CONTEXT-WEIGHTED DISCRETE FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Discrete flow matching provides a flexible framework for generative modeling on discrete structures (Gat et al., 2024). However, the standard factorized training objective exposes the model to targets of varying difficulty, mixing well-conditioned, predictable tokens with ambiguous, high-entropy ones. We empirically demonstrate that the uncertainty over the value of each token is closely related to the density of available context in its neighborhood. Motivated by this observation, we propose a simple modification to the underlying Continuous-Time Markov Chain (CTMC) that incorporates local context information. Our context-weighted sampler improves generation quality with negligible computational overhead, while our scaled cross-entropy loss function reweights the training signal from different tokens and reduces generative perplexity by up to 63% on OpenWebText (Gokaslan & Cohen, 2019). Moreover, our method approaches a strong semi-autoregressive block diffusion baseline (Arriola et al., 2025) in quality while retaining the ability for any-order generation. These results highlight the role of local context as an important factor in discrete generative modeling and show that simple context-aware modifications can significantly improve both sampling and training efficiency.

1 INTRODUCTION

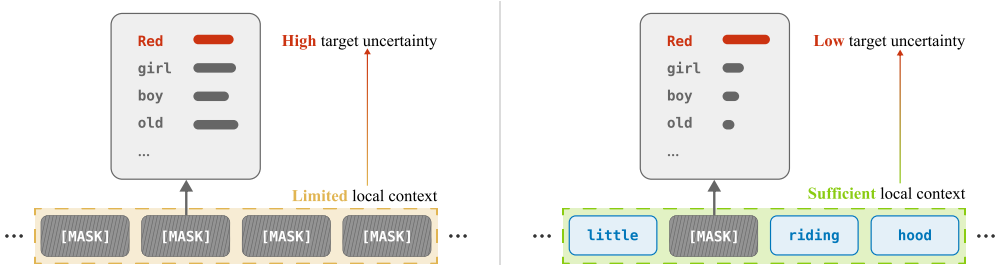

Figure 1: A motivational example. The target word "Red" is much more predictable in the second sentence because of the available local context.

Diffusion and flow matching extend generative modeling beyond autoregressive approaches (Gat et al., 2024; Campbell et al., 2022), with applications to discrete image latents (Gu et al., 2022), molecules (Vignac et al., 2023; Kelvinius et al., 2025), and text (Sahoo et al., 2024). Discrete Flow Matching (DFM) learns continuous-time Markov dynamics along probability paths, enabling parallel and any-order generation (Lipman et al., 2024; Gat et al., 2024).

DFM and discrete diffusion decompose generation into coordinate-wise prediction problems. Unlike autoregressive models conditioned on a fixed prefix, they predict under many partial contexts and token orderings, with widely varying prediction difficulty (Kim et al., 2025). Some tokens are strongly constrained by context, while others remain ambiguous (fig. 1). Existing approaches enforce structured generation orders (Besnier et al., 2025; Arriola et al., 2025; Gat et al., 2025) or adapt sampling at inference time (Ben-Hamu et al., 2025), but do not incorporate uncertainty directly into the generative dynamics.

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.

In this work, we empirically observe that prediction difficulty is closely tied to available local context, measured by the number of unmasked tokens in each target token's neighborhood. Motivated by this, we modify the CTMC update probabilities to depend on local context while preserving the distribution of completed source-to-target transitions, with coordinate marginals recovered under suitable symmetry. This keeps the same noise and data endpoints, but shifts intermediate updates toward better-conditioned tokens. The resulting formulation provides both an inference-time sampler that requires no fine-tuning, and a scaled cross-entropy objective that reweights the training signal. Context-weighted sampling improves generation quality across domains: on OpenWebText (Gokaslan & Cohen, 2019), it improves MAUVE by up to 23%, while for molecule generation (Ramakrishnan et al., 2014), it increases the number of valid and novel samples by up to $2.8\times$ and $1.9\times$, respectively. Our loss objective reduces generative perplexity on OpenWebText by up to 63%, and approaches a strong block diffusion baseline while retaining any-order generation.

Our contributions are:

- We empirically show that prediction entropy is closely linked to available local context, with denser unmasked neighborhoods leading to lower uncertainty.
- We modify the CTMC transition dynamics to depend on local context while preserving active-transition counts and endpoints; coordinate marginals follow under additional symmetry.
- We derive two practical context-aware mechanisms: a sampler applicable to pre-trained models without fine-tuning, and a scaled cross-entropy objective that reweights the training signal from different tokens, resulting in significant gains in generation quality.

2 Background

2.1 Discrete Flow Matching

We consider finite discrete sequences $x \in \mathcal{D}^N$, where $\mathcal{D}$ is a finite vocabulary, and N is the sequence length. Discrete Flow Matching (DFM)(Gat et al., 2024) typically defines a conditional probability path $p_t(x \mid x_0, x_1)$ between a source noise sample $x_0 \sim p_0$ and a data sample $x_1 \sim p_{\text{data}}$. In this work, we consider two common source distributions: a fully masked source, where all coordinates are initialized to a special mask token absent from the data, and a uniform source, where coordinates are initialized from the vocabulary uniformly at random. For both sources, we say that coordinate i is unmasked whenever $x^i = x_1^i$, and masked otherwise. Here x_1 is the fixed endpoint in the conditional-path analysis; at inference it is replaced by a model-sampled endpoint, as described in section 3.2.

The path satisfies the endpoint conditions

$$p_0(x \mid x_0, x_1) = \delta(x_0, x), \qquad p_1(x \mid x_0, x_1) = \delta(x_1, x).$$

where $\delta(z, x)$ denotes the Kronecker delta, equal to 1 when $x = z$ and 0 otherwise. This path is realized as a Continuous-Time Markov Chain (CTMC). For a small step h, its transition kernel can be written as

$$\mathbb{P}(X_{t+h} = y \mid X_t = x) = \delta(x, y) + h\, u_t(y, x \mid x_0, x_1) + o(h),$$

where $u_t(y, x \mid x_0, x_1)$ is the probability velocity, or transition rate, from state x to state y, and $o(h)$ denotes a remainder term $r(h)$ satisfying $r(h)/h \to 0$ as $h \to 0$. DFM uses factorized velocities, where transitions change one coordinate at a time:

$$u_t(y, x \mid x_0, x_1) = \sum_{i=1}^{N} \delta(y^{\bar{i}}, x^{\bar{i}}) u_t^i(y^i, x \mid x_0, x_1).$$

Here $i \in \{1, \ldots, N\}$ indexes positions and $x^{\bar{i}}$ contains all coordinates except i. Factorized velocities enable coordinate-wise simulation without requiring a factorized path distribution. The Kolmogorov forward equation links the velocity to the path marginals.

A common choice is the convex mixture path (Lipman et al., 2024; Gat et al., 2024), defined independently for each coordinate as

$$p_t(x^i \mid x_0, x_1) = (1 - \kappa_t)\delta(x^i, x_0^i) + \kappa_t \delta(x^i, x_1^i),$$

where $\kappa_t \in [0, 1]$ is a scalar noise scheduler that increases with time. Its corresponding coordinate-wise velocity is

$$u_t^i(y^i, x \mid x_0, x_1) = \frac{\dot{\kappa}_t}{1 - \kappa_t} \left[\delta(y^i, x_1^i) - \delta(y^i, x^i)\right].$$

Each coordinate therefore moves toward its data value at the same global rate, independently of local context. We instead let this rate depend on x_t, retaining coordinate-wise simulation.

The model is trained to predict the data endpoint from a partially noised state $X_t \sim p_t(\cdot \mid x_0, x_1)$ using the factorized Conditional Matching objective (Gat et al., 2024)

$$\mathcal{L}_{\mathrm{CM}}(\theta) = -\mathbb{E}_{t, X_0, X_1, X_t} \sum_{i=1}^{N} \log p_{1|t}^{\theta,i}(X_1^i \mid X_t).$$

Thus training decomposes into coordinate-wise prediction problems, where each token is predicted from the same partially noised context but all coordinates are weighted uniformly.

2.2 Probability Path Design

The choice of probability path determines the family of conditional prediction problems encountered during training and sampling. In masked diffusion, different token orderings induce conditionals of widely varying difficulty, and learning all such factorizations can be substantially harder than following a favorable one (Kim et al., 2025). When the data admit an intrinsic structure, aligned orderings yield low-entropy, well-conditioned predictions, whereas unfavorable ones lead to ambiguous, high-entropy targets. Existing approaches address this by restricting the set of factorizations: for example, Halton schedules impose spatially uniform token selection in MaskGIT, while neighboring autoregressive decoding exploits local spatial dependencies in vision models (Besnier et al., 2025; He et al., 2025). Semi-autoregressive and block-wise decoding strategies similarly constrain the generation process by revealing continuous groups of tokens rather than arbitrary subsets (Arriola et al., 2025; Gat et al., 2025; Song et al., 2025; Nie et al., 2025; Zhu et al., 2025; Bie et al., 2025; 2026; Li et al., 2025). While these methods improve the induced conditionals, they hard-code an ordering, which may be suboptimal for individual samples and reduces the flexibility of order-agnostic generation. Test-time ordering methods instead adapt the generation order using confidence, margin, or entropy-based criteria (Zheng et al., 2024; Kim et al., 2025; Ben-Hamu et al., 2025), but rely on the model's own uncertainty estimates and leave the training objective and probability path unchanged. General path formulations and kinetic-optimal path design offer a more direct way to modify discrete probability paths (Shaul et al., 2025), but primarily target global path geometry rather than connecting locality of context to token-level prediction uncertainty. This leaves open how to incorporate such uncertainty into the probability path in a way that can inform both training and inference.

3 Context-weighted Discrete Flow Matching

3.1 Local Context and Token Uncertainty

We find that **local context provides a simple proxy for prediction difficulty**: the number of unmasked neighbors correlates with both predictive entropy and the true token's negative log-likelihood.

Using a pre-trained OWT DFM (Gokaslan & Cohen, 2019), we sample partially masked states $X_t \sim p_t(\cdot \mid x_0, x_1)$ and measure these quantities for each masked token. Both decrease with the number of unmasked neighbors in a fixed window (fig. 2); measurement details are in appendix B.3.

