SemanticDLM+: Improving Diffusion LLMs through Bias-variance Trade-off in Transition Kernel Design

Keyue Jiang[1,4,*,†], **Yuxiang Wang**[1,2,*,†], **Xiang Yu**[1,2,*,†], **Yanan Zhao**[3], **Qifang Zhao**[1]
Bohan Tang[5], **Baojian Zhou**[2], **Yanghua Xiao**[2], **Lin Qu**[1], **Xiaoxiao Xu**[1]

[1]Alibaba Group, [2]Fudan University, [3]Nanyang Technological University,
[4] University College London, [5] University of Oxford
[*] Equal Contribution. [†] Work done during an internship at Alibaba Group.

Abstract

Diffusion Language Models (DLMs) have demonstrated strong scaling capacity as alternatives to autoregressive language models. However, their performance is highly sensitive to the choice of transition kernels, and poorly designed kernels can lead to issues like training instability, slow convergence, and biased sampling. In this paper, we study this sensitivity through a principled analysis of generalization error and identify three critical factors: asymptotic bias (difficulty in approximating the posterior distribution), exposure bias (error propagation during sampling), and optimization variance induced by kernel dispersion. We further compare different transition kernels: masking diffusion yields sparse and easier posterior-approximation targets, while uniform diffusion provides stronger sampling-side repair but induces harder approximation. Motivated by this trade-off, we revisit a previously overlooked variant, semantic DLM (SemDLM), where the transition kernel corrupts tokens to neighborhoods that are semantically similar. Our theory suggests that SemDLM can serve as a plausible middle ground by reducing the posterior approximation difficulty of uniform diffusion while retaining repair ability. However, we find that SemDLM suffers from a semantic basin problem, where sampling repeatedly stays within a semantic region and produces low-diversity text. To address this, we propose SemDLM+, which adds a global transition and a semantic-frequency penalty during sampling. Experiments on LM1B and OpenWebText show that SemDLM+ improves training dynamics and achieves competitive language modeling and generation quality with satisfactory diversity.

1 Introduction

Diffusion language models (DLMs) [42, 24] have emerged as a compelling alternative to autoregressive language models (ALMs) [9, 34, 8] due to their parallelizable training and faster decoding speeds. DLMs need to design a transition kernel to gradually corrupt the clean data to noise, and different kernels can lead to substantially different training and sampling dynamics. The predominant archetype is absorbing kernel [29, 32, 26], as prior work shows that masking DLMs [3, 14] can effectively alleviate slow convergence, training instability, and weak generalization [39] compared to others such as uniform, marginal, and semantic neighborhood diffusion [3, 14, 31, 28]. However, recent studies demonstrated that uniform DLMs can enjoy better scaling capacity when sufficient data and training budget are given [36, 35, 30]. This reveals a gap that is not fully explained:

RQ1: Why does uniform diffusion scale well under large resources, while masking diffusion remains stronger in many practical regimes? Can a transition kernel combine the advantages of both?

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

In this paper, we answer this question by developing a principled error analysis framework for DLMs. We first decompose the generation error into *Approximation Error*, *Sampling Error*, and *Forward Kernel Mismatch*, and then separate the first two through a bias-variance lens. This highlights three kernel-dependent factors: asymptotic bias that reflects approximation difficulty, exposure bias that measures error accumulation during reverse sampling, and optimization variance that captures finite-resource instability. Our analysis uncovers a trade-off in existing paradigms: masking diffusion has sparse posterior targets that are easier to fit, but provides limited intrinsic repair during sampling. Uniform diffusion has denser posterior targets that are harder to optimize, but can be preferable for sampling because its reverse dynamics naturally preserve the ability to repair earlier errors.

This trade-off motivates us to revisit a previously overlooked variant, namely semantic diffusion language models (SemDLMs), where the forward kernel corrupts a token into semantically related tokens. SemDLMs are theoretically attractive because they restrict the posterior to a meaningful neighborhood, reducing the approximation difficulty relative to uniform diffusion, while still allowing richer transitions than masking diffusion. However, prior SemDLM designs have not consistently delivered strong generation performance [3, 43]. This leads to our second question:

RQ2: While theoretically plausible, why does SemDLM still underperform existing methods? And how can we translate the theoretical advantages of SemDLM into practical gains?

In our experiments, we find a *semantic basin* issue: the reverse sampling may repeatedly generate semantically adjacent tokens, producing locally plausible but low-diversity text. This happens because the semantic likelihood term and the model's rollout-induced bias can reinforce sampling within a same semantic cluster. To solve this issue, we propose SemDLM+. First, we add a global transition on top of the semantic transition kernel to prevent the sampling being trapped in some semantic neighborhoods. Second, we introduce a semantic-frequency penalty mechanism during sampling that counteracts the rollout-induced tendency to overproduce tokens from the same semantic basin. Together, these two mechanisms turn SemDLM into a powerful variants as DLMs: easier to train than fully uniform diffusion, but more repairable during sampling than purely masking-based diffusion.

In summary, our contributions are threefold. 1) We provide a principled error analysis for DLMs that explains how transition-kernel design affects approximation difficulty, sampling dynamics, and finite-resource optimization. 2) Guided by this analysis, we develop an improved SemDLM+, which augments SemDLM with a global transition for sampling repair and semantic-frequency penalty to avoid semantic-basin collapse. These designs make SemDLM practically successful and preserve its properties of efficient training and reliable sampling. 3) Experiments on LM1B and OpenWebText show that SemDLM+ improves training dynamics and achieves strong language modeling and generation performance, highlighting SemDLM+ as a promising direction for DLM kernel design.

2 Preliminaries

We denote by q_0 the data distribution over support $\mathcal{X}$, and by q_1 a reference distribution that is easy to sample (e.g., the absorbing or uniform distribution). In language modeling, $\mathcal{X}$ represents the space of length-L sequences where $x = \left(x^{(1)}, \ldots, x^{(L)}\right) \in \mathcal{X} := \mathcal{V}^L$ with $\mathcal{V}$ being a vocabulary of size $|\mathcal{V}| = V$. Diffusion models aim to construct a probability path $q_t, 0 \le t \le 1$ such that one can sample from q_1 and transform it through the learned reverse process to get samples that approximately follow q_0.

Diffusion as Continuous-time Markov Chains (CTMC). Following Lou et al. [22], we view the forward noising process to construct the probability path as a CTMC with *infinitesimal generator* Q_t, i.e., $\frac{dq_t}{dt} = Q_t q_t, 0 \le t \le 1$. One can simulate the forward CTMC via:

$$\text{Forward Process via Euler Sampling: } q\left(x_{t+dt} = y \mid x_t = z\right) = \delta_{zy} + Q_t(z, y)dt + O\left(dt^2\right) \tag{1}$$

Diffusion deploys a parameterized model to mimic the reverse process, $p_\theta(x_{t-dt} \mid x_t) \approx q(x_{t-dt} \mid x_t)$, such that one can iteratively sample a trajectory from the reference distribution to the data distribution through $p_\theta(x_{t-dt} \mid x_t)$. A predominantly used parameterization in DLMs is x-prediction [24, 42, 7, 20], which builds $p_\theta\left(x_0 \mid x_t\right)$ instead of directly approximating $q(x_{t-dt} \mid x_t)$ as:

$$p_\theta\left(x_{t-dt} \mid x_t\right) := \int_{x_0 \in \mathcal{X}} q\left(x_{t-dt} \mid x_t, x_0\right) p_\theta\left(x_0 \mid x_t\right) dx_0. \tag{2}$$

The posterior is $q\left(x_{t-dt} \mid x_t, x_0\right) \propto q\left(x_t \mid x_{t-dt}\right) q(x_{t-dt} \mid x_0)$. We note that the local transition $q_{t|t-dt}$, the cumulative forward $q_{t|0}$, and the generator Q_t are equivalent representations of the same forward process. As such, we describe the process in terms of $q_{t|0}$ and x-prediction in following.

Training objective. DLMs optimize over a variational upper bound of negative log-likelihood [13], $-\log p_\theta(x_0) \leq \ell_0 + \ell_{\text{prior}} + \sum_t \ell_t$, with $\ell_t = \mathbb{E}_{x_t}[D_{KL}(q(x_{t-dt}|x_t, x_0)\|p_\theta(x_{t-dt}|x_t))]$, $\ell_0 = \mathbb{E}_{q(x_{0:1}|x_0)}[-\log p_\theta(x_{0:1})]$ and $\ell_{\text{prior}} = D_{KL}(q(x_1|x_0)\|p_\theta(x_1))$. Taking $dt \to 0$ will make the first two terms negligible. With Eq. (2), we can derive $D_{\mathrm{KL}}\left(q\left(x_{t-dt} \mid x_t\right) \| p_\theta\left(x_{t-dt} \mid x_t\right)\right) \leq D_{\mathrm{KL}}\left(q\left(x_0 \mid x_t\right) \| p_\theta\left(x_0 \mid x_t\right)\right)$ [18], which then yields the common training objective for DLMs:

$$\mathcal{L}(\theta) = \mathbb{E}_{t, x_0, x_t \sim q(x_t|x_0)} D_{\mathrm{KL}}\left(q\left(x_0 \mid x_t\right) \| p_\theta\left(x_0 \mid x_t\right)\right). \tag{3}$$

Sampling Objective. After learning $p_\theta(x_0 \mid x_t)$, the synthesized data points x_0 are generated by iteratively sampling from the induced reverse kernel $x_{t-dt} \sim p_\theta(x_{t-dt} \mid x_t)$ from $t = 1$ to 0. The quality of synthesized samples is measured by the generation risk:

$$\mathcal{R}(\theta) = D_{\mathrm{KL}}\left(q\left(x_0\right) \| p_\theta\left(x_0\right)\right). \tag{4}$$

Transition Kernel Design. With the CTMC framework, we can interpret DLM variants via their transition kernel design. In this paper, we mainly consider the following variants of $q(x_t \mid x_0)$.

