 ±0.93	100.00 ±0.00	100.00 ±0.01	42.72 ±3.46	123.8 H100	123.5 H100
AJD-LOO	512	91.08 ±0.06	100.00 ±0.00	100.00 ±0.00	16.11 ±0.09	258.8 A100/V100	114.4 H100
AJD-BC-D	512	86.83 ±0.28	100.00 ±0.00	100.00 ±0.00	24.16 ±0.37	124.4 H100	124.2 H100

I.5 Development sampling ablations at 20,000 updates

We separately compare the five sampling rules from Section 4.1 on fixed 20,000-update EMA checkpoints: CAT, MAX, TEMP with temperature 0.7, TOPP with mass 0.9, and CONF. For CONF, event counts are preserved separately for atom and bond coordinates, then assigned to the most confident eligible coordinates within each group. The drift remains at temperature one and retains the prediction choice in Table 18; only jump sampling is varied. These rules change the sampling dynamics and are not covered by the repeated-reset marginal-preservation guarantee.

Tables 21 and 22 report the complete seed-0 sweep with 1,000 generated graphs per setting and 5,000 development references. MAX markedly increases EPJ validity but worsens FCD and NSPDK. For AJD-LOO at 128 NFE, TEMP improves validity, connected validity, and FCD over CAT, while worsening NSPDK; TOPP improves all six reported metrics over CAT. Thus a sampling rule that improves validity need not improve distributional agreement, and these development results do not substitute for the 40,000-update test comparison.

Table 21: ZINC250k development sampling sweep at 64 NFE: 20,000-update EMA weights, seed 0, 1,000 attempted graphs, and 5,000 development references. Validity percentages use all generated graphs. NSPDK is squared MMD. These scores use a different checkpoint, sample count, and reference from the test comparison.

		Raw				Corrected	
Method	Rule	Valid (%) ↑	Valid & conn. (%) ↑	FCD ↓	NSPDK ↓	FCD ↓	NSPDK ↓
EPJ	CAT	57.8	31.7	23.125	0.05012	22.489	0.05342
EPJ	MAX	97.7	21.9	34.198	0.11466	36.581	0.08680
EPJ	TEMP	70.7	18.2	26.424	0.05897	23.525	0.05374
EPJ	TOPP	83.6	32.6	24.914	0.04133	24.032	0.03467
EPJ	CONF	14.6	9.9	27.652	0.05537	25.745	0.07739
AJD	CAT	76.6	34.2	23.596	0.05172	22.812	0.05234
AJD	MAX	76.5	32.8	23.763	0.05328	23.073	0.05375
AJD	TEMP	77.4	32.7	23.867	0.05283	23.060	0.05310
AJD	TOPP	77.2	33.0	23.619	0.05191	22.834	0.05259
AJD	CONF	77.7	36.1	23.400	0.05360	22.905	0.05366
AJD-LOO	CAT	70.9	61.6	22.173	0.05323	22.594	0.05507
AJD-LOO	MAX	80.9	63.3	23.565	0.09184	23.504	0.08439
AJD-LOO	TEMP	76.1	63.3	21.908	0.05365	22.354	0.05533
AJD-LOO	TOPP	73.8	62.1	21.715	0.04921	21.981	0.05044
AJD-LOO	CONF	56.0	32.6	22.605	0.05338	21.350	0.04609
AJD-BC-D	CAT	63.8	49.7	21.471	0.04470	22.066	0.04739
AJD-BC-D	MAX	71.1	55.6	23.430	0.06766	23.425	0.06458
AJD-BC-D	TEMP	67.8	53.6	22.092	0.05049	22.111	0.05108
AJD-BC-D	TOPP	68.1	53.0	21.242	0.04635	21.664	0.04833
AJD-BC-D	CONF	34.2	16.7	23.743	0.04131	21.374	0.05049

Table 22: ZINC250k development sampling sweep at 128 NFE, with the same 20,000-update checkpoints, seed 0, 1,000 attempted graphs, and 5,000 development references as Table 21. Metric definitions and sampling rules are unchanged.

		Raw				Corrected	
Method	Rule	Valid (%) ↑	Valid & conn. (%) ↑	FCD ↓	NSPDK ↓	FCD ↓	NSPDK ↓
EPJ	CAT	66.8	39.3	21.843	0.05129	21.554	0.05298
EPJ	MAX	97.4	21.9	33.926	0.13650	36.180	0.10706
EPJ	TEMP	77.2	21.7	25.524	0.05885	23.017	0.05377
EPJ	TOPP	87.9	37.4	24.598	0.04136	23.504	0.03666
EPJ	CONF	21.7	15.9	26.011	0.05571	25.490	0.07004
AJD	CAT	81.5	39.0	23.132	0.05230	22.468	0.05219
AJD	MAX	80.7	39.3	23.215	0.05250	22.594	0.05203
AJD	TEMP	81.5	39.3	23.055	0.05184	22.450	0.05188
AJD	TOPP	81.1	39.2	23.018	0.05257	22.567	0.05252
AJD	CONF	79.7	43.1	22.964	0.05158	22.342	0.05160
AJD-LOO	CAT	81.9	71.5	22.087	0.05565	22.387	0.05665
AJD-LOO	MAX	95.2	79.2	23.468	0.10235	23.382	0.09912
AJD-LOO	TEMP	87.6	73.8	21.345	0.06001	21.521	0.05928
AJD-LOO	TOPP	89.3	76.2	21.836	0.05439	22.114	0.05415
AJD-LOO	CONF	79.9	50.3	22.068	0.05828	21.329	0.05310
AJD-BC-D	CAT	80.0	63.6	21.680	0.04538	21.854	0.04673
AJD-BC-D	MAX	85.0	66.5	22.769	0.07209	22.817	0.06924
AJD-BC-D	TEMP	82.2	63.5	21.656	0.05076	21.784	0.05112
AJD-BC-D	TOPP	81.3	63.4	21.346	0.04509	21.634	0.04658
AJD-BC-D	CONF	47.3	26.2	22.769	0.03769	20.746	0.04308