This variation matters because the standard CM objective weights all coordinate-wise targets uniformly, even though their reducible learning signal may differ. For a fixed partially noised state X_t, with $X_1^i \sim p_{1|t}^i(\cdot \mid X_t)$, the expected loss of the model predictor $p_{1|t}^{\theta,i}$ decomposes as

$$\mathbb{E}_{X_1^i \sim p_{1|t}^i(\cdot|X_t)} \left[-\log p_{1|t}^{\theta,i}(X_1^i \mid X_t)\right] = \underbrace{H\left(p_{1|t}^i(\cdot \mid X_t)\right)}_{\text{irreducible uncertainty}} + \underbrace{\mathrm{KL}\left(p_{1|t}^i(\cdot \mid X_t) \,\|\, p_{1|t}^{\theta,i}(\cdot \mid X_t)\right)}_{\text{model error}}.$$

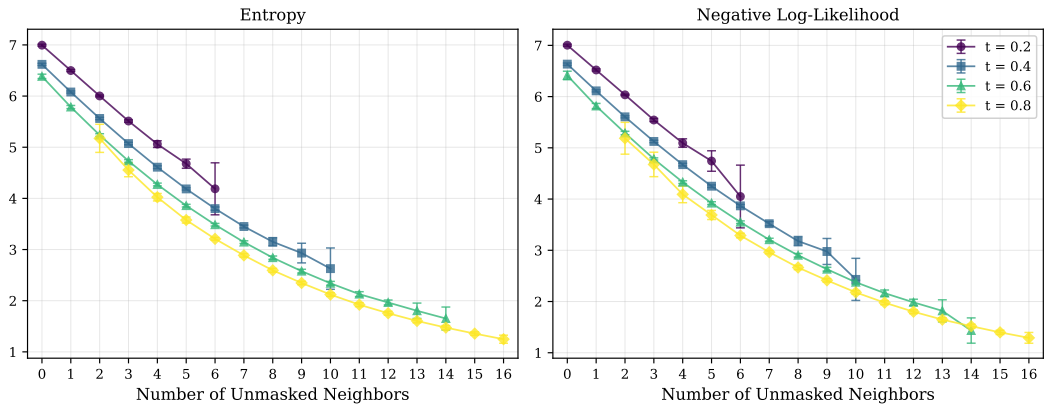

Figure 2: The average entropy (left) and negative log-likelihood (right) of the masked token vs. the number of its unmasked neighbors in the vicinity of size $2r = 16$. Both metrics decrease monotonically as the amount of available local context increases.

The loss combines irreducible uncertainty and reducible model error. A sampled target at a high-entropy coordinate provides less informative supervision than a better-determined target. Ideally, weighting would emphasize reducible error rather than raw loss dominated by ambiguity. Since the true entropy is unavailable, we use local context as an inexpensive proxy for token-level difficulty.

3.2 Context-weighted CTMC

In this section, we propose the context-aware modification to the standard CTMC process (Gat et al., 2024). In order to keep coordinate-wise simulation tractable, we will employ only factorized velocities. For fixed endpoints (x_0, x_1), consider a context weight function $\alpha \colon \mathcal{D}^N \to \mathbb{R}_+^N$. To integrate it into the CTMC, we define a new factorized velocity v_t as:

$$v_t^i(x^i, z \mid x_0, x_1) \coloneqq \alpha^i(z) \cdot u_t^i(x^i, z \mid x_0, x_1)$$

Note that it is a valid probability velocity, since it satisfies the rate conditions:

$$\sum_{x^i \in [d]} v_t^i(x^i, z) = \sum_{x^i \in [d]} \alpha^i(z) u_t^i(x^i, z) = \alpha^i(z) \sum_{x^i \in [d]} u_t^i(x^i, z) = 0$$

and $v_t^i(x^i, z) \geq 0, \forall x^i \neq z^i$. With endpoint dependence suppressed in the notation, α depends on z (the current state) and controls which token positions will be changed, but not the actual value of the token, since it is independent of x. This new velocity $v_t(y, x)$ generates a new conditional probability path $\tilde{p}_t(x \mid x_0, x_1)$, defined as the solution to the corresponding Kolmogorov forward equation (see section 2.1 for details). A necessary requirement for $\tilde{p}_t$ is to connect noise and data distributions, imposing boundary conditions $\tilde{p}_{t=0}(x \mid x_0, x_1) = \delta(x, x_0)$ and $\tilde{p}_{t=1}(x \mid x_0, x_1) = \delta(x, x_1)$. Satisfying them in the general case requires solving the aforementioned equation for $\tilde{p}_t$, which we find intractable. However, for the standard convex mixture path, we are able to derive the conditions on the context weight function α^i to satisfy the boundary conditions and preserve the marginal distribution.

Theorem 1. *Let $\tilde{p}_t(\cdot \mid x_0, x_1)$ denote the context-weighted discrete path associated with the standard convex probability path. Define the active set and its currently masked coordinates by*

$$\mathcal{I} = \{i : x_0^i \neq x_1^i\}, \qquad \mathcal{A}_t = \{i \in \mathcal{I} : x_t^i \neq x_1^i\}.$$

If, at every reachable state,

$$\sum_{i \in \mathcal{A}_t} \alpha^i(x_t) = |\mathcal{A}_t|,$$

then $\tilde{p}_t$ satisfies the endpoint constraints

$$\tilde{p}_0(x \mid x_0, x_1) = \delta(x, x_0), \qquad \tilde{p}_1(x \mid x_0, x_1) = \delta(x, x_1),$$

and induces the same distribution over the number of updated tokens in $\mathcal{I}$ as the standard convex path:

$$\tilde{M}_t = \sum_{i \in \mathcal{I}} \mathbf{1}[\tilde{X}_t^i = x_1^i] \sim \text{Binomial}(|\mathcal{I}|, \kappa_t).$$

If, moreover, the weights depend only on the binary reveal mask and active set,

$$\alpha^i(x) = \bar{\alpha}^i(b(x), \mathcal{I}), \qquad b^j(x) = \mathbf{1}[x^j = x_1^j],$$

and are equivariant under joint circular shifts,

$$\bar{\alpha}^{i+k}(\text{shift}_k(b), \text{shift}_k(\mathcal{I})) = \bar{\alpha}^i(b, \mathcal{I}),$$

then, in the collision-free case $\mathcal{I} = \{1, \dots, N\}$,

$$\mathbb{P}(\tilde{X}_t^i = x_1^i \mid x_0, x_1) = \kappa_t.$$

The first condition requires the weights to average to 1 over the currently masked active coordinates. This preserves the total update rate and the distribution of the number of updated tokens. Coordinates outside $\mathcal{I}$ already equal x_1 and remain unchanged. The symmetry condition gives matching per-coordinate marginals in the stated scope; a fixed collision pattern need not be shift-invariant. The formal proof is given in appendix A.

Inference-time sampling. Our formulation enables a purely inference-time modification of the sampling procedure, without fine-tuning and with negligible computational overhead. The true endpoint x_1 is unavailable at test time: at each step, following the standard Euler sampler, we draw $\widehat{x}_1 \sim p^{\theta}_{1|t}(\cdot \mid x_t)$ and use it in place of x_1. Thus an unmasked token satisfies $x_t^i = \widehat{x}_1^i$, and weights are normalized over $\{i : x_t^i \neq \widehat{x}_1^i\}$. This uses no ground-truth sample or update history.

Specifically, we inject a multiplicative weight $\alpha^i(x)$ into each coordinate of the Euler solver (Lipman et al., 2024; Gat et al., 2024), whose infinitesimal update is

$$\mathbb{P}(X_{t+h}^i = y^i \mid X_t = x, \widehat{x}_1) = \delta(y^i, x^i) + h\alpha^i(x) u_t^i(y^i, x \mid x_0, \widehat{x}_1) + o(h).$$

We apply these weights at each model evaluation, with almost unchanged computational complexity.

We introduce two context-weighted solvers: Neighbor-weighted and Entropy-weighted. Both define position-wise weights $\alpha^i(x_t)$, normalized with a softmax inverse temperature s so that the fixed-endpoint dynamics satisfy the normalization condition of theorem 1. We provide a verification of this in appendix A.3.

For the Neighbor-weighted solver, we use the number of unmasked tokens in a local window of radius r:

$$\alpha^i_{\text{NW}}(x) \propto \sum_{j=i-r}^{i+r} \mathbf{1}[x^j = \widehat{x}_1^j]$$

For the Entropy-weighted solver, we use token-level predictive entropy available at test time:

$$\alpha^i_{\text{EW}}(x) \propto H\left(p^{\theta,i}_{1|t}(\cdot \mid x)\right)$$

In practice, we use $s > 0$ for Neighbor-weighted and $s < 0$ for Entropy-weighted, so locally well-contextualized or low-entropy positions receive larger weights. Unlike Ben-Hamu et al. (2025), our Entropy-weighted sampler is applicable to arbitrary source distributions and uses entropy only to select which positions to update, while preserving the stepwise distribution of the number of updated tokens.

Train-time path $\tilde{p}_t$ sampling. For general context weights $\alpha^i(x)$, the context-weighted path $\tilde{p}_t$ is not available in an analytic form, and therefore, simulation-free sampling is impossible. Moreover, the token values are no longer conditionally independent for $\tilde{p}_t$, so sampling cannot be reduced to a

simple parallel factorized procedure. A straightforward approach would therefore rely on time-wise simulation of the path, which is quite costly. Instead, we leverage the fact that the distribution of the number of updated active tokens M_t is known, which allows us to construct an exact token-wise sampler with at most N token selections (see algorithm 1). For time-independent weights, we first sample $M_t \sim \text{Binomial}(|\mathcal{I}|, \kappa_t)$, and then update the tokens one-by-one. This eliminates the need for costly simulation along the time axis while still returning exact samples from $\tilde{p}_t$. We prove the exactness of the algorithm in appendix A.3.

Algorithm 1: Train-time $\tilde{p}_t(x)$ sampling

Input: N, κ_t, x_0, x_1, δ, α
Output: A sample $\tilde{X}_t \sim \tilde{p}_t$

Set $\mathcal{I} \leftarrow \{i : x_0^i \neq x_1^i\}$;
Sample $m \sim \text{Binomial}(|\mathcal{I}|, \kappa_t)$; `// Number of updated active tokens`
Set $\tilde{X}_t \leftarrow x_0$;
for *step* $= 1$ **to** m **do**
  Sample $i \in \mathcal{I}$ with probability proportional to $(1 - \delta(\tilde{X}_t^i, x_1^i)) \cdot \alpha^i(\tilde{X}_t)$; `// Choose masked token position`
  Set $\tilde{X}_t^i \leftarrow x_1^i$; `// Unmask the token`
return $\tilde{X}_t$;

Scaled Cross-Entropy. Motivated by the empirical link between prediction uncertainty and the available local context (see section 3.1), we introduce *Scaled Cross-Entropy* as a plug-in alternative to the standard Discrete Flow Matching objective:

$$\mathcal{L}_\alpha(\theta) = -\mathbb{E}_{t,X_1,X_0,X_t}\left[\sum_{i=1}^{N} \alpha^i(X_t)\, \log p_{1|t}^{\theta,i}(X_1^i \mid X_t)\right].$$

Here, the context weight function α^i reweights the contribution of each coordinate, upweighting better-conditioned coordinates and downweighting ambiguous ones. It leaves the train-time path sampling unchanged, and therefore retains the efficiency of standard training and any-order sampling at inference time, while capturing the context-weighted importance of different token updates.