Absorbing (Masking) Transition defines a special token $[\texttt{MASK}]$ as the absorbing state such that,

$$q(x_t = j \mid x_0 = i) = \alpha_t \delta_{ij} + (1 - \alpha_t)\delta_{j,[\texttt{MASK}]}, \tag{5}$$

where α_t is the decay factor and δ_{ij} is the Kronecker delta.

Uniform transition. Uniform diffusion spreads the corrupted mass over the full vocabulary:

$$q(x_t = j \mid x_0 = i) = \alpha_t \delta_{ij} + V^{-1}(1 - \alpha_t), \qquad i, j \in \mathcal{V}. \tag{6}$$

Semantic Transition is first introduced in Austin et al. [3], which has its kernel defined as,

$$q(x_t = j \mid x_0 = i) = \alpha_t \delta_{ij} + (1 - \alpha_t) s_t^{\text{sem}}(j \mid i), \tag{7}$$

where the semantic kernel was initially designed as transitions over semantic clusters, such that $s_t^{\text{sem}}(j \mid i) = (k_t)^{-1}\mathbb{I}\big(j \in \mathcal{N}_{k_t}(i)\big)$ where $\mathcal{N}_{k_t}(i)$ is the top-k_t semantic neighborhood of token i. Unfortunately, this design suffers from strong training-sampling mismatch [25] that leads to significant performance degeneration. So we can lightly modify the kernel to make sure the reference distribution matches in forward and reverse process. This gives the transition as

$$s_t^{\text{sem}}(j \mid i) = \exp(\tau_t^{-1}\text{sim}(i, j))/\sum_{k \in \mathcal{V}} \exp(\tau_t^{-1}\text{sim}(i, k)), \tag{8}$$

where $\text{sim}(i, j)$ is a similarity score between embeddings for word i and j; $\tau_t > 0$ is a temperature parameter scheduled to monotonically increase from $\tau_t \to 0, t \to 0$ and $\tau_t \to +\infty, t \to 1$.

3 Principled Error Analysis in DLMs through Bias-Variance Trade-off

In this section, we first build a principled error analysis for DLMs (Sec. 3.1), where we identify several sources that cause the generation error. Then, we provide an interpretation of the transition kernel designs' impact on algorithm behaviors in sec. 3.2.

3.1 Error Analysis and Bias-variance Trade-off in DLMs

We can decompose the generation error in eq. (4) as (detailed proof in sec. B.1):

$$\begin{aligned} &D_{\mathrm{KL}}\big(q(x_0)\|p_\theta(x_0)\big) = \mathbb{E}_t\Big[D_{\mathrm{KL}}\big(q(x_t)\|p_\theta(x_t)\big)\Big] + \\ &\mathbb{E}_{t, x_t \sim q_t}\left[D_{\mathrm{KL}}\big(q(x_0 \mid x_t)\|p_\theta(x_0 \mid x_t)\big)\right] - \mathbb{E}_{t, x_0 \sim q}\left[D_{\mathrm{KL}}\big(q(x_t \mid x_0)\|p_\theta(x_t \mid x_0)\big)\right]. \end{aligned} \tag{9}$$

Effectively, 1) **Approximation Error.** The second term is the approximation error under the true marginals $\{q_t\}$, which is our training objective in Eq. (3). 2) **Sampling Error.** The first term is a time-averaged marginal mismatch between the model's roll-out $\{p_t\}$ and the true marginals $\{q_t\}$. 3) **The forward path mismatch.** The last term is a correction term measuring the mismatch between the model's forward conditionals $p(x_t \mid x_0)$ and the true forward process $q(x_t \mid x_0)$.

Proposition 1 (Error Decomposition). *It is common to assume that $p(x_t \mid x_0) \equiv q(x_t \mid x_0)$ for all (x_0, t) as the forward paths are manually designed. Then the generation error becomes,*

$$D_{\mathrm{KL}}\big(q(x_0)\|p(x_0)\big) = \underbrace{\mathbb{E}_t\Big[D_{\mathrm{KL}}\big(q_t\|p_t\big)\Big]}_{\textit{Sampling Error}} + \underbrace{\mathbb{E}_{t,x_t}\Big[D_{\mathrm{KL}}\big(q(x_0 \mid x_t)\|p(x_0 \mid x_t)\big)\Big]}_{\textit{Approximation Error}}. \tag{10}$$

Bias-Variance Decomposition. In general ML, the generalization error, a measurement that gives the prediction ability of an ML algorithm, can be decomposed into 3 meaningful terms[1]:

Generalization Error = Bias + Variance + Irreducible Risk

Bias is the error between the model's expected prediction and the ground truth, primarily stemming from limitations in the algorithm design or hypothesis space. *Variance* quantifies the spread of the estimated model parameters around their expected value and its impact on inference, typically arising from sensitivity to finite data sampling and the stochasticity of the optimization process.

We can utilize a similar framework to respectively decompose the approximation and sampling error from Eq. (10) (details in Appendix B.3).

Proposition 2 (Bias-Variance Trade-off). *Let $\bar{p}(x_0 \mid x_t) := \mathbb{E}_S\big[p_{\hat{\theta}_S}(x_0 \mid x_t)\big] = \mathbb{E}_S\big[\hat{p}_S(x_0 \mid x_t)\big]$ denote the expected predictive distribution, where $\hat{\theta}(S)$ is the parameter induced by a specific training set and optimization randomness S (short as $\hat{p}_S := p_{\hat{\theta}_S}$). The approximation error in Eq.* (10) *can be decomposed as,*

$$\mathbb{E}_{t,x_t}\mathbb{E}_S\Big[D_{\mathrm{KL}}(q(x_0 \mid x_t)\|\hat{p}_S(x_0 \mid x_t))\Big] = \underbrace{\mathbb{E}_{t,x_t}\Big[D_{\mathrm{KL}}\big(q(x_0 \mid x_t)\|\bar{p}(x_0 \mid x_t)\big)\Big]}_{\textit{Asymptotic Bias } \mathcal{B}_{asym}} + \underbrace{\mathbb{E}_{t,x_t}\big[\mathcal{V}(x_t)\big]}_{\textit{Variance } \mathcal{V}}, \tag{11}$$

where x_t is constructed over the forward process, $q(x_t) = \int_{x_0} q(x_t \mid x_0)p(x_0)dx_0$. The posterior approximation variance term is $\mathcal{V}(x_t) := \mathbb{E}_{q(x_0|x_t)}\Big[\log \bar{p}(x_0 \mid x_t) - \mathbb{E}_S \log(\hat{p}_S(x_0 \mid x_t))\Big] \geq 0$. And the step-wise sampling error in Eq. (10) *can be written as,*

$$D_{\mathrm{KL}}\big(q_t(x_t)\|p_t(x_t)\big) = \underbrace{D_{\mathrm{KL}}\big(q_t(x_t)\|\bar{p}_t(x_t)\big)}_{\textit{Exposure Bias } \mathcal{B}_t} + \underbrace{\mathbb{E}_{x_t}[\log \bar{p}(x_t) - \mathbb{E}_S(\log \hat{p}_S(x_t))]}_{\textit{Sampling Roll-out Variance } \mathcal{V}_t} \tag{12}$$

Sec. B.6.1 shows that sampling variance in Eq. (12) can be controlled by approximation variance. We therefore unify posterior and roll-out variances in the following analysis.

3.2 How Transition Kernel Design affects Generation Error?

From Sec. 3.1, we effectively identified three important sources of error: asymptotic bias in training, exposure bias in sampling, and the variance. In this section, we will illustrate how different transition designs introduced in Sec. 2 affect the model performance of DLMs through these terms. We first provide an overview of the three sources.

Asymptotic Bias. $\mathcal{B}_{\text{asym}}$ reflects the intrinsic difficulty of approximating the true denoising posterior $q(x_0 \mid x_t)$ with a parameterized predictor p_θ even with infinite training data and optimal convergence. It comes from *Model Misspecification* and *Capacity Bottlenecks*, especially when the architecture lacks the expressivity to represent the complexity of the true noise-corrupted posterior.
Exposure Bias. $\mathcal{B}_{\text{exp}} := \sum_{t=0}^{T} \mathcal{B}_t$. While sampling, the model generates the sequence iteratively from $t = 1$ to $t = 0$. The input x_t is not drawn from the true marginal q_t, but from the model's own previous generative distribution $\bar{p}_t$. If the distribution $\bar{p}_t$ deviates even slightly from q_t, then subsequent reverse steps are evaluated on shifted inputs. This phenomenon is known as Exposure Bias, and it quantifies the error coming from *Sampling Dynamics* and *Error Accumulation*.
Variance. $\mathcal{V}$ measures the instability of the learned predictor under finite training resources, which captures the error caused by *Finite Data*, *Optimization Stochasticity*, and *Resource-limited Training*. Different transition kernels $q(x_t \mid x_0)$ can induce different levels of variance.