4 Experiments

We evaluate our method on two domains: text and molecular generation. For text, we use OpenWebText (OWT) (Gokaslan & Cohen, 2019), a large-scale corpus, while for molecules we use QM9 in SMILES format (Ramakrishnan et al., 2014), a significantly smaller dataset that allows us to study performance in the low-data regime. We follow the MDLM architecture and training setup to ensure fair comparison (Sahoo et al., 2024). Our OWT model has approximately 170M parameters; for QM9, we modify only the vocabulary, resulting in a model with approximately 92M parameters. For text, we report generative perplexity (Gen. PPL), computed with GPT-2(Radford et al., 2019), entropy to measure text diversity, and MAUVE as a balanced quality–diversity metric (Pillutla et al., 2021). For QM9, we report the number of valid molecules and the number of novel samples (out of 1024 generations). All setup details are provided in appendix B.1.

4.1 Inference-time context-weighted sampling

Context-weighted sampling improves generation quality at negligible computational cost. On OpenWebText with uniform source noise, the Neighbor-weighted solver improves MAUVE by up to 23% and reduces generative perplexity by 22% relative to Euler, while preserving comparable entropy (see fig. 3). Similar perplexity gains are observed with masked-source noise (see fig. 7 in the Appendix).

The two solvers behave differently across compute regimes. Entropy-weighted sampling is stronger at low NFE, whereas Neighbor-weighted sampling benefits from more update steps: it overtakes Euler in MAUVE from NFE $= 256$ and in perplexity from NFE $= 512$, and eventually surpasses

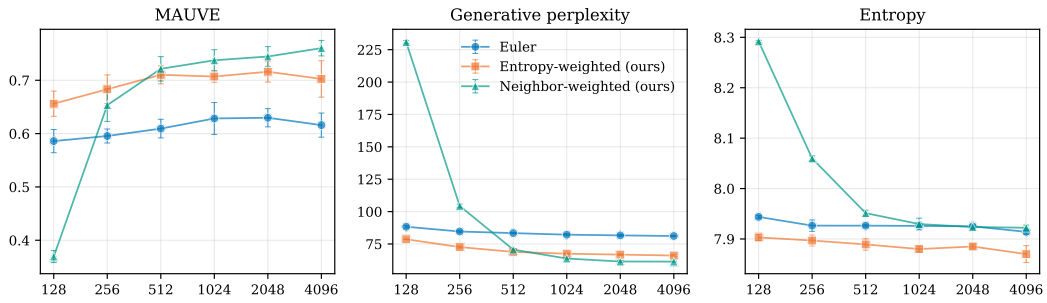

Figure 3: **Neighbor-weighted sampling improves generation quality without fine-tuning or additional model evaluations.** At larger NFE, it outperforms both the Euler baseline and the model-uncertainty-based Entropy-weighted solver in MAUVE and generative perplexity, while maintaining comparable token-level entropy.

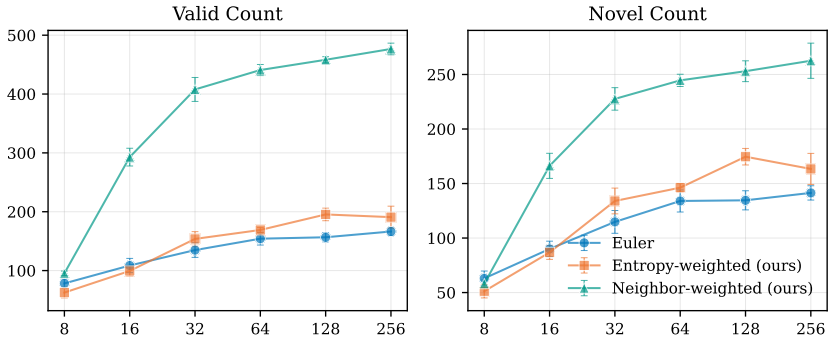

Figure 4: **Local context can be more reliable than model uncertainty in low-data regimes.** On the small QM9 dataset, Neighbor-weighted sampling outperforms both Euler and Entropy-weighted baselines across all NFE levels, nearly tripling the number of valid molecules and doubling the number of novel molecules.

Entropy-weighted sampling from NFE $= 1024$. We attribute the weaker low-NFE behavior to local inconsistencies caused by neighboring tokens being updated too independently when only few updates are performed.

Local context information is robust in the low-data regime. On QM9, Neighbor-weighted sampling yields much larger gains than Entropy-weighted sampling, increasing valid molecules by approximately $\times 2.8$ and novel molecules by approximately $\times 1.9$ over Euler (see fig. 4). This suggests that, when data is limited, model-based uncertainty estimates can be noisy, while local context remains a stable signal for guiding updates.

Context-weighted sampling is complementary to predictor–corrector schemes. To test whether local-context weighting provides gains beyond stronger samplers, we combine Neighbor-weighted scaling with the predictor–corrector framework of Gat et al. (2024). We search over correctors of the form

$$\eta_t = 1 + ct^a(1-t)^b,$$

with $a = b$ to reduce the search space. On a pre-trained DFM with uniform source noise, the combined sampler further improves MAUVE and generative perplexity, at the cost of lower entropy (see fig. 5).

4.2 Train-time modifications and ablations

SCE approaches strong semi-autoregressive baselines. For a uniform source, SCE increases MAUVE from 0.628 to 0.777 and **reduces generative perplexity by 63%**, from 82.17 to 30.20

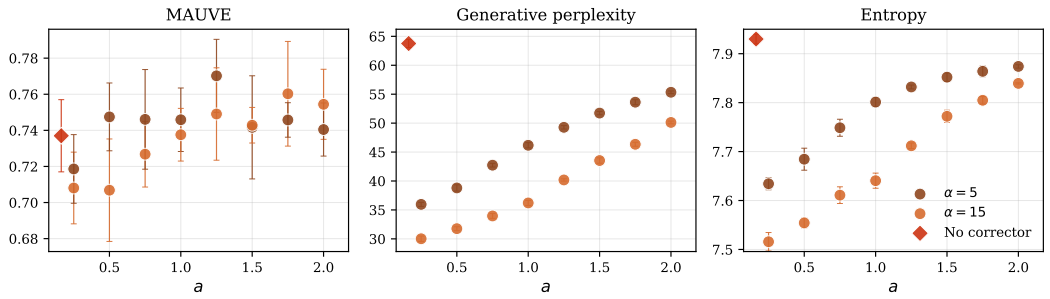

Figure 5: **Context-weighted sampling is complementary to predictor–corrector sampling.** Adding Neighbor-weighted updates to a predictor–corrector sampler further lowers generative perplexity and slightly improves MAUVE, showing that local-context weighting provides an additional quality-improving mechanism.

Table 1: Context-weighted Discrete Flow Matching approaches strong semi-autoregressive block diffusion baselines in MAUVE while substantially improving over non-autoregressive baselines.

Type	Method	MAUVE (↑)	Gen. PPL (↓)	Entropy (↑)
Data	-	0.890±0.008	14.66±0.13	7.86±0.01
(Semi-)AR	AR	0.817±0.017	**12.72**±0.25	7.57±0.02
	BD3LM ($L' = 4$)	0.784 ±0.025	24.19±0.08	7.69±0.01
	BD3LM ($L' = 8$)	0.734±0.022	29.47±0.08	7.70±0.01
	BD3LM ($L' = 16$)	0.727±0.023	32.95±0.39	7.71±0.01
Non-AR	SEDD	0.574±0.024	110.23±0.63	8.14±0.01
	MDLM	0.685±0.021	41.73±0.49	7.73±0.01
	DFM, Uniform	0.628±0.030	82.17±0.84	7.93±0.01
	DFM, Mask	0.672±0.024	40.90±0.08	7.64±0.01
	DFM+SCE, Uniform	0.777±0.017	30.20±0.22	7.66±0.01
	DFM+SCE, Mask	0.690±0.010	38.85±0.50	7.67±0.01

(table 1). It approaches BD3LM ($L' = 4, 8$) and outperforms SEDD and MDLM while retaining any-order generation and the standard Euler sampler.

Table 2: DFM trained with scaled cross-entropy achieves lower perplexity while maintaining reasonable entropy.

		OWT			QM9	
Source	Loss	MAUVE (↑)	Gen. PPL (↓)	Entropy (↑)	Valid (↑)	Novel (↑)
Uniform	CE	0.628 ±0.030	82.17 ±0.84	7.93±0.01	475.4±17.1	287.0±15.6
	NELBO	0.677±0.035	74.18±0.40	7.88±0.01	481.4±14.4	299.0±15.8
	Bregman	0.682±0.015	63.06±0.53	7.85±0.01	494.4±10.4	**308.0**±8.7
	SCE (ours)	**0.777**±0.017	**30.20**±0.22	7.66±0.01	**556.0**±16.1	297.6±10.1
Mask	CE	0.672±0.024	40.90±0.31	7.64±0.01	134.8±12.4	114.8±10.5
	NELBO	<u>0.685</u>±0.020	40.46±0.24	7.66±0.01	158.8±7.0	128.6±7.4
	Bregman	0.594±0.018	51.02±0.50	7.74±0.01	119.8±4.7	98.2±6.9
	SCE (ours)	**0.690**±0.010	**38.85**±0.50	7.67±0.01	**177.2**±11.3	**137.8**±7.6

SCE performs best among the considered losses. We compare CE (Gat et al., 2024), NELBO (Sahoo et al., 2024), Bregman divergence (Lipman et al., 2024), and SCE in table 2. SCE's advantage is largest on uniform-source OWT, improving MAUVE and generative perplexity while preserving reasonable entropy. On QM9 it yields the most valid molecules and competitive novelty.