[1] Irreducible risks are usually introduced by the noise, so we omit them in the following analysis.

3.2.1 Asymptotic Bias Analysis through Posterior Geometry.

To describe the posterior approximation difficulty, we can characterize the *local geometry* of the target posterior. Let $f_\theta(x_t)^i \in \mathbb{R}^{|V|}$ denote the predicted logits at position $i \in [1:L]$ such that $p_\theta^i(\cdot \mid x_t) = \mathrm{softmax}\big(f_\theta(x_t)^i\big)$. Assuming conditional factorization across positions given x_t, the asymptotic bias can be decomposed into local losses:

$$\mathcal{B}_{\text{asym}} = \mathbb{E}_{t,x_t}\left[\sum_{i=1}^{L} \ell\left(f_\theta(x_t)^i; q(x_0^i \mid x_t)\right)\right], \quad \ell(f;q) \coloneqq D_{\mathrm{KL}}\left(q(x_0^i \mid x_t) \,\|\, \mathrm{softmax}(f)\right), \tag{13}$$

Applying the chain rule, the parameter gradient can be written as,

$$\nabla_\theta \mathcal{L} = \mathbb{E}_{t,x_t}\left[\sum_{i=1}^{n}\sum_{k\in V}\left(p_\theta(x_0^i = k \mid x_t) - q(x_0^i = k \mid x_t)\right)\cdot \nabla_\theta f_\theta(x_t)_k^i\right]. \tag{14}$$

To formalize the local approximation difficulty, let $f^\star(q)$ denote optimal logit vector satisfying $\mathrm{softmax}\big(f^\star(q)\big) = q$. We apply a local Taylor expansion which yields

$$\nabla_f \ell(f;q) = \Sigma(q)(f - f^*(q)) + o\left(\|f - f^*(q)\|\right), \text{with } \Sigma(q) \coloneqq \nabla_f^2 \ell(f;q) = \mathrm{Diag}(q) - qq^\top. \tag{15}$$

where $\Sigma(q)$ is the softmax Hessian. Thus, the posterior approximation difficulty is fully determined by the following properties of $\Sigma(q)$.

Posterior approximation difficulty. For a target posterior $q \coloneqq q(x_0 \mid x_t)$, we quantify its local approximation difficulty via the following metrics of $\Sigma(q)$:

$$\begin{aligned} &\text{Logits Active Direction: } d_{\mathrm{act}}(q) \coloneqq \mathrm{rank}\big(\Sigma(q)\big) = |\mathrm{supp}(q)| - 1, \\ &\text{Logits Error Sensitivity: } \mathcal{I}_1(q) \coloneqq \mathrm{tr}\big(\Sigma(q)\big) = 1 - \|q\|_2^2, \quad \mathcal{I}_2(q) \coloneqq \mathrm{tr}\big(\Sigma(q)^2\big). \end{aligned} \tag{16}$$

d_{act} measures the number of independent logit directions that must be simultaneously fitted. $\mathcal{I}_1$ quantifies how local logit mismatch is converted into error, and $\mathcal{I}_2$ quantifies gradient energy. Lower values mean easier approximation. We provide more explanations in sec. B.4.

Based on these metrics, we can now study how the shape of the posterior $q(x_0 \mid x_t)$ affects the difficulty of fitting. By Bayes' rule, $q(x_0 \mid x_t) \propto p_{\mathrm{data}}(x_0)\, q(x_t \mid x_0)$. Therefore, the geometry of the posterior is determined jointly by $p_{\mathrm{data}}(x_0)$ and the forward kernel $q(x_t \mid x_0)$. Substituting the kernel for masking (Eq. (5)), uniform (Eq. (6)), and semantic diffusion (Eq. (7)) obtains:

Proposition 3 (Approximation difficulty across diffusion variants.)**.** *We derive in Sec. B.4 that,*

$$\mathbb{E}\left[d_{\mathrm{act}}^{\mathrm{mask}}\right] \le \mathbb{E}\left[d_{\mathrm{act}}^{\mathrm{sem}}\right]\mathbb{E} \le \left[d_{\mathrm{act}}^{\mathrm{uni}}\right], \quad \mathbb{E}\left[\mathcal{I}_1^{\mathrm{mask}}\right] \le \mathbb{E}\left[\mathcal{I}_1^{\mathrm{sem}}\right] \le \mathbb{E}\left[\mathcal{I}_1^{\mathrm{uni}}\right] \tag{17}$$

where superscripts suggest diffusion variants. This suggests two complementary regimes. When compute or model capacity is limited, uniform diffusion is harder to optimize because it forces the model to fit dense posteriors at nearly every position (large d_{act}), and any local mismatch would lead to large generalization error (large $\mathcal{I}_1$). Masking diffusion and semantic diffusion are easier to optimize as they have a more restricted posterior space and activate fewer logit directions. In the data-sufficient and compute-abundant regime, however, these additional active directions can become useful supervision: masking diffusion receives little learning signal from visible tokens, whereas uniform diffusion keeps almost every position informative for training. This partially explains why uniform diffusion scales better but underperforms other variants in most scenarios.

3.2.2 The Impact of Generators on Exposure Bias Propagation

Exposure Bias. $\mathcal{B}_{\mathrm{exp}} \coloneqq \sum_{t=0}^{T} \mathcal{B}_t$ measures the mismatch accumulated along the generation trajectory. Unlike the asymptotic bias in the approximation stage, exposure bias propagates through the reverse generator itself. We give the following proposition and leave the derivation in Sec. B.5.

Proposition 4 (Exposure Bias Propagation)**.** *With mild conditions, the propagation satisfies:*

$$\mathcal{B}_{t-dt} \le \eta_t \mathcal{B}_t + \rho_t, \tag{18}$$

where ρ_t upper-bounds the step-wise error: $\rho_t \ge \sup_{x_t} D_{\mathrm{KL}}\big(q_t(x_{t-dt} \mid x_t) \| p_t^\theta(x_{t-dt} \mid x_t)\big)$ and $\eta_t^{\mathrm{KL}} \in [0,1]$ is the error propagation coefficient for forwarding q_t to q_{t-dt}. In particular,

$$\eta_{t,\mathrm{mask}} \approx 1, \quad \eta_{t,\mathrm{uni}} \le 1 - \lambda_t^{\mathrm{uni}} < 1, \quad \eta_{t,\mathrm{sem}} \le 1 - \lambda_t^{\mathrm{sem}} < 1.$$

*Consequently, for **semantic and uniform diffusion**, $\rho_t \le \rho$ and $\eta_t \le \eta < 1$, then $\mathcal{B}_{\exp} = \sum_t \mathcal{B}_t = \mathcal{O}\Big(\frac{T\rho}{1-\eta}\Big)$; whereas for **masking diffusion** $\eta_t \approx 1$ and $\mathcal{B}_{\exp} = \mathcal{O}\big(T^2\rho\big)$ with sampling step T.*

Implication. The coefficient η_t measures how strongly sampling errors propagate across reverse steps. For masking diffusion, $\eta_t \approx 1$, so early errors will accumulate quadratically w.r.t T and can become "early commitments" that later steps cannot easily repair. Remasking can mitigate this but cannot completely fix the issue. By contrast, uniform diffusion and semantic diffusion have $\eta_t < 1$, so the reverse dynamics can contract accumulated error and provide an intrinsic repair mechanism, which leads to linear error accumulation w.r.t T.

3.2.3 The Impact of Generators on Variance

Then we explain how the transition kernel affects optimization variance. Let $\mathcal{L}_{x_t}(\theta) := -\log p_\theta(x_0 \mid x_t)$ denote the per-sample denoising likelihood and $\mathcal{L}_t(\theta) := \mathbb{E}_{x_t}[\mathcal{L}_{x_t}(\theta)]$ the per-step loss. Let $\theta^\star$ minimize $\mathcal{L}(\theta) = \mathbb{E}_t[\mathcal{L}_t(\theta)]$ so that $\mathbb{E}_t[\nabla_\theta \mathcal{L}_t(\theta^\star)] = 0$. We look into the gradient variance as a surrogate for the total variance (Details in Appendix B.6.2). At $\theta^\star$, the law of total variance yields

$$\mathrm{Var}_{t,x_t}\big(\nabla_\theta \mathcal{L}_{x_t}(\theta^\star)\big) = \underbrace{\mathbb{E}_t\left[\mathrm{Var}_{x_t}\big(\nabla_\theta \mathcal{L}_{x_t}(\theta^\star)\big)\right]}_{\text{within-step noise}} + \underbrace{\mathrm{Var}_t\left(\nabla_\theta \mathcal{L}_t(\theta^\star)\right)}_{\text{between-step heterogeneity}}. \tag{19}$$

The first term is the usual gradient noise within a fixed step t. The second term measures how the optimization tasks vary across diffusion times. This term is shaped by the transition kernel because $\nabla_\theta \mathcal{L}_t(\theta^\star)$ depends on $q_t(x_t \mid x_0)$ through the corrupted input x_t and the induced posterior target $q(x_0 \mid x_t)$. Therefore, kernels with more dispersed or more time-varying corruption patterns can induce larger between-t heterogeneity (such as a uniform). We prove this connection in Appendix B.6.2, and Fig. 4 empirically validates the correlation between gradient variance and the transition kernel dispersion across time t. This suggests that a dispersed kernel will induce training instability.