Table 3: Mixture (p_t) vs. neighbor ($\tilde{p}_t$) path comparison.

		OWT			QM9	
Source	Path	MAUVE (↑)	Gen. PPL (↓)	Entropy (↑)	Valid (↑)	Novel (↑)
Uniform	p_t	**0.777**$\pm$0.017	**30.20**$\pm$0.22	7.66$\pm$0.01	556.0$\pm$16.1	297.6$\pm$10.1
	$\tilde{p}_t$	0.768$\pm$0.017	37.45$\pm$0.38	7.70$\pm$0.01	**658.4**$\pm$20.3	**310.2**$\pm$19.6
Mask	p_t	0.690$\pm$0.010	38.85$\pm$0.50	7.67$\pm$0.01	177.2$\pm$11.3	137.8$\pm$7.6
	$\tilde{p}_t$	**0.751**$\pm$0.012	**36.25**$\pm$0.52	7.65$\pm$0.01	**298.0**$\pm$13.6	**181.6**$\pm$9.62

Context-weighted paths help most in low-data settings. With SCE fixed, replacing the mixture path p_t by $\tilde{p}_t$ does not improve uniform-source OWT MAUVE or perplexity, though entropy increases slightly (table 3). Both quality metrics improve for the masked source. On QM9, validity and novelty improve for both sources, with the masked-source path **nearly doubling valid molecules**.

An intermediate neighborhood radius provides the best quality–diversity trade-off. We ablate the vicinity radius r used in the Neighbor-weighted scaling $\alpha^i(x)$ under the SCE objective (see fig. 9). As r increases, both generative perplexity and entropy decrease, indicating more confident but less diverse predictions. MAUVE peaks at $r = 3$, suggesting that moderate context aggregation best balances coherence and diversity.

4.3 Transfer to larger models and image generation

Larger text models. We apply Neighbor-weighted sampling without retraining to a public 1.3B-parameter uniform-source DFM checkpoint from FS-DFM (Karimi Monsefi et al., 2025), retaining the hyperparameters selected for the smaller model. On 1,024 generated samples evaluated with OLMo-3 7B, generative perplexity decreases from 49.75 to 43.19, MAUVE increases from 0.1923 to 0.3175, and entropy increases from 7.891 to 8.093 (table 6). These results provide initial evidence that the inference-time improvement transfers across model scales and training corpora.

Image generation. With two-dimensional neighborhoods, SCE reduces direct per-pixel CIFAR-10 FID from 14.86 to 11.14 for a uniform source at NFE $= 256$; the masked-source result changes from 3.47 to 3.51 (table 8). On the public 2B-parameter FUDOKI model (Wang et al., 2025), Neighbor-weighted sampling raises GenEval from 0.7456 to 0.7531, with effectively unchanged CLIP-IQA (table 9). These modest point-estimate changes do not establish statistical significance. FUDOKI's metric-induced path lies outside our convex-path analysis, so this is an empirical transfer result.

The complete transfer setups and tables and an Entropy-Bounded inference baseline are provided in appendix B.3.

5 Conclusion

Local context improves both sampling and training in Discrete Flow Matching. Context-weighted sampling complements predictor–corrector methods at little additional cost; SCE narrows the gap to semi-autoregressive baselines while preserving any-order generation; and context-weighted paths are especially useful in low-data regimes.

Several limitations remain. Neighbor-weighted sampling is less effective at very low NFE, and the additional 1.3B text and 2B image experiments provide initial transfer evidence. Image gains depend on the source distribution, and longer contexts and broader tasks remain to be studied. Future work should develop better proxies for token-level uncertainty and use them for more adaptive probability path design, while further narrowing the gap to strong autoregressive models.

AI USE STATEMENT

Generative AI assistance was used to adapt the manuscript's LaTeX formatting and supporting submission files to the ICLR 2027 format, make editorial revisions, and summarize previously reported experimental results in the main text.

REPRODUCIBILITY STATEMENT

The theoretical assumptions and proofs are provided in appendix A. The datasets, model architecture, optimization settings, evaluation metrics, sampling configuration, and hyperparameters are described in appendix B.1; additional sweeps and results are included in the experimental appendix.

REFERENCES

Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. *arXiv preprint arXiv:2505.24857*, 2025.

Victor Besnier, Mickael Chen, David Hurych, Eduardo Valle, and Matthieu Cord. Halton scheduler for masked generative image transformer. In *The Thirteenth International Conference on Learning Representations*, 2025.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, Chengxi Li, Chongxuan Li, Jianguo Li, Zehuan Li, Huabin Liu, Ling Liu, Guoshan Lu, Xiaocheng Lu, Yuxin Ma, Jianfeng Tan, Lanning Wei, Ji-Rong Wen, Yipeng Xing, Xiaolu Zhang, Junbo Zhao, Da Zheng, Jun Zhou, Junlin Zhou, Zhanchao Zhou, Liwang Zhu, and Yihong Zhuang. Llada2.0: Scaling up diffusion language models to 100b, 2025. URL https://arxiv.org/abs/2512.15745.

Tiwei Bie, Maosong Cao, Xiang Cao, Bingsen Chen, Fuyuan Chen, Kun Chen, Lun Du, Daozhuo Feng, Haibo Feng, Mingliang Gong, Zhuocheng Gong, Yanmei Gu, Jian Guan, Kaiyuan Guan, Hongliang He, Zenan Huang, Juyong Jiang, Zhonghui Jiang, Zhenzhong Lan, Chengxi Li, Jianguo Li, Zehuan Li, Huabin Liu, Lin Liu, Guoshan Lu, Yuan Lu, Yuxin Ma, Xingyu Mou, Zhenxuan Pan, Kaida Qiu, Yuji Ren, Jianfeng Tan, Yiding Tian, Zian Wang, Lanning Wei, Tao Wu, Yipeng Xing, Wentao Ye, Liangyu Zha, Tianze Zhang, Xiaolu Zhang, Junbo Zhao, Da Zheng, Hao Zhong, Wanli Zhong, Jun Zhou, Junlin Zhou, Liwang Zhu, Muzhi Zhu, and Yihong Zhuang. Llada2.1: Speeding up text diffusion via token editing, 2026. URL https://arxiv.org/abs/2602.08676.

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

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=GTDKo3Sv9p.

Itai Gat, Heli Ben-Hamu, Marton Havasi, Daniel Haziza, Jeremy Reizenstein, Gabriel Synnaeve, David Lopez-Paz, Brian Karrer, and Yaron Lipman. Set block decoding is a language model inference accelerator. *arXiv preprint arXiv:2509.04185*, 2025.

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

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10696–10706, 2022.

Yefei He, Yuanyu He, Shaoxuan He, Feng Chen, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Neighboring autoregressive modeling for efficient visual generation. *arXiv preprint arXiv:2503.10696*, 2025.

Amin Karimi Monsefi, Nikhil Bhendawade, Manuel Rafael Ciosici, Dominic Culver, Yizhe Zhang, and Irina Belousova. FS-DFM: Fast and accurate long text generation with few-step diffusion language models, 2025. URL https://arxiv.org/abs/2509.20624.

Filip Ekström Kelvinius, Oskar B Andersson, Abhijith S Parackal, Dong Qian, Rickard Armiento, and Fredrik Lindsten. Wyckoffdiff–a generative diffusion model for crystal symmetry. In *Forty-second International Conference on Machine Learning*, 2025.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M. Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DjJmre5IkP.

Haopeng Li, Jinyue Yang, Guoqi Li, and Huan Wang. Autoregressive image generation with randomized parallel decoding. *arXiv preprint arXiv:2503.10568*, 2025.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024. URL https://arxiv.org/abs/2412.06264.

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

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=CNicRIVIPA.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. Fineweb-edu: the finest collection of educational content, 2024. URL https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2025.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 1525–1534, 2016.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828, 2021.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=L4uaAR4ArM.

Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky TQ Chen. Flow matching with general discrete paths: A kinetic-optimal perspective. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, et al. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. In *The Eleventh International Conference on Learning Representations*, 2023.

Jin Wang, Yao Lai, Aoxue Li, Shifeng Zhang, Jiacheng Sun, Ning Kang, Chengyue Wu, Zhenguo Li, and Ping Luo. FUDOKI: Discrete flow-based unified understanding and generation via kinetic-optimal velocities, 2025. URL https://arxiv.org/abs/2505.20147.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=PEQFHRUFca.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, et al. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*, 2025.

A THEORETICAL RESULTS

A.1 DISCRETE FLOW MATCHING BACKGROUND

We consider discrete sequences $x \in \mathcal{D}^N$ of length N, where $\mathcal{D}$ is a finite vocabulary. We denote the i-th coordinate of x by x^i and the sequence without the i-th coordinate by $x^{\bar{i}}$. For $a \in \mathcal{D}$, we write $\delta(x^i, a) = \mathbf{1}[x^i = a]$ for a Dirac delta function. We consider independent coupling $(X_0, X_1) \sim p(X_0)q(X_1)$, where X_0 is sampled from a simple source noise distribution $p(X_0)$ and $X_1 \sim p_{\text{data}}$ is a data point.

Discrete Flow Matching (Gat et al., 2024) defines a probability path $p_t(x \mid x_0, x_1)$ between x_0 and x_1, and models its evolution via a Continuous-Time Markov Chain (CTMC). The dynamics are governed by the Kolmogorov forward equation:

$$\frac{d}{dt}p_t(x \mid x_0, x_1) = \sum_{z \in \mathcal{D}^N} \sum_{i=1}^{N} \delta(x^{\bar{i}}, z^{\bar{i}})\, p_t(z \mid x_0, x_1)\, u_t^i(x^i, z \mid x_0, x_1),$$

where $u_t^i(x^i, z \mid x_0, x_1)$ denotes the coordinate-wise probability velocity, specifying the rate of transition at position i.

One key aspect of the standard DFM setup is that the velocity factorizes over coordinates:

$$u_t(x, z \mid x_0, x_1) = \sum_{i=1}^{N} \delta(x^{\bar{i}}, z^{\bar{i}}) u_t^i(x^i, z \mid x_0, x_1).$$

Importantly, this factorization does not imply that the distribution $p_t(x \mid x_0, x_1)$ factorizes across coordinates. However, it enables efficient coordinate-wise sampling, as the CTMC can be simulated using an Euler discretization, updating the tokens independently of each other.

A common choice in DFM is the convex mixture path, defined independently for each coordinate as

$$p_t(x^i \mid x_0, x_1) = (1 - \kappa_t)\delta(x^i, x_0^i) + \kappa_t \delta(x^i, x_1^i),$$

where κ_t is a monotonically increasing scheduler. The corresponding coordinate-wise velocity is then given by

$$u_t^i(x^i, z \mid x_0, x_1) = \frac{\dot{\kappa}_t}{1 - \kappa_t} \left[\delta(x^i, x_1^i) - \delta(x^i, z^i)\right].$$

This construction leads to a simple and efficient training and sampling procedure, but treats all coordinates uniformly, regardless of their conditioning context.

At inference time, the denoiser supplies proposals $\widehat{X}_1^i \sim p_{1|t}^{\theta,i}(\cdot \mid X_t)$ instead of a known data endpoint. The baseline finite step keeps X_t^i or replaces it with $\widehat{X}_1^i$ with probability $1 - \exp(-h\lambda_t)$ when the two values differ. Proposals are redrawn at the next model evaluation; with a uniform source, repeated corrections are possible. Section 3.2 describes the corresponding context-weighted update.

A.2 BOUNDARY CONDITIONS AND MARGINALS

We prove Theorem 1 in three steps. First, we show that under the standard convex path, the number of unmasked tokens follows a pure-birth CTMC with binomial marginals. Second, we show that the normalization condition on α makes the context-weighted path induce the same CTMC for this count process, which gives the endpoint constraints. Finally, under circular-shift equivariance, the reveal probabilities are equal in the collision-free case, so matching the count distribution also gives the correct per-coordinate marginals.

Let $\mathcal{I} = \{i : x_0^i \neq x_1^i\}$, $n = |\mathcal{I}|$, and $\mathcal{A}_t = \{i \in \mathcal{I} : X_t^i \neq x_1^i\}$. For a path X_t, define the number of unmasked tokens within the active set

$$M_t = \sum_{i \in \mathcal{I}} \mathbf{1}[X_t^i = x_1^i].$$

Under the standard convex path, each active coordinate is unmasked independently with probability κ_t, hence

$$M_t \sim \text{Binomial}(n, \kappa_t).$$

Collision coordinates are not counted in M_t; they remain equal to x_1 throughout. All count statements are conditional on (x_0, x_1). The case $n = 0$ is immediate. The following proposition describes the corresponding count process.

Proposition 1. *M_t is a CTMC*

$$\mathbb{P}(M_{t+h} = y \mid M_t = x) = \delta(y, x) + hq(y, x) + o(h)$$

where

$$q(y, x) = \begin{cases} (n-x)\frac{\dot\kappa_t}{1-\kappa_t}, & y = x+1 \\ -(n-x)\frac{\dot\kappa_t}{1-\kappa_t}, & y = x \\ 0, & \textit{otherwise} \end{cases}$$

Proof. Let's derive the transition probabilities from $p_t(m) = \mathbb{P}(M_t = m)$:

$$\begin{aligned}
\frac{d}{dt}p_t(m) &= \frac{d}{dt}\binom{n}{m}(\kappa_t)^m(1-\kappa_t)^{n-m} \\
&= m\binom{n}{m}(\kappa_t)^{m-1}(1-\kappa_t)^{n-m}\dot\kappa_t + (n-m)\binom{n}{m}(\kappa_t)^m(1-\kappa_t)^{n-m-1}(-\dot\kappa_t) \\
&= \frac{\dot\kappa_t}{1-\kappa_t}m\binom{n}{m}(\kappa_t)^{m-1}(1-\kappa_t)^{n-(m-1)} - \frac{\dot\kappa_t}{1-\kappa_t}(n-m)\binom{n}{m}(\kappa_t)^m(1-\kappa_t)^{n-m} \\
&= \frac{\dot\kappa_t}{1-\kappa_t}(n-(m-1))\binom{n}{m-1}(\kappa_t)^{m-1}(1-\kappa_t)^{n-(m-1)} - \frac{\dot\kappa_t}{1-\kappa_t}(n-m)p_t(m) \\
&= \frac{\dot\kappa_t}{1-\kappa_t}(n-(m-1))p_t(m-1) - \frac{\dot\kappa_t}{1-\kappa_t}(n-m)p_t(m)
\end{aligned}$$

Note the transition probabilities

$$\begin{aligned}
q_t(m, m-1) &= \frac{\dot\kappa_t}{1-\kappa_t}(n-(m-1)) \\
q_t(m, m) &= -\frac{\dot\kappa_t}{1-\kappa_t}(n-m)
\end{aligned}$$

and so

$$\frac{d}{dt}p_t(m) = q_t(m, m-1)p_t(m-1) + q(m, m)p_t(m) = \sum_{z=0}^{n} q(m, z)p_t(z)$$

which is exactly the Kolmogorov forward equation with $q(y, x)$. □

We next show that the context-weighted path has the same count process whenever the weights are normalized over the masked coordinates. This is the key property needed for the endpoint constraints: if $\tilde{M}_t$ has the same law as M_t, then $\tilde{M}_0 = 0$ and $\tilde{M}_1 = n$ almost surely.

Proposition 2. *A sufficient condition for $\tilde{M}_t \sim$ Binomial(n, κ_t) is:*

$$\sum_{i \in \mathcal{I}} \alpha^i(x_t) \cdot \delta(x_t^i, x_0^i) = \sum_{i \in \mathcal{I}} \delta(x_t^i, x_0^i)$$

Proof. First, note that probability of M_t not changing in $[t, t+h)$ is the same as probability of X_t not changing:

$$\begin{aligned}\mathbb{P}(X_{t+h} = x \mid X_t = x, X_0 = x_0, X_1 = x_1) &= 1 + h \sum_{i \in \mathcal{I}} u_t^i(x^i, x \mid x_0, x_1) + o(h) \\ &= 1 + h \sum_{i \in \mathcal{I}} \frac{\dot{\kappa}_t}{1 - \kappa_t} \left[\delta(x^i, x_1^i) - 1\right] + o(h) \\ &= 1 - h \frac{\dot{\kappa}_t}{1 - \kappa_t} (n - \sum_{i \in \mathcal{I}} \delta(x^i, x_1^i)) + o(h) \\ &= 1 - h \frac{\dot{\kappa}_t}{1 - \kappa_t} (n - m) + o(h) \\ &= \mathbb{P}(M_{t+h} = m \mid M_t = m)\end{aligned}$$

which makes sense since x_1^i is absorbing for each token, so no cyclic movements ($x_0^i \to x_1^i \to x_0^i$) are allowed. Now, if we plug in v_t^i instead of u_t^i in the same derivation

$$\begin{aligned}\mathbb{P}(\tilde{X}_{t+h} = x \mid \tilde{X}_t = x, \tilde{X}_0 = x_0, \tilde{X}_1 = x_1) &= 1 + h \sum_{i \in \mathcal{I}} v_t^i(x^i, x \mid x_0, x_1) + o(h) \\ &= 1 - h \frac{\dot{\kappa}_t}{1 - \kappa_t} \sum_{i \in \mathcal{I}} \alpha^i(x) \delta(x^i, x_0^i) + o(h)\end{aligned}$$

so if $\sum_{i \in \mathcal{I}} \alpha^i(x) \delta(x^i, x_0^i) = \sum_{i \in \mathcal{I}} \delta(x^i, x_0^i) = n - m$, we have

$$P(\tilde{X}_{t+h} = x \mid \tilde{X}_t = x, \tilde{X}_0 = x_0, \tilde{X}_1 = x_1) = \mathbb{P}(X_{t+h} = x \mid X_t = x, X_0 = x_0, X_1 = x_1)$$

and so $\tilde{M}_t$ is following the same CTMC as M_t. □

Thus the normalization condition preserves the distribution of the number of unmasked tokens. Since $\kappa_0 = 0$ and $\kappa_1 = 1$, it follows that $\tilde{M}_0 = 0$ and $\tilde{M}_1 = n$ almost surely, while coordinates outside $\mathcal{I}$ already equal the target. This implies

$$\tilde{p}_0(x \mid x_0, x_1) = \delta(x, x_0), \qquad \tilde{p}_1(x \mid x_0, x_1) = \delta(x, x_1).$$

Equivalently, the normalization condition requires the weights to average to one over the currently masked coordinates:

$$\frac{1}{|\{i \in \mathcal{I} \colon z^i = x_0^i\}|} \sum_{i \in \mathcal{I} \colon z^i = x_0^i} \alpha^i(z) = 1.$$

This controls only the total number of updates, not which coordinates receive them. For example, a positional bias such as $\alpha^i(z) \propto (N - i)$ can preserve the total rate while making left-most coordinates transition earlier. To recover the standard per-coordinate marginals, we additionally require the weighting rule to treat positions symmetrically. Let $\mathrm{shift}_k(x)$ denote a circular shift of x by k positions, i.e.

$$\mathrm{shift}_k(x)^i = x^{i-k \pmod N}.$$

Proposition 3. *Let $B_t \in \{0,1\}^N$ denote the binary reveal mask, where $B_t^i = 1$ if $\tilde{X}_t^i = x_1^i$, and $B_t^i = 0$ otherwise. Suppose that the context weights depend only on the reveal mask and active set, i.e., there exists a function*

$$\bar{\alpha} : \{0,1\}^N \times 2^{\{1,\dots,N\}} \to \mathbb{R}_+^N$$

such that

$$\alpha^i(\tilde{X}_t) = \bar{\alpha}^i(B_t, \mathcal{I}) \qquad \forall i.$$

Assume that $\bar{\alpha}$ is equivariant under circular shifts:

$$\bar{\alpha}^{i+k}\big(\mathrm{shift}_k(b), \mathrm{shift}_k(\mathcal{I})\big) = \bar{\alpha}^i(b, \mathcal{I}) \qquad \forall b \in \{0,1\}^N,\ \forall i, k,$$

with indices taken modulo N, and that the normalization condition in proposition 2 holds. If $\mathcal{I} = \{1, \dots, N\}$, then

$$\mathbb{P}(B_t^i = 1 \mid x_0, x_1) = \mathbb{P}(\tilde{X}_t^i = x_1^i \mid x_0, x_1) = \kappa_t \qquad \forall i.$$

Proof. Consider the collision-free case $\mathcal{I} = \{1, \dots, N\}$. Let

$$S_t = \{i : B_t^i = 1\}$$

be the set of revealed coordinates, so that

$$|S_t| = \tilde{M}_t.$$

Since $\bar{\alpha}$ is equivariant under circular shifts, shifting the reveal mask and the corresponding coordinate by the same offset preserves all transition rates. Since $B_0 = \mathbf{0}$ is invariant under circular shifts, the law of B_t is also invariant under circular shifts. Hence, for every offset k,

$$\mathbb{P}(S_t = S \mid x_0, x_1) = \mathbb{P}\big(S_t = \mathrm{shift}_k(S) \mid x_0, x_1\big).$$

Thus all coordinates have the same marginal probability of being revealed. Indeed, for any i, j, there exists a circular shift mapping i to j, and therefore

$$\mathbb{P}(i \in S_t \mid x_0, x_1) = \mathbb{P}(j \in S_t \mid x_0, x_1).$$

Denote this common probability by p_t. Then

$$\mathbb{E}[\tilde{M}_t \mid x_0, x_1] = \mathbb{E}[|S_t| \mid x_0, x_1] = \sum_{i=1}^{N} \mathbb{P}(i \in S_t \mid x_0, x_1) = N p_t.$$

By proposition 2, the normalization condition implies

$$\tilde{M}_t \sim \mathrm{Binomial}(N, \kappa_t),$$

and hence

$$\mathbb{E}[\tilde{M}_t \mid x_0, x_1] = N\kappa_t.$$

Combining the two equalities gives

$$p_t = \kappa_t.$$

Therefore,

$$\mathbb{P}(\tilde{X}_t^i = x_1^i \mid x_0, x_1) = \kappa_t.$$

□

Now, we are ready to prove the theorem 1.