Take-home Message for Sec. 3. The analysis in this section reveals a fundamental trade-off in the transition kernel design for DLMs. *Masking diffusion* is easy to train because the absorbing kernel induces sparse denoising posteriors, but it lacks intrinsic repair and can suffer from sampling error propagation. *Uniform diffusion* repairs such errors through global transition, but its dense and dispersed posteriors make training harder and noisier. *Semantic diffusion* is therefore a natural middle ground: semantic locality can reduce approximation difficulty, while global transition can preserve repair ability.

4 Semantic Diffusion: Towards Bias-Variance Minimization in DLMs

4.1 Revisiting Semantic Diffusion

The analysis in Sec. 3 suggests that semantic diffusion has the potential to mitigate the approximation difficulty of uniform diffusion, and can avoid error accumulation in sampling. However, prior studies report that semantic diffusion often underperforms alternatives such as uniform or marginal diffusion [3]. Although a concurrent work [43] reports extremely low test perplexity with a semantic kernel, it does not evaluate unconditional generation. In our experiments, when testing the generation ability of semantic diffusion, we consistently observe a semantic basin problem.

Definition 4.1 (Semantic Basin)**.** We define a semantic basin as a local semantic region in which the reverse chain repeatedly samples semantically adjacent tokens, producing locally plausible but low-diversity text. An example is shown in Table 1.

Why does semantic basin occur? Consider a token position i, and let $\hat{x}_t$ denote the current state during sampling. Ideally, the denoiser should approximate the clean posterior $q(x_0^i = k \mid \hat{x}_t)$. By

Table 1: An example of text generation that suffers from semantic basin.

Model	Generated Text
Semantic basin	[CLS] Because bone was being made up, bone bone bone completely destroyed and dies in it. dies in it. dna … dies in it … the bone cell was not so " dying " " the bone cell " dies the bone marrow decays.
Ours	[CLS] the main aim is to increase the number of people who use the internet and reduce the amount of competition.

Bayes' rule, the posterior logit can be decomposed as

$$l_i^\star(k;\hat{x}_t) = \underbrace{\log p_{\text{data}}(x_0^i = k \mid \hat{x}_t^{-i})}_{\text{contextual prior}} + \underbrace{\log q_t(\hat{x}_t^i \mid x_0^i = k, \hat{x}_t^{-i})}_{\text{local forward likelihood}} + \text{const.} \tag{20}$$

During sampling, however, $\hat{x}_t$ is generated by the model itself rather than drawn from the true forward marginal. Therefore, the denoiser is evaluated on rollout states that may already contain accumulated errors. We write the actual sampling logit as

$$l_{\theta,i}(k;\hat{x}_t) = l_i^\star(k;\hat{x}_t) + \Delta_{\text{roll},i}(k;\hat{x}_t) + \epsilon_i(k) \tag{21}$$

where $\Delta_{\text{roll},i}$ denotes the logit bias induced by the current sampling trajectory. Let $n_i^{(W)}(l)$ be the count of token l in a recent window W, and let $A(k,l)$ denote the contextual contribution of token l to the logit of token k. In sec. D.1, we show that the rollout bias can be locally approximated as $\Delta_{\text{roll},i}(k;\hat{x}_t) \approx \sum_{l\in V} A(k,l) n_i^{(W)}(l)$. If $A(k,l) > 0$ for semantically related tokens $k, l \in C$, then over-producing tokens from cluster C increases the logits of other tokens in the same cluster:

$$n_i^{(W)}(l \in C) \uparrow \Rightarrow \Delta_{\text{roll},i}(k) \uparrow \text{ for } k \in C \Rightarrow p_\theta(x_0^i \in C \mid \hat{x}_t) \uparrow .$$

Meanwhile, in a semantic diffusion kernel, the local likelihood $q_t(j \mid k)$ is large when k is semantically close to j. Thus, the likelihood term in Eq. (20) also favors tokens within the same semantic cluster. These two effects reinforce each other and create a positive feedback loop, which drives the reverse chain into a semantic basin. We provide a formal analysis in sec. D.1.

4.2 From SemDLM to SemDLM+: Mitigating Semantic Basins.

The above analysis suggests that semantic basins can be mitigated by weakening this positive feedback. We use two complementary mechanisms to construct a negative feedback:

(a) Global jumping in the forward kernel. Instead of using a purely local semantic transition, we add a global jumping component:

$$q_t(j \mid k, c) = \alpha_t \delta_{kj} + \beta_t \nu_t(j) + (1 - \alpha_t - \beta_t) s_t^{\text{sem}}(j \mid k, c). \tag{22}$$

The global transition $\beta_t \nu_t(j)$ provides a mixing channel, preventing the model from absorbing into the local semantic cluster. The semantic kernel follows the KNN version as in Eq. (7), with $s_t^{\text{sem}}(j \mid i) = (k_t)^{-1} \mathbb{I}\big(j \in \mathcal{N}_{k_t}(i)\big)$ where the number of semantic neighborhoods k_t increases over time. Specifically, we set $\alpha_t = 1 - t$, $\beta_t = t^2$ and $k_t = 1 + (k_{\max} - 1)\, t^\gamma$ with a hyperparameter γ.

(b) Semantic-frequency Penalty. We further counteract rollout-induced positive feedback during sampling. Ideally, one would subtract the positive feedback term $\sum_{l\in V} [A(k,l)]_+ n_i^{(W)}(l)$. However, $A(k,l)$ is model- and context-dependent and is generally unavailable during sampling. We therefore use semantic similarity scores $S_+(k,l) \geq 0$ as a practical surrogate, and define $m_i^{(W)}(k) = \sum_{l\in V} S_+(k,l) n_i^{(W)}(l)$ as a semantic-frequency regularizer in sampling.

Taken together with a frequency penalty, we then apply a correction on the logits:

$$\tilde{l}_{\theta,i}(k) = l_{\theta,i}(k) - \lambda_{\text{freq}} \log\Big(1 + n_i^{(W)}(k)\Big) - \lambda_{\text{sem}} \log\Big(1 + m_i^{(W)}(k)\Big). \tag{23}$$

The first penalty suppresses exact token over-production, while the second suppresses over-production of the semantic neighborhood of k. This correction introduces a negative feedback mechanism that counteracts the rollout-induced positive feedback which leads semantic basins. Finally, we show in sec. B that SemDLM+ preserves the desirable properties of semantic diffusion: it reduces posterior approximation difficulty while retaining sampling-side repair ability.

5 Experiments

We now evaluate SemDLM+'s language modeling and text generation. We first outline the setup in sec. 5.1 and compare model-size/data-budget scaling and generation quality in sec. 5.2. We then examine component effects and semantic feedback in sec. 5.3.

5.1 Experiment settings

Training Setup. Following Arriola et al. [1], we train two variants of SemDLM+ on the One Billion Words dataset (LM1B [6]) and OpenWebText (OWT [11]) at various model scales. Models trained on both datasets use the *bert-base-uncased* tokenizer. For both datasets, we set up the maximum training steps to be 200K global steps. We fixed the context length to be 128 for LM1B and 1024 for OWT. For the results that we reproduced, we utilize a change-aware loss, which we found helpful in stabilizing training. We refer to Appendix E.1 for a detailed model setup and algorithm implementation details.

Generation Setup. We follow semi-autoregressive sampler. Each stride is initialized from the global transition prior, and each denoising step applies Continuous-time Markov Chain sampling (CTMC) sampler with tau-leaping and predictor-corrector [4]. We vary the sampling steps ranging $\{128, 256, 512\}$ for LM1B and $\{128, 256, 512, 1024\}$ for OWT, and set the number of strides as 2.

Evaluation Setup. Given the issue of semantic basin, perplexity (PPL) and entropy together serve as the evaluation metrics for language modelling and generation ability. We draw the generation frontier and calculate the area-under-the-curve (AUC) as the metric. Testing PPL is measured on texts produced by the model-native sampler, and Generation PPL is measured by GPT-2-Large. x'

5.2 Main Results

Scaling Law via Likelihood Evaluation. We compare the scaling laws of MDLM, Uniform DLMs, and SemDLM+ on OWT at four scales, $N \in \{0.1, 0.3, 0.6, 1.0\}$ B and $D/N \in \{2, 5, 10, 20\}$, where N counts non-embedding parameters and D is training tokens. We utilize the Kaplan form, where we fit the parameter and data coefficients α_N and α_D towards NLL, with $\mathcal{L}(N, D) = \mathcal{L}_\infty + \left[\left(\frac{N_c}{N}\right)^{\alpha_N/\alpha_D} + \frac{D_c}{D}\right]^{\alpha_D}$. Sec. 5.2 reports the fitted coefficient over three training seeds per combination. Prop. 3 motivates the expected orderings parameter bound of $\alpha_N^{\text{MDLM}} > \alpha_N^{\text{SemDLM}} > \alpha_N^{\text{Uniform}}$ and the data bound of $\alpha_D^{\text{MDLM}} > \alpha_D^{\text{SemDLM}} > \alpha_D^{\text{Uniform}}$. The fitted parameter exponents instead give $\alpha_N^{\text{MDLM}} > \alpha_N^{\text{Uniform}} \approx \alpha_N^{\text{SemDLM}}$, while the fitted data exponents give $\alpha_D^{\text{SemDLM}} \geq \alpha_D^{\text{Uniform}} > \alpha_D^{\text{MDLM}}$. This suggests a better scaling law for SemDLM.