Proof of theorem 1. By proposition 2, $\tilde{M}_t$ has the same law as the active count process of the standard convex path, namely $\mathrm{Binomial}(|\mathcal{I}|, \kappa_t)$. Since $\kappa_0 = 0$ and $\kappa_1 = 1$, all active coordinates reach x_1, while collision coordinates are already there. This gives the endpoint constraints. Under the additional symmetry condition, proposition 3 gives the stated coordinate-wise marginals. □

A.3 Correctness of sampling

Solver normalization and marginals. Both Neighbor-weighted and Entropy-weighted solvers are normalized over the currently masked coordinates. If $\tilde{\alpha}^i(x_t)$ denotes the raw score, we set

$$\alpha^i(x_t) \propto \exp(s\tilde{\alpha}^i(x_t)), \qquad i \in \mathcal{A}_t = \{i \in \mathcal{I} : x_t^i \neq x_1^i\},$$

and choose the proportionality constant so that

$$\frac{1}{|\mathcal{A}_t|} \sum_{i \in \mathcal{A}_t} \alpha^i(x_t) = 1.$$

Therefore both solvers satisfy the normalization condition in proposition 2 and preserve the distribution of the number of updated active tokens.

For Neighbor-weighted, if local windows are computed with circular padding, the rule is shift-equivariant: shifting the sequence shifts all neighbor counts by the same amount. Hence all positions are treated symmetrically, and the per-coordinate marginals, in the scope stated there, are preserved by proposition 3.

For Entropy-weighted, the normalization still preserves the total number of updates, and therefore the endpoint constraints. However, exact circular-shift equivariance of the model predictions is not guaranteed:

$$H\Big(p^{\theta,(i+k \bmod N)}(\cdot \mid \mathrm{shift}_k(x_t))\Big) \neq H\big(p^{\theta,i}(\cdot \mid x_t)\big)$$

in general. Therefore, Entropy-weighted sampling is not covered by the per-coordinate marginal preservation result: the model may induce position- or value-dependent preferences over the reveal order.

Exactness of train-time path sampling. algorithm 1 avoids simulating the full continuous-time trajectory by separating the process into the number of jumps and their identities. By proposition 2, the number of updated active tokens at time t is preserved:

$$\tilde{M}_t \sim \mathrm{Binomial}(|\mathcal{I}|, \kappa_t).$$

For time-independent weights, we can first sample $m \sim \mathrm{Binomial}(|\mathcal{I}|, \kappa_t)$. It remains to sample which m coordinates are unmasked. Under the context-weighted CTMC, if the current state is x, a masked coordinate i has transition rate

$$\alpha^i(x)\frac{\dot{\kappa}_t}{1-\kappa_t}.$$

Therefore, conditioned on a jump occurring, the next coordinate is selected with probability

$$\frac{\alpha^i(x)}{\sum_{j\in\mathcal{I}:x^j\neq x_1^j}\alpha^j(x)},$$

since the time-dependent factor cancels. The total exit rate depends only on the count and time, so the jump count can be sampled independently of the embedded sequence of coordinate choices. This is exactly the sequential selection rule used in algorithm 1. Hence the algorithm samples the same time marginal $\tilde{p}_t$ as the continuous-time process.

A.4 Evidence Lower Bound (ELBO)

In the case of the mixture path, Shaul et al. (2025) presents a formula for evidence lower bound (ELBO)

$$\log p_1^\theta(x_1) \geq -\mathbb{E}_{t,X_0,X_t\sim p_{t|0,1}} \sum_i D\left(u_t^i(\cdot, X_t^i \mid X_0, x_1), u_t^{\theta,i}(\cdot, X_t)\right)$$

where

$$\begin{aligned} &D\left(u_t^i(\cdot, x^i \mid x_0, x_1), u_t^{\theta,i}(\cdot, x)\right) \\ &\quad = \frac{\dot{\kappa}_t}{1-\kappa_t}\Big[(\delta(x_1^i, x^i) - 1)\log p_{1|t}^{\theta,i}(x_1^i \mid x) + \delta(x_1^i, x^i) - p_{1|t}^{\theta,i}(x^i \mid x)\Big]. \end{aligned}$$

Proposition 4. *Consider the context-weighted velocity*

$$v_t^i(\cdot, x \mid x_0, x_1) = \alpha^i(x) u_t^i(\cdot, x \mid x_0, x_1),$$

where $\alpha^i(x) \geq 0$ does not depend on the target state. Let the learned marginal velocity be parameterized as

$$v_t^{\theta,i}(\cdot, x) = \alpha^i(x) u_t^{\theta,i}(\cdot, x).$$

For the CTMC rate divergence

$$D(a,b) = a\log\frac{a}{b} - a + b,$$

applied coordinate-wise over transition rates, we have

$$D\Big(v_t^i(\cdot, x \mid x_0, x_1), v_t^{\theta,i}(\cdot, x)\Big) = \alpha^i(x) D\Big(u_t^i(\cdot, x \mid x_0, x_1), u_t^{\theta,i}(\cdot, x)\Big).$$

Thus the corresponding NELBO term is reweighted by $\alpha^i(x)$.

Proof. Since $v_t^i = \alpha^i(x) u_t^i$ and $v_t^{\theta,i} = \alpha^i(x) u_t^{\theta,i}$, it is enough to use the positive homogeneity of the CTMC rate divergence. For scalar rates,

$$D(\alpha a, \alpha b) = \alpha a \log\frac{\alpha a}{\alpha b} - \alpha a + \alpha b = \alpha\left(a\log\frac{a}{b} - a + b\right) = \alpha D(a,b).$$

Summing over possible target states preserves the same factor, since $\alpha^i(x)$ is independent of the target state. Therefore,

$$D\Big(v_t^i(\cdot, x \mid x_0, x_1), v_t^{\theta,i}(\cdot, x)\Big) = \alpha^i(x) D\Big(u_t^i(\cdot, x \mid x_0, x_1), u_t^{\theta,i}(\cdot, x)\Big).$$

□

A.5 Loss Functions Analysis

The Bregman divergence objective uses the rate divergence displayed above.

The loss is scaled by the jump coefficient $\dot{\kappa}_t/(1-\kappa_t)$, which depends only on time. For common schedules this coefficient increases as t approaches 1, giving more weight to later, less-masked states and less weight to heavily masked ones. This captures a coarse notion of prediction difficulty through the amount of global context, but it does not distinguish tokens within the same state. In particular, two masked tokens at the same time step receive the same global scaling even if one has many visible neighbors and the other has little local context. Therefore, Bregman and NELBO-style objectives (Sahoo et al., 2024) still mix token-level prediction problems of different difficulty, motivating an explicit local-context weighting.

B Experiments

B.1 Experimental setup

Datasets. We use OpenWebText (OWT) (Gokaslan & Cohen, 2019) as our main training dataset to remain comparable with standard text diffusion baselines such as MDLM and BD3LM. OWT contains roughly 8M web documents, making it a suitable large-scale text benchmark. As a smaller alternative from a different domain, we use QM9 (Ramakrishnan et al., 2014), which contains about 134k small molecules. We represent QM9 molecules in SMILES format, a linear encoding of molecular graphs, where local token neighborhoods often correspond to meaningful local chemical structure; therefore, locality remains a relevant inductive signal.

Metrics. For text, we report generative perplexity, computed with GPT-2(Radford et al., 2019), as a proxy for coherence; token entropy, computed with $\log_2$ as in DFM (Gat et al., 2024), as a measure of token-level diversity; and MAUVE (Pillutla et al., 2021) as our main quality–diversity metric. For QM9, we report the number of valid SMILES strings (Valid) and the number of generated molecules that do not appear in the training set (Novel), both measured out of 1024 generated samples. For all metrics except MAUVE, we estimate standard deviations by splitting generated samples into 5 folds. Since MAUVE is a two-sample metric and depends on sample size, we instead estimate its variance using 5 bootstrap resamples. For the data row in table 1, we compute MAUVE between two independent subsets of 5000 validation samples. Although the theoretical upper bound is 1.0 in the infinite-sample limit, we report this finite-sample estimate as a more realistic reference point and a fairer comparison for generated samples of the same size.

Training and inference. For training models with Discrete Flow Matching, we employ the Transformer model, with the same architecture and setup as in Sahoo et al. (2024). On OWT, we train with a total batch size of 512 samples with a fixed length of 1024 for 1 million steps. We use AdamW optimizer (Loshchilov & Hutter, 2019), setting $\beta_1 = 0.9$ and $\beta_2 = 0.999$, with peak learning rate of 3×10^{-4} and cosine annealing to 3×10^{-6}, and a warmup of 2.5k steps. The training takes around 4.5 days on 8 NVIDIA H100 GPUs. Training on QM9 took 30 minutes with the same GPU setup. For testing models trained on OWT, we follow Sahoo et al. (2024) and generate 5000 samples with nucleus sampling $p = 0.9$. For models trained on QM9, we set $p = 1.0$ (no nucleus sampling) and generate 5120 samples in total. See complete setup in table 4.