Method	L_∞	α_N	α_D
MDLM	2.9478	0.7528	0.5337
Uniform	3.9891	0.4525	0.7220
SemDLM+	4.0223	0.4201	0.8541

Table 2: Scaling-law coefficients.

Generation Quality Evaluation. Since semantic basin can produce deceptively low PPL with low diversity, we compare generation frontiers and AUC in Fig. 1. We record a common range of entropy $[4.892, 5.170]$ by sweeping sampling temperature. The AUC of the Gen-PPL curve is 8.95 for SemDLM+, 10.14 for MDLM, and 17.95 for Uniform (lower is better). This supports a favorable overall trade-off for SemDLM+, although MDLM is slightly better near the lower boundary.

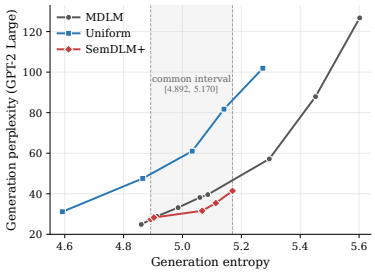


Figure 1: Gen-PPL - entropy frontier curves.

5.3 Behavior Analysis

The Impacts of Generator Dispersion. In our theoretical results in Sec. 3.2.1 and 3.2.3, we show that reducing the dispersion of generator q_t can effectively reduce approximation difficulty and stabilize training. In SemDLM+, this is reflected in the choice of γ. Large γ induces a more complex optimization landscape and lower dispersion. Thus, we systematically analyze the impact of γ in Fig. 2 and 3 and validate this hypothesis. With a larger γ, the training convergence significantly improves. However, with γ overly enlarged, the generation quality degrades, as the transition is constrained in the semantic basin without any escape, thus becoming harmful for the sampling.

Table 3: Full 2^3 ablation on 0.1B OWT models. G: global jumping; S/F: semantic-/exact-frequency penalties. Collapse denotes severe low diversity; ± gives reported run means and sample SD.

G	S	F	Gen-PPL ↓	$H \geq 4.8$	Mean entropy	Semantic repetition ↓
off	off	off	Collapse	1/75	1.96	–
off	on	off	Collapse	1/75	2.73	–
off	off	on	104.73 ± 14.81	52/75	4.79	–
off	on	on	129.88 ± 20.37	54/75	4.86	–
on	off	off	Collapse	0/75	4.13	0.348 ± 0.009
on	on	off	Collapse	1/75	4.18	0.227 ± 0.005
on	off	on	28.34 ± 1.39	57/75	4.90	0.142 ± 0.006
on	on	on	34.48 ± 1.09	58/75	4.93	0.123 ± 0.006

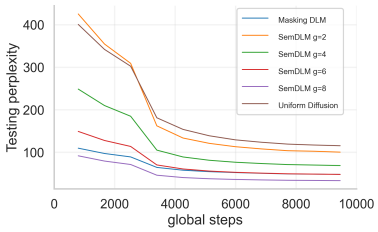


Figure 2: Testing PPL with varying γ during training in LM1B.

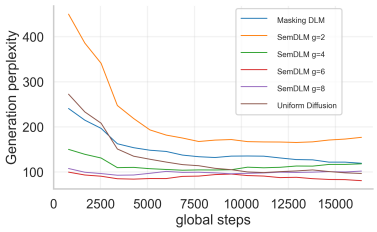


Figure 3: Generation PPL with varying γ during training in LM1B.

Model	PPL ↓
SemDLM (KNN) [2]	149.50
Uniform DLM [2]	88.03
D3PM (Masking) [2]	82.34
Masking DLM [29]	31.78
SEDD (uniform) [22]	40.68
SEDD (Masking) [22]	32.68
GIDD [36]	34.34
SemDLM+ (Ours)	**27.19**

Table 4: Test PPL comparison in LM1B.

Table 5: Generation perplexities (PPL↓) of models trained on LM1B and OWT. The generation length for LM1B is 128 and OWT is 1024. We believe scaling sampling steps infinitely is problematic thus constrained the sampling steps to {128, 256, 512} for LM1B and {128, 256, 512, 1024} for OWT. We conducted 3 tests and results are upper bound, and those with † are reproduced by us and with ‡ are taken from the respective paper.

Model/ Sampling Steps	LM1B			Openwebtext			
	128	256	512	128	256	512	1024
D3PM (Uniform) [2]	/	150.37	133.92	407.72	146.89	112.86	96.46
MDLM (no remasking) [29]	93.69†	85.24†	78.02†	76.50†	64.20†	<u>50.70</u>†	43.00†
ReMDM (MDLM+remask) [38]	91.19†	85.21†	75.98†	<u>72.61</u>†	**62.43**†	50.75†	<u>41.25</u>†
SEDD (uniform) [22]	117.33†	104.51†	106.06†	153.01	150.21	163.97	52.00‡
SEDD (Masking) [22]	<u>74.48</u>†	<u>65.63</u>†	<u>70.74</u>†	80.51†	69.93†	68.97†	46.8‡
GIDD ($a = 1, b = -2$) [36]	85.09†	92.44†	89.76†	345.53†	245.71†	106.18†	94.92†
SemDLM+ (Ours)	**74.52**	**54.08**	**36.60**	**64.39**	<u>62.72</u>	**41.91**	**34.37**

Component-wise Effects. Table 3 reports a full 2^3 ablation on 0.1B OWT models, where we carefully evaluate the effects of global jumping (G), semantic frequency penalty (S) and exact frequency penalty (F). All F-off settings collapse or yield at most one sample with $H \geq 4.8$ out of 75. With F enabled, adding G reduces Gen-PPL from 104.73 ± 14.81 to 28.34 ± 1.39, consistent with a non-local repair channel. Adding S to G+F reduces semantic repetition from 0.142 ± 0.006 to 0.123 ± 0.006, while increasing Gen-PPL to 34.48 ± 1.09. Thus the semantic penalty trades PPL for less semantic repetition.

The Impacts of Semantic Frequency Penalty. Fig. 5 and 6 further illustrate the role of the semantic frequency penalties. We can observe an extremely low generation ppl and entropy when $\lambda_{\text{freq}} = 0$, which coincides with the results in [43] but this actually indicates a collapsed low-diversity regime rather than genuinely strong generation. Increasing λ_{freq} raises entropy and prevents exact-token repetition, while λ_{sem} further suppresses repetition within a semantic neighborhood. However, overly strong penalties can over-flatten the clean posterior and worsen PPL. This shows that these penalties trade off likelihood and diversity by counteracting the semantic basin.

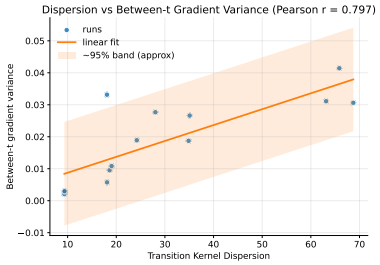


Figure 4: Observed association between transition-kernel dispersion and gradient variance in the measured settings.

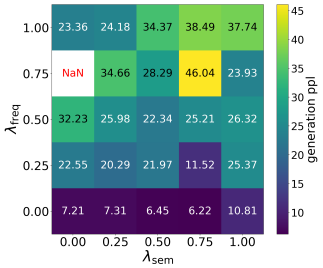


Figure 5: Generation Perplexity with varying λ_{freq} and λ_{sem}. SemDLM+ trained on OWT.

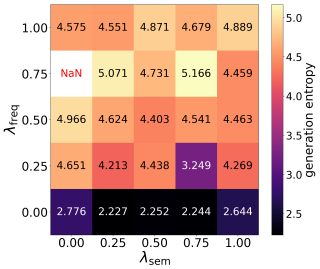


Figure 6: Generation entropy with varying λ_{freq} and λ_{sem}. SemDLM+ trained on OWT.

Sampling Error Accumulation. The step-scaling results in Table 5 and Fig. 7 show how kernels handle accumulated sampling errors. SemDLM+ benefits more clearly from additional reverse steps than MDLM variants: its PPL drops from 74.52 to 36.60 on LM1B and from 64.39 to 34.37 in OWT. This suggests that SemDLM+ successfully preserves the repairing mechanism in uniform DLMs.

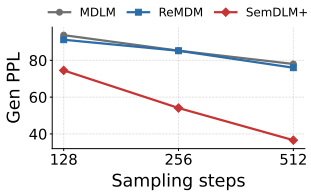


Figure 7: Gen-PPL on LM1B as sampling steps increase.

6 Discussions and Limitations

In this work, we establish an error analysis framework for understanding how transition kernel design affects DLMs and propose SemDLM+, which balances training difficulty and sampling-side repair. We hereby list important limitations for the benefit of the community. 1) Our semantic graph is token-level, so it cannot fully capture context-dependent meanings. 2) The semantic penalty is still an approximation to the ideal transition, which makes generation sensitive to the hyperparameter design. 3) Semantic basin behavior shows that generation PPL alone can be misleading. We believe broader diversity and quality evaluations are needed for the DLM community for a comprehensive analysis.

References

[1] 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.

[2] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *NeurIPS*, pages 17981–17993, 2021.

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

[4] Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *NeurIPS*, 2022.

[5] Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi S. Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *ICML*. OpenReview.net, 2024.

[6] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling, 2014.

[7] Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, and Bowen Zhou. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation, 2025. URL https://arxiv.org/abs/2510.06303.

[8] DeepSeek-AI. Deepseek-v3 technical report. *ArXiv*, abs/2412.19437, 2024. URL https://api.semanticscholar.org/CorpusID:275118643.