Table 4: Training, inference and flow hyperparameters

Parameter	OWT	QM9
# parameters	169M	92M
Batch size	512	2048
Length	1024	32
Peak LR	3×10^{-4}	3×10^{-5}
Min LR	3×10^{-6}	3×10^{-7}
warmup	2500	1000
Weight decay	0.03	0.01
(β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)
# training steps	1000k	25k
Scheduler $\kappa(t)$	t^2	t^2
# gen. samples	5000	5120
Nucleus p	0.9	1.0
Temperature	1.0	1.0

Baselines. Our main direct baseline is standard Discrete Flow Matching (DFM) (Gat et al., 2024), trained and sampled with the same architecture, data, and evaluation setup as our method. We also compare against autoregressive and semi-autoregressive language modeling baselines, including AR and BD3LM (Arriola et al., 2025) with different block lengths L', which provide strong references for generation quality under more structured decoding orders. Finally, we include representative discrete diffusion baselines, SEDD (Lou et al., 2024) and MDLM (Sahoo et al., 2024), to compare against non-autoregressive diffusion-style models trained on the same text generation setting.

B.2 Sweeps

Inference-time experiments. For neighbor-weighted and entropy-weighted solvers, we run the sweep on $r \in \{1, 2, 3\}$, $s \in \{0.5, 1.0, 2.0, 3.0, 4.0, 5.0\}$ and $\beta \in \{1.0, 2.0, \ldots, 7.0\}$.

Table 5: Best hyperparameters for Entropy-weighted (β) and Neighbor-weighted (r, s) solvers

Source	Dataset	r	s	β
Uniform	OWT	1	5.0	6.0
	QM9	1	4.0	7.0
Mask	OWT	1	5.0	6.0
	QM9	1	4.0	6.0

Trained models. For SCE-trained models, we use only Euler solver, no hyperparameter sweeps. We find that using Neighbor-weighted solver provides no improvement. For CW-DFM ($\tilde{p}_t$), on OWT, we set $r = 3$ and run training with $s \in \{0.5, 1.0\}$. We run the sweep on neighbor-weighted solver with the trained model on $r \in \{1, 2, 3\}$ and $s \in \{0.5, 1.0, 2.0\}$. For training CW-DFM and SCE in QM9, we performed the sweep in $r \in \{1, 2, 3\}$ and $s \in \{0.5, 1.0, 2.0\}$. For CW-DFM, we also run the sweep at the test time to find the optimal neighbor solver parameters.

B.3 ADDITIONAL RESULTS

Scaling to larger models. To evaluate transfer beyond the approximately 170M-parameter models in the main experiments, we apply NW sampling without retraining to the public 1.3B-parameter uniform-source DFM checkpoint from FS-DFM (Karimi Monsefi et al., 2025), trained on FineWeb-Edu (Lozhkov et al., 2024). We retain the NW hyperparameters from the smaller text model and evaluate 1,024 generated samples, using OLMo-3 7B for generative perplexity and MAUVE features.

Table 6: Inference-only transfer to a 1.3B uniform-source DFM. The FineWeb-Edu MAUVE reference compares two independent data samples.

Corpus / sampler	Gen. PPL ↓	Entropy ↑	MAUVE ↑
DFM baseline	$49.75_{\pm 1.16}$	$7.891_{\pm 0.013}$	0.1923
DFM + NW	$\mathbf{43.19}_{\pm 0.87}$	$\mathbf{8.093}_{\pm 0.011}$	**0.3175**
FineWeb-Edu	$8.746_{\pm 0.250}$	$7.931_{\pm 0.014}$	0.9310

NW improves all three metrics relative to the DFM baseline (table 6). Generative perplexity falls by approximately 13%, while MAUVE increases from 0.1923 to 0.3175 and entropy increases. Transferring the same hyperparameters provides initial evidence of robustness across these two model scales and training corpora.

Additional inference-time baseline. We evaluate the Entropy-Bounded (EB) sampler (Ben-Hamu et al., 2025) on our masked-source DFM checkpoint with entropy budget $\gamma = 0.001$ (table 7). EB adaptively selects how many positions to reveal per model evaluation. At nucleus probability $p = 0.9$, its low generative perplexity of 15.91 is accompanied by low entropy (6.65) and MAUVE (0.50). Removing nucleus truncation raises entropy to 7.56, but perplexity rises to 72.09 and MAUVE remains 0.49. On this checkpoint and budget, EB does not improve the quality–diversity balance over the main masked-source DFM results. EB assumes absorbing masked dynamics and does not directly support uniform-source self-correction.

Table 7: Entropy-Bounded sampling on the masked-source DFM checkpoint, with $\gamma = 0.001$.

Nucleus p	MAUVE ↑	Gen. PPL ↓	Entropy ↑
0.9	0.50	15.91	6.65
1.0	0.49	72.09	7.56

Transfer to image generation. We first apply DFM+SCE to direct per-pixel CIFAR-10 generation at NFE $= 256$, using two-dimensional neighborhoods. A pixel is counted as agreeing only when all its color channels equal the training target, $b^i = \mathbf{1}[\boldsymbol{x}_t^i = \boldsymbol{x}_1^i]$. With a uniform source, FID improves from 14.86 to 11.14, a reduction of approximately 25%. With a masked source, FID changes from 3.47 to 3.51, providing no improvement over the stronger baseline (table 8).

Table 8: Direct per-pixel CIFAR-10 generation with two-dimensional SCE neighborhoods, NFE $= 256$.

Source	Baseline FID ↓	DFM+SCE FID ↓
Uniform	14.86	**11.14**
Mask	**3.47**	3.51

We also adapt NW sampling to the 24×24 VQ-token grid of the public 2B-parameter FUDOKI model (Wang et al., 2025). FUDOKI starts from uniform VQ tokens and allows iterative self-correction. NW raises GenEval from 0.7456 to 0.7531, while CLIP-IQA is effectively unchanged (0.7348 versus 0.7354; table 9). These are modest point-estimate improvements; without uncertainty estimates, they do not establish statistical significance. FUDOKI uses a metric-induced probability

path outside the standard convex-path analysis, so this experiment demonstrates possible empirical transfer rather than a direct application of theorem 1.

Table 9: NW transfer to the 2B-parameter FUDOKI VQ-token image model.

Metric	Baseline	NW
GenEval ↑	0.7456	**0.7531**
CLIP-IQA ↑	0.7348	**0.7354**

Local context under a uniform source. We repeat the motivating locality analysis with a uniform-source DFM on QM9. At both $t = 0.4$ and $t = 0.6$, increasing the number of endpoint-agreeing neighbors lowers the prediction NLL and entropy (table 10), reproducing the masked-source trend. These measurements use the known data endpoint in conditional-path analysis. Values are in nats and uncertainties are cluster-robust 95% confidence-interval half-widths, rather than standard deviations.

Table 10: Uniform-source QM9 locality analysis; mean ± cluster-robust 95% confidence-interval half-width, in nats.

t	Agreeing neighbors	NLL ↓	Entropy ↓
0.4	2	$1.697_{\pm 0.011}$	$1.652_{\pm 0.005}$
0.4	4	$1.365_{\pm 0.028}$	$1.584_{\pm 0.013}$
0.4	6	$0.972_{\pm 0.170}$	$1.350_{\pm 0.092}$
0.6	2	$1.465_{\pm 0.015}$	$1.200_{\pm 0.007}$
0.6	4	$1.131_{\pm 0.016}$	$1.087_{\pm 0.008}$
0.6	6	$0.728_{\pm 0.035}$	$0.814_{\pm 0.020}$

Entropy, NLL and Local context. To illustrate the locality bias, we took a pre-trained DFM baseline model with masked source distribution trained on OpenWebText (OWT) (Gokaslan & Cohen, 2019). We sample x_1 from the validation subset and sample $x_t \sim p_t(x \mid x_0, x_1)$ at various timesteps for a total of 16k samples. For each masked token i in each x_t, we measured the entropy of the model prediction $H\left(p^{\theta,i}_{1|t}(\cdot \mid x_t)\right)$ and the negative log-likelihood of the correct token, $\mathcal{L}_i = -\log p^{\theta,i}_{1|t}(x^i_1 \mid x_t)$, a.k.a. the error of the prediction. We then evaluated two simple locality statistics: the number of unmasked tokens within a radius $r = 8$, and the distance to the nearest unmasked token (see fig. 6). In both cases, stronger local context corresponds to lower entropy and lower NLL: increasing the number of visible neighbors decreases both metrics, while increasing the distance to the nearest visible token increases them. This suggests that several simple locality measures can serve as inexpensive proxies for prediction uncertainty.

Training dynamics. fig. 10 shows that SCE follows a smooth training trajectory and leads to substantially lower generative perplexity. The validation objectives are not directly comparable, since CE and SCE use different token weightings. The accompanying moderate entropy decrease suggests that SCE makes generations more coherent without collapsing diversity.

Table 11: Test perplexity (↓) for a variety of datasets.

Model	Source	Wikitext103	LAMBADA	FineWeb-Edu
DFM	Uniform	≤ 63.38	≤ 84.33	≤ 48.20
DFM+SCE	Uniform	≤ 91.53	≤ 133.25	≤ 72.90
DFM	Mask	≤ 45.33	≤ 57.88	≤ 35.00
DFM+SCE	Mask	≤ 45.27	≤ 59.59	≤ 34.80

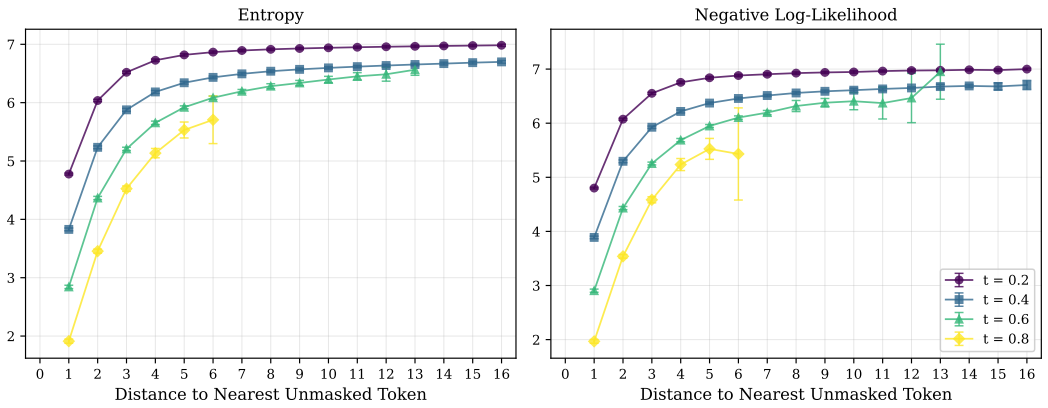

Figure 6: **The distance to the nearest unmasked token provides another proxy for the token-level uncertainty.** Both the entropy and negative log-likelihood increase with the distance, but saturate quite early. Vicinity radius $r = 16$, $\kappa(t) = t^2$.