[9] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aur'elien Rodriguez, Austen Gregerson, and Ava Spataru et al. The llama 3 herd of models. *ArXiv*, abs/2407.21783, 2024. URL https://api.semanticscholar.org/CorpusID:271571434.

[10] Guhao Feng, Yihan Geng, Jian Guan, Wei Wu, Liwei Wang, and Di He. Theoretical benefit and limitation of diffusion language model. *ArXiv*, abs/2502.09622, 2025. URL https://api.semanticscholar.org/CorpusID:276317202.

[11] Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

[12] Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language models via adaptation from autoregressive models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=j1tSLYKwg8.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[14] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in neural information processing systems*, 34:12454–12465, 2021.

[15] Keyue Jiang, Jiahao Cui, Xiaowen Dong, and Laura Toni. Bures-wasserstein flow matching for graph generation. In *ICML 2025 Generative AI and Biology (GenBio) Workshop*, 2025. URL https://openreview.net/forum?id=qjAkn9mW4C.

[16] 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.

[17] Gen Li and Changxiao Cai. A convergence theory for diffusion language models: An information-theoretic perspective. *arXiv preprint arXiv:2505.21400*, 2025.

[18] Puheng Li, Zhong Li, Huishuai Zhang, and Jiang Bian. On the generalization properties of diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=hCUG1MCFk5.

[19] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343, 2022.

[20] Aiwei Liu, Minghua He, Shaoxun Zeng, Sijun Zhang, Linhao Zhang, Chuhan Wu, Wei Jia, Yuan Liu, Xiao Zhou, and Jie Zhou. Wedlm: Reconciling diffusion language models with standard causal attention for fast inference, 2025. URL https://arxiv.org/abs/2512.22737.

[21] Sulin Liu, Juno Nam, Andrew Campbell, Hannes Stark, Yilun Xu, Tommi Jaakkola, and Rafael Gomez-Bombarelli. Think while you generate: Discrete diffusion with planned denoising. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=MJNywBdSDy.

[22] 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, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=CNicRIVIPA.

[23] Jinjie Ni, Qian Liu, Longxu Dou, Chao Du, Zili Wang, Hang Yan, Tianyu Pang, and Michael Qizhe Shieh. Diffusion language models are super data learners. *CoRR*, abs/2511.03276, 2025.

[24] 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. *arXiv preprint arXiv:2502.09992*, 2025.

[25] Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Önal Ertugrul. Elucidating the exposure bias in diffusion models. In *ICLR*. OpenReview.net, 2024.

[26] Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sMyXP8Tanm.

[27] Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion models with applications to biological sequence generation. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2025. URL https://openreview.net/forum?id=fFuVPKpSt0.

[28] Yiming QIN, Manuel Madeira, Dorina Thanou, and Pascal Frossard. Defog: Discrete flow matching for graph generation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=KPRIwWhqAZ.

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

[30] Subham Sekhar Sahoo, Jean-Marie Lemercier, Zhihan Yang, Justin Deschenaux, Jingyu Liu, John Thickstun, and Ante Jukic. Scaling beyond masked diffusion language models, 2026. URL https://arxiv.org/abs/2602.15014.

[31] Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dallatorre, Bernardo P. de Almeida, Alexander M. Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *ICLR*. OpenReview.net, 2025.

[32] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *NeurIPS*, 2024.

[33] Xiaohang Tang, Keyue Jiang, Che Liu, Qifang Zhao, Xiaoxiao Xu, Sangwoong Yoon, and Ilija Bogunovic. Gdsd: Reinforcement learning as guided denoiser self-distillation for diffusion language models, 2026. URL https://arxiv.org/abs/2605.29398.

[34] Qwen Team. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024. URL https://api.semanticscholar.org/CorpusID:274859421.

[35] Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=rvZv7sDPV9.

[36] Dimitri von Rütte, Janis Fluri, Omead Pooladzandi, Bernhard Schölkopf, Thomas Hofmann, and Antonio Orvieto. Scaling behavior of discrete diffusion language models, 2025. URL https://arxiv.org/abs/2512.10858.

[37] Binxu Wang, Jiaqi Shang, and Haim Sompolinsky. Diverse capability and scaling of diffusion and auto-regressive models when learning abstract rules. *CoRR*, abs/2411.07873, 2024. doi: 10.48550/ARXIV.2411.07873. URL https://doi.org/10.48550/arXiv.2411.07873.

[38] Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2025. URL https://openreview.net/forum?id=xNwZ8kDC7T.

[39] Yunhe Wang, Kai Han, Huiling Zhen, Yuchuan Tian, Hanting Chen, Yongbing Huang, Yufei Cui, Yingte Shu, Shan Gao, Ismail Elezi, Roy Vaughan Miles, Songcen Xu, Feng Wen, Chao Xu, Sinan Zeng, and Dacheng Tao. Top 10 open challenges steering the future of diffusion language model and its variants, 2026. URL https://arxiv.org/abs/2601.14041.

[40] Yuxiang Wang, Yu Xiang, Baojian Zhou, Qifang Zhao, Keyue Jiang, Yanghua Xiao, and Xiaoxiao Xu. On the trainability of masked diffusion language models via blockwise locality, 2026. URL https://arxiv.org/abs/2604.24832.

[41] Zhe Wang, Jiaxin Shi, Nicolas Heess, Arthur Gretton, and Michalis Titsias. Learning-order autoregressive models with application to molecular graph generation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=EY6pXIDi3G.

[42] Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b, 2025. URL https://hkunlp.github.io/blog/2025/dream.

[43] Oussama Zekri, Théo Uscidda, Nicolas Boullé, and Anna Korba. Generalized discrete diffusion from snapshots. *CoRR*, abs/2603.21342, 2026.

[44] Siyue Zhang, Yilun Zhao, Liyuan Geng, Arman Cohan, Anh Tuan Luu, and Chen Zhao. Diffusion vs. autoregressive language models: A text embedding perspective. *CoRR*, abs/2505.15045, 2025. doi: 10.48550/ARXIV.2505.15045. URL https://doi.org/10.48550/arXiv.2505.15045.

A Related Work

Discrete Diffusion Models. Diffusion language models (DLMs) [3, 14] have emerged as a promising alternative to autoregressive (AR) language models. Unlike AR LLMs [9, 34, 8] that decode tokens strictly left-to-right, DLMs generate text by iterative refinement, progressively denoising from completely random noise sampled from reference distribution (e.g., all [MASK] tokens or samples from uniform distribution) into a clean data points [13, 19]. DLMs have achieved competitive language modeling performance [24, 40], attracting interest for their potential to reduce decoding latency via parallel updates without sacrificing quality. Representative formulations include D3PM [3] and its language-model instantiations, masking-based diffusion (MDLM) [29, 32, 26], as well as alternative discrete-time or continuous-time constructions such as SEDD [22], tau-leaping for discrete-state continuous-time diffusion [4], and discrete flow matching [5]. These models vary in their design of the *transition kernel*. Masking DLMs are the predominant archetype, due to stable training and convenient likelihood objectives [29, 32, 26]. Beyond masking, prior work considers uniform diffusion [3, 31], marginal diffusion [28], and semantic-neighborhood diffusion [3] to improve mixing and address exposure to hard corruptions. Early studies argue that masking-style diffusion can alleviate slow convergence, training instability, and weaker generalization observed in some discrete diffusion setups [3, 14, 39]. More recent scaling analyses, however, report that uniform diffusion can enjoy better scaling behavior given sufficient data and training budget [36, 35, 30]. To bridge these regimes, *GIDD* [35] interpolates between masking and uniform noising, offering a controlled mixture family for both training and inference.

Large Diffusion Language Models. At scale, Nie et al. [24] introduces LLaDA, the first 8B scale diffusion Large Language Models (DLLM) with masked diffusion trained *from scratch* with a standard pretraining + supervised fine-tuning (SFT) pipeline. LLaDA demonstrates strong scalability and competitive in-context learning and instruction-following ability at the 8B scale. Beyond purely diffusion training from scratch, hybrid paradigms aim to combine AR coherence with diffusion parallelism: Ye et al. [42] presents Dream 7B and Cheng et al. [7] propose SDAR. They both convert a pretrained AR model into a block-wise diffusion decoder. Finally, Tencent [20] target deployment efficiency by reconciling diffusion decoding with standard KV caching: they propose WeDLM, which realizes diffusion-style parallel decoding under causal attention via reordering and streaming commitment, enabling substantial speedups over optimized AR serving (e.g., vLLM). Overall, these works indicate a rapidly maturing ecosystem of large diffusion LLMs.

Comparing Diffusion and Autoregressive Models. A growing literature compares diffusion and autoregressive (AR) generation across empirical performance, architectural design, and learning/inference dynamics. Empirically, Wang et al. [37] compares AR and diffusion models on downstream tasks and characterizes practical trade-offs. From an architectural viewpoint, Zhang et al. [44] analyzes how causal (AR) versus bidirectional (diffusion-style) attention affects modeling and decoding efficiency. On the learning side, Kim et al. [16] studies diffusion trajectory construction (token update ordering) and shows that it can induce a harder optimization problem than AR's fixed causal ordering; Gong et al. [12] demonstrates that diffusion and AR objectives can be cast in closely related forms, enabling transfer and distillation across paradigms. Recent scaling evidence further suggests that diffusion models may generalize better as data and compute increase [23]. Despite these advances, existing studies do not offer a unified framework that links training dynamics, generator transition dynamics. We serve as the first to provide a unified framework that originated from generalization error analysis, and identify the bias-variance trade-offs in DLLMs

Kernel Design in Diffusion Language Models. Beyond model architectures, another line of work in discrete diffusion improves decoding quality by manipulating the generation path. QIN et al. [28] propose target-guided diffusion to steer generation toward desired outcomes during sampling. Wang et al. [41] learns the token update order with reinforcement learning, adapting the path to the input and improving decoding efficiency. Wang et al. [38] introduces remasking strategies that re-corrupt tokens during inference to correct early mistakes and reduce degeneration. From a geometric perspective, Jiang et al. [15] redesigns the diffusion path via optimal-transport alignment to better match intermediate states with the target distribution. More recently, Peng et al. [27], Liu et al. [21] propose trainable planners that predict which positions to decode or commit at each step, enabling selective refinement and faster convergence. Collectively, these methods highlight the importance of the transition path as a controllable degree of freedom in diffusion-based sequence generation.

Theoretical Guarantees of DLMs. While diffusion language models (DLMs) are often motivated by the prospect of parallel decoding, recent theory makes precise when such acceleration is *provably* attainable and when it is fundamentally limited. Feng et al. [10] provide a rigorous analysis for masked diffusion models (MDMs) and show that the efficiency–accuracy trade-off is highly metric-dependent. In particular, under mild regularity assumptions, they prove that MDMs can reach near-optimal *perplexity* with a number of refinement steps that does not grow with the sequence length, supporting the intuition that parallel refinement can match AR-level likelihood quality with bounded-step sampling. Kim et al. [16] and Tang et al. [33] demonstrate that diffusion LLMs are difficult to train, with training instability, severe training-inference mismatch, and a complex optimization landscape, but are more powerful in sampling once trained. Later, Li and Cai [17] develop an information-theoretic convergence theory for DLM sampling. Their results offers a concrete theoretical lens: DLMs can admit provable acceleration for likelihood-oriented metrics and weakly dependent sequences, yet face intrinsic limitations when the goal is exact-sequence correctness or when token dependencies are strong.

B Omitted Proofs

B.1 Error decomposition

Proposition 5 (Error Decomposition)**.** *The exact KL divergence in the generation process can be decomposed into:*

$$\begin{aligned} &D_{\mathrm{KL}}\big(q(x_0)\|p_\theta(x_0)\big) \\ &= \underbrace{\mathbb{E}_t\Big[D_{\mathrm{KL}}\big(q(x_t)\|p_\theta(x_t)\big)\Big]}_{\textit{Sampling Error}} + \underbrace{\mathbb{E}_{t,x_t\sim q_t}\left[D_{\mathrm{KL}}\big(q(x_0\mid x_t)\|p_\theta(x_0\mid x_t)\big)\right]}_{\textit{Approximation Error}} \\ &\quad - \underbrace{\mathbb{E}_{t,x_0\sim q}\left[D_{\mathrm{KL}}\big(q(x_t\mid x_0)\|p_\theta(x_t\mid x_0)\big)\right]}_{\textit{Forward Process Mismatch}}. \end{aligned} \tag{24}$$

1) ***Approximation Error.*** *The middle term is the* approximation (denoising) error *under the true marginals $\{q_t\}$, which has almost the same form as our training objective in Eq.* (3)*. 2)* ***Sampling Error.*** *The first term is a time-averaged* marginal mismatch *between the model's rolled-out marginals $\{p_t\}$ and the true marginals $\{q_t\}$. 3)* ***The forward path mismatch*** *The last term is a time-averaged correction term depending on whether the model's forward conditionals $p(x_t \mid x_0)$ match the true noising process $q(x_t \mid x_0)$.*

Proof. For simplicity, in the following we omit the parameter θ but just denote p_θ as p. We first consider the KL divergence on the joint distribution of (x_0, x_t), where

$$D_{\mathrm{KL}}\big(q(x_0,x_t)\|p(x_0,x_t)\big) \coloneqq \int_{x_0,x_t} dx_0 dx_t q(x_0,x_t)\log\frac{q(x_0,x_t)}{p(x_0,x_t)}. \tag{25}$$

where x_t and x_0 are sampled through $x_t \sim q(x_t \mid x_0)$, $x_0 \sim q(x_0)$. We use $q(x_0, x_t) = q_t(x_t)q(x_0 \mid x_t)$, and $p(x_0, x_t) = p_t(x_t)p(x_0 \mid x_t)$ to write,

$$\log\frac{q(x_0,x_t)}{p(x_0,x_t)} = \log\frac{q_t(x_t)q(x_0\mid x_t)}{p_t(x_t)p(x_0\mid x_t)} = \log\frac{q_t(x_t)}{p_t(x_t)} + \log\frac{q(x_0\mid x_t)}{p(x_0\mid x_t)}. \tag{26}$$

Substitute Eq. (26) into Eq. (25), we obtain,

$$\begin{aligned} &D_{\mathrm{KL}}\big(q(x_0,x_t)\|p(x_0,x_t)\big) \\ &= \underbrace{\int_{x_0,x_t} q(x_0,x_t)\log\frac{q_t(x_t)}{p_t(x_t)}dx_0dx_t}_{\text{Term (1)}} + \underbrace{\int_{x_0,x_t} q(x_0,x_t)\log\frac{q(x_0\mid x_t)}{p(x_0\mid x_t)}dx_0dx_t}_{\text{Term (2)}}. \end{aligned} \tag{27}$$

Given that $\log \frac{q_t(x_t)}{p_t(x_t)}$ depends only on x_t but not on x_0, term (1) becomes

$$\begin{aligned}\text{Term (1)} &= \int_{x_t} \left(\int_{x_0} q(x_0, x_t) dx_0 \right) \log \frac{q_t(x_t)}{p_t(x_t)} dx_t \\ &= \int_{x_t} q_t(x_t) \log \frac{q_t(x_t)}{p_t(x_t)} dx_t = D_{\mathrm{KL}}\big(q_t(x_t) \| p_t(x_t)\big).\end{aligned} \tag{28}$$

Similarly, using $q(x_0, x_t) = q_t(x_t) q(x_0 \mid x_t)$, Term (2) becomes,

$$\begin{aligned}\text{Term (2)} &= \int_{x_t, x_0} q_t(x_t) q(x_0 \mid x_t) \log \frac{q(x_0 \mid x_t)}{p(x_0 \mid x_t)} dx_t dx_0 \\ &= \int_{x_t} dx_t q_t(x_t) \left[\int_{x_0} q(x_0 \mid x_t) \log \frac{q(x_0 \mid x_t)}{p(x_0 \mid x_t)} dx_0 \right] \\ &= \int_{x_t} q_t(x_t) D_{\mathrm{KL}}\big(q(x_0 \mid x_t) \| p(x_0 \mid x_t)\big) dx_t \\ &= \mathbb{E}_{x_t \sim q_t}\Big[D_{\mathrm{KL}}\big(q(x_0 \mid x_t) \| p(x_0 \mid x_t)\big) \Big].\end{aligned} \tag{29}$$

Plugging Eq. (28) and Eq. (29) into Eq. (27) yields

$$D_{\mathrm{KL}}\big(q(x_0, x_t) \| p(x_0, x_t)\big) = D_{\mathrm{KL}}\big(q_t \| p_t\big) + \mathbb{E}_{x_t \sim q_t}\Big[D_{\mathrm{KL}}\big(q(x_0 \mid x_t) \| p(x_0 \mid x_t)\big) \Big]. \tag{30}$$

Now consider $q(x_0, x_t) = q(x_0) q(x_t \mid x_0)$ and $p(x_0, x_t) = p(x_0) p(x_t \mid x_0)$, we get

$$\log \frac{q(x_0, x_t)}{p(x_0, x_t)} = \log \frac{q(x_0) q(x_t \mid x_0)}{p(x_0) p(x_t \mid x_0)} = \log \frac{q(x_0)}{p(x_0)} + \log \frac{q(x_t \mid x_0)}{p(x_t \mid x_0)}. \tag{31}$$

Substitute Eq. (31) into Eq. (25):

$$\begin{aligned}&D_{\mathrm{KL}}\big(q(x_0, x_t) \| p(x_0, x_t)\big) \\ &= \int_{x_0, x_t} dx_0 dx_t q(x_0, x_t) \left[\log \frac{q(x_0)}{p(x_0)} + \log \frac{q(x_t \mid x_0)}{p(x_t \mid x_0)} \right] \\ &= \underbrace{\int_{x_0, x_t} dx_0 dx_t q(x_0, x_t) \log \frac{q(x_0)}{p(x_0)}}_{\text{Term (3)}} + \underbrace{\int_{x_0, x_t} dx_0 dx_t q(x_0, x_t) \log \frac{q(x_t \mid x_0)}{p(x_t \mid x_0)}}_{\text{Term (4)}}.\end{aligned}$$

Then, using the same marginalization trick as Eq. (28),

$$\begin{aligned}\text{Term (3)} &= \int_{x_0} \left(\int_{x_t} q(x_0, x_t) dx_t \right) \log \frac{q(x_0)}{p(x_0)} dx_0 \\ &= \int_{x_0} q(x_0) \log \frac{q(x_0)}{p(x_0)} dx_0 = D_{\mathrm{KL}}\big(q(x_0) \| p(x_0)\big).\end{aligned} \tag{32}$$