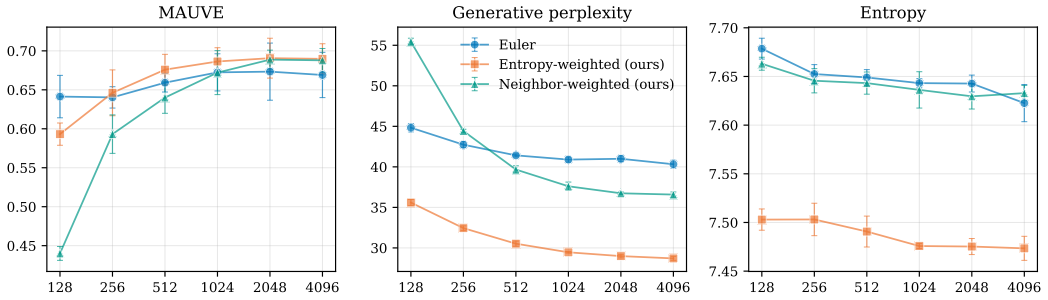

Figure 7: **Context-weighted sampling remains effective with masked source noise.** With a masked source distribution, context-weighted solvers improve generative perplexity over Euler sampling while maintaining comparable token-level entropy.

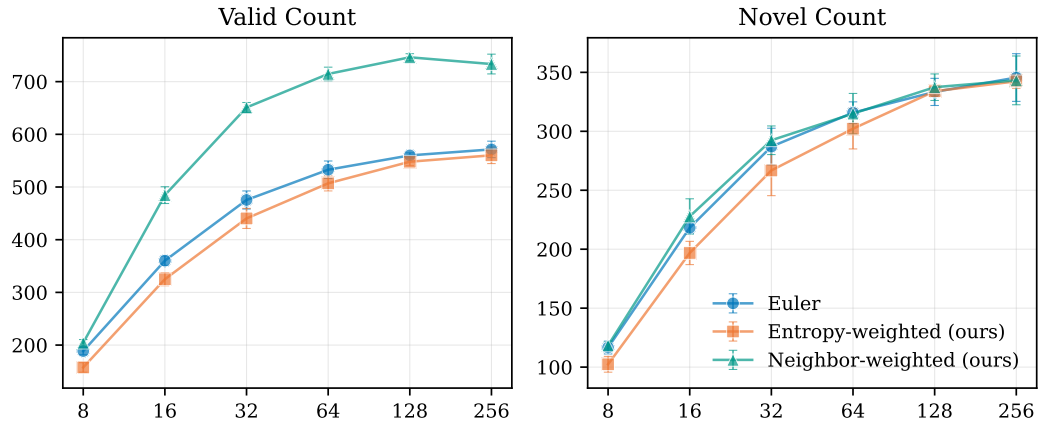

Figure 8: **Neighbor-weighted sampling improves molecular validity with uniform source noise.** On QM9, Neighbor-weighted sampling generates more valid molecules than the baselines across NFE, while maintaining a comparable number of novel molecules out of 1024 samples.

Test perplexity. We report test perplexity on Wikitext103(Merity et al., 2016), LAMBADA(Paperno et al., 2016), and FineWeb-Edu (Lozhkov et al., 2024) datasets in table 11. We

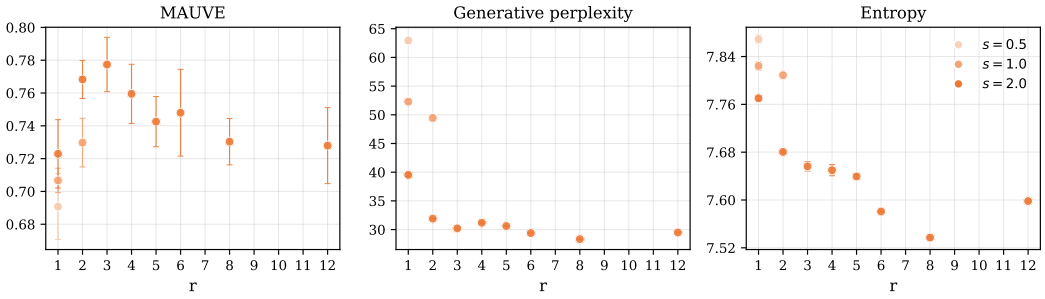

Figure 9: **Moderate local neighborhoods give the best quality–diversity trade-off.** For DFM trained with SCE on OpenWebText, increasing the neighborhood radius reduces the perplexity but also the entropy; MAUVE peaks at an intermediate radius, suggesting that too little or too much context aggregation is suboptimal.

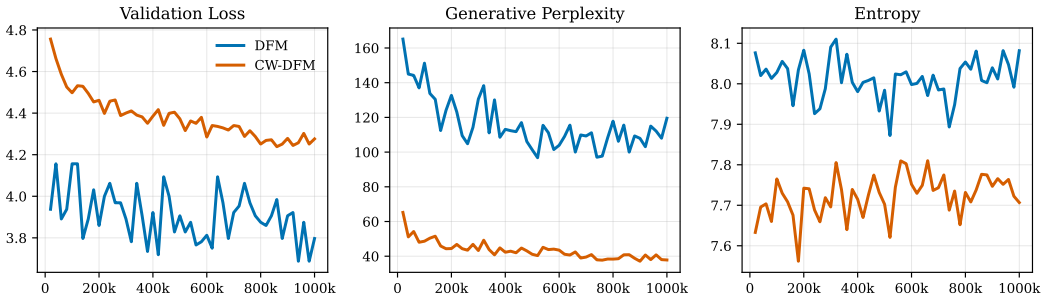

Figure 10: **SCE changes the optimization trajectory toward lower generative perplexity.** Although the weighted SCE validation loss is not directly comparable to standard CE, SCE decreases smoothly during training and yields generations with lower perplexity and a moderate reduction in entropy.

observe a discrepancy between test perplexity and generation quality: models trained with SCE can have higher test perplexity under the standard evaluation, while producing better samples according to generative perplexity, MAUVE, and qualitative inspection. This is expected because SCE deliberately changes the weighting of the prediction problem rather than optimizing the unweighted likelihood of all coordinates equally. Standard test perplexity evaluates all token predictions uniformly, including highly ambiguous positions with little available context, whereas SCE prioritizes well-contextualized predictions that provide a sharper learning signal. As a result, SCE may be less favorable under the standard perplexity objective, but better aligned with the iterative sampling process, where improving reliable local updates can compound into more coherent generations. A fully aligned likelihood-style evaluation would require accounting for the same context-dependent weighting or path used during training.

Self-correction. Naturally, we observe that Neighbor-weighted solver performs more updates per token than the standard Euler sampler (see fig. 11). For Euler solver, the average number of per-token corrections was $1.033_{\pm 0.001}$, whereas for Neighbor-weighted solver, we have $1.157_{\pm 0.001}$ corrections (uniform source). There is very little redundancy in these updates, since the proportion of loops for Neighbor-weighted sampling with uniform source is 0.02%.

Scale ablation. table 12 ablates the inverse-temperature parameter s used in the context-weighted path $\tilde{p}_t$. Increasing s makes the weighting more selective, which generally lowers generative perplexity but can also reduce entropy. The best value depends on the source distribution: for the uniform source, $s = 0.5$ gives slightly higher MAUVE and diversity, while $s = 1.0$ yields much lower perplexity; for the masked source, $s = 1.0$ provides the best overall trade-off.

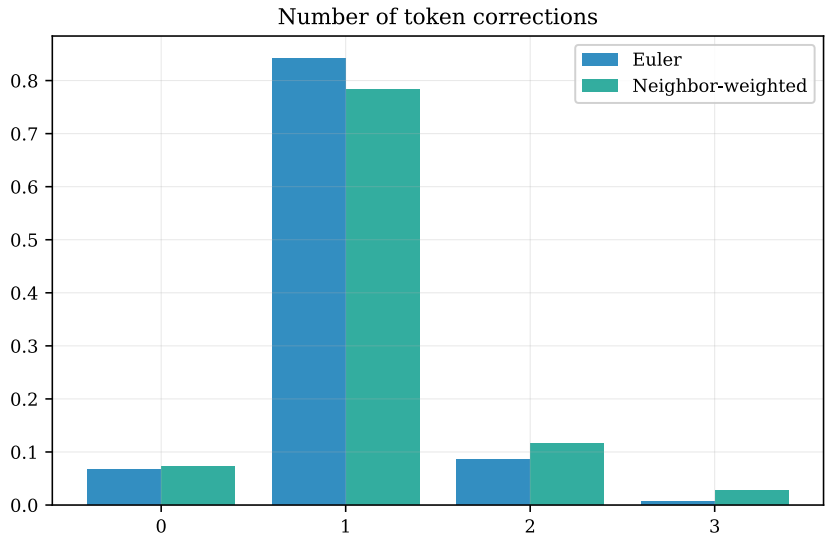

Figure 11: Distribution of the number of per-token updates for Euler and Neighbor-weighted solver, uniform source. The Neighbor-weighted solver has higher number of per-token corrections on average.

Societal impact. This work is methodological and aims to improve the efficiency and controllability of discrete generative models. Potential positive impacts include reducing sampling cost and improving the accessibility of generative modeling research. At the same time, stronger generative models can be misused for synthetic text generation at scale, including spam or disinformation. Our work does not introduce a deployed system or use sensitive data, but future applications should consider appropriate monitoring and safeguards.

Table 12: Ablation of the scale parameter s for CW-DFM ($\tilde{p}_t$).

Source	s	MAUVE	Gen. PPL	Entropy
Uniform	0.5	0.779$\pm$0.017	48.91$\pm$0.29	7.83$\pm$0.01
	1.0	0.768$\pm$0.017	37.45$\pm$0.38	7.70$\pm$0.01
Mask	0.5	0.712$\pm$0.023	35.48$\pm$0.26	7.60$\pm$0.01
	1.0	0.751$\pm$0.012	36.25$\pm$0.52	7.65$\pm$0.01