And similarly for Term (4),

$$\begin{aligned}\text{Term (4)} &= \int_{x_0, x_t} dx_t dx_0 \quad q(x_0) q(x_t \mid x_0) \log \frac{q(x_t \mid x_0)}{p(x_t \mid x_0)} \\ &= \int dx_0 q(x_0) \left[\int_{x_t} q(x_t \mid x_0) \log \frac{q(x_t \mid x_0)}{p(x_t \mid x_0)} dx_t \right] \\ &= \mathbb{E}_{x_0 \sim q}\Big[D_{\mathrm{KL}}\big(q(x_t \mid x_0) \| p(x_t \mid x_0)\big) \Big].\end{aligned} \tag{33}$$

Plugging Eq. (32) and Eq. (33) into Eq. (31) yields

$$D_{\mathrm{KL}}\big(q(x_0, x_t) \| p(x_0, x_t)\big) = D_{\mathrm{KL}}\big(q(x_0) \| p(x_0)\big) + \mathbb{E}_{x_0 \sim q}\Big[D_{\mathrm{KL}}\big(q(x_t \mid x_0) \| p(x_t \mid x_0)\big) \Big]. \tag{34}$$

Both Eq. (30) and Eq. (34) equal the same quantity $D_{\mathrm{KL}}(q(x_0, x_t) \| p(x_0, x_t))$, then

$$\begin{aligned}&D_{\mathrm{KL}}\big(q_t(x_t) \| p_t(x_t)\big) + \mathbb{E}_{x_t \sim q(x_t)}\Big[D_{\mathrm{KL}}\big(q(x_0 \mid x_t) \| p(x_0 \mid x_t)\big) \Big] \\ =&D_{\mathrm{KL}}\big(q(x_0) \| p(x_0)\big) + \mathbb{E}_{x_0 \sim q(x_0)}\Big[D_{\mathrm{KL}}\big(q(x_t \mid x_0) \| p(x_t \mid x_0)\big) \Big].\end{aligned}$$

Hence

$$\begin{aligned}&D_{\mathrm{KL}}\big(q(x_0)\|p(x_0)\big)\\ =&D_{\mathrm{KL}}\big(q_t\|p_t\big)+\mathbb{E}_{x_t\sim q_t}\Big[D_{\mathrm{KL}}\big(q(x_0\mid x_t)\|p(x_0\mid x_t)\big)\Big]-\mathbb{E}_{x_0\sim q}\Big[D_{\mathrm{KL}}\big(q(x_t\mid x_0)\|p(x_t\mid x_0)\big)\Big].\end{aligned} \tag{35}$$

Taking expectation of Eq. (35) with respect to t yields,

$$\begin{aligned}D_{\mathrm{KL}}\big(q(x_0)\|p(x_0)\big)=&\mathbb{E}_t\Big[D_{\mathrm{KL}}\big(q_t\|p_t\big)\Big]+\mathbb{E}_{t,x_t\sim q_t}\Big[D_{\mathrm{KL}}\big(q(x_0\mid x_t)\|p(x_0\mid x_t)\big)\Big]\\ &-\mathbb{E}_{t,x_0\sim q}\Big[D_{\mathrm{KL}}\big(q(x_t\mid x_0)\|p(x_t\mid x_0)\big)\Big].\end{aligned} \tag{36}$$

which is exactly the three-term error decomposition that we are interested.

B.2 When the forward process matches

Assume that for all (x_0,t),$p(x_t\mid x_0)\equiv q(x_t\mid x_0)$. Then for every (x_0,t), $D_{\mathrm{KL}}\big(q(x_t\mid x_0)\|p(x_t\mid x_0)\big)=0$. Hence the entire time-averaged correction term in Eq. (36) is zero:

$$\mathbb{E}_{t,x_0\sim q}\Big[D_{\mathrm{KL}}\big(q(x_t\mid x_0)\|p(x_t\mid x_0)\big)\Big]=0. \tag{37}$$

Therefore Eq. (36) reduces to

$$D_{\mathrm{KL}}\big(q(x_0)\|p(x_0)\big)=\underbrace{\mathbb{E}_t\Big[D_{\mathrm{KL}}\big(q_t\|p_t\big)\Big]}_{\text{Sampling Error}}+\underbrace{\mathbb{E}_{t,x_t\sim q_t}\Big[D_{\mathrm{KL}}\big(q(x_0\mid x_t)\|p(x_0\mid x_t)\big)\Big]}_{\text{Approximation Error}}. \tag{38}$$

This equation is importance, as it helps us to consider the error seperately from sampling and training.

B.3 Bias-Variance Decomposition

B.3.1 Training and sampling bias-variance trade-off

The bias-variance decomposition holds for both training and sampling, so we utilize $p(x_0\mid x_t)$ in approximation error as an illustration, which can be easily generalized to sampling distribution $p(x_t)$.

With the expected predictor

$$\bar{p}(x_0\mid x_t):=\mathbb{E}_S\big[p_{\hat{\theta}_S}(x_0\mid x_t)\big]=\mathbb{E}_S\big[\hat{p}_S(x_0\mid x_t)\big]. \tag{39}$$

where $\hat{\theta}(S)$ denote parameter induced by specific training set and optimization randomness S (short as $\hat{p}_S:=p_{\hat{\theta}_S}$). Then Eq. (3) admits the following exact decomposition ($p(x_0\mid x_t)$ simplified as p):

$$\begin{aligned}&\mathbb{E}_S\Big[D_{\mathrm{KL}}(q\|\hat{p}_S)\Big]=D_{\mathrm{KL}}(q\|\bar{p})+\mathcal{V}(x_t),\\ &\text{where, }\mathcal{V}(x_t):=\mathbb{E}_{q(x_0|x_t)}\Big[\log\bar{p}-\mathbb{E}_S\log(\hat{p}_S)\Big]\geq 0.\end{aligned} \tag{40}$$

The nonnegativity of $\mathcal{V}(x_t)$ follows from Jensen's inequality: $\mathbb{E}_S[\log\hat{p}_S(x_0\mid x_t)]\leq\log\mathbb{E}_S[\hat{p}_S(x_0\mid x_t)]=\log\bar{p}(x_0\mid x_t)$. Specifically,

$$\begin{aligned}&\mathbb{E}_S\Big[D_{\mathrm{KL}}\big(q(x_0\mid x_t)\|p_{\hat{\theta}_S}(x_0\mid x_t)\big)\Big]=\mathbb{E}_S\left[\int q(x_0\mid x_t)\log\frac{q(x_0\mid x_t)}{p_{\hat{\theta}_S}(x_0\mid x_t)}dx_0\right]\\ &=\int q(x_0\mid x_t)\log q(x_0\mid x_t)dx_0-\int q(x_0\mid x_t)\underbrace{\mathbb{E}_S\left[\log p_{\hat{\theta}_S}(x_0\mid x_t)\right]}_{\log\bar{p}(x_0)}dx_0\\ &=\int q(x_0\mid x_t)\left(\log q(x_0\mid x_t)-\log\bar{p}(x_0\mid x_t)\right)dx_0\\ &\quad+\int q(x_0\mid x_t)\left(\log\bar{p}(x_0\mid x_t)-\mathbb{E}_S[\log p_{\hat{\theta}_S}(x_0\mid x_t)]\right)dx_0\quad\text{(Add \& Sub)}\\ &=\underbrace{D_{\mathrm{KL}}(q\|\bar{p})}_{\textbf{Bias}}+\underbrace{\mathbb{E}_{q(x_0|x_t)}\left[\log\bar{p}-\mathbb{E}_S\log(\hat{p}_S)\right]}_{\textbf{Variance}}.\end{aligned} \tag{41}$$

Averaging over x_t yields an expected risk decomposition:

$$\mathbb{E}_{t,x_t}\mathbb{E}_S\Big[D_{\mathrm{KL}}(q(x_0 \mid x_t)\|\hat{p}_S(x_0 \mid x_t))\Big] = \underbrace{\mathbb{E}_{t,x_t}\Big[D_{\mathrm{KL}}\big(q(x_0 \mid x_t)\|\bar{p}(x_0 \mid x_t)\big)\Big]}_{\textbf{Asymptotic Bias } \mathcal{B}_{\text{asym}}} + \underbrace{\mathbb{E}_{t,x_t}\big[\mathcal{V}(x_t)\big]}_{\textbf{Variance } \mathcal{V}}, \tag{42}$$

where x_t are distributed according to the forward process, $q(x_t) = \int_{x_0} q(x_t \mid x_0)p(x_0)dx_0$. The sampling bias-variance trade-off follows the exact same derivation, which only requires replacing $p(x_0 \mid x_t)$ with $p(x_t)$, such that

$$D_{\mathrm{KL}}\big(q_t(x_t)\|p_t(x_t)\big) = \underbrace{D_{\mathrm{KL}}\big(q_t(x_t)\|\bar{p}_t(x_t)\big)}_{\textbf{Exposure Bias } \mathcal{B}_t} + \underbrace{\mathbb{E}_{x_t}[\log \bar{p}(x_t) - \mathbb{E}_S(\log \hat{p}_S(x_t))]}_{\textbf{Sampling Roll-out Variance } \mathcal{V}_t} \tag{43}$$

B.4 Derivation for Asymptotic Bias: Local Geometry and Posterior Approximation Difficulty

This section, we prove Proposition 3, which we restate here

Proposition 6 (Approximation difficulty across diffusion variants.)**.** *We derive in Sec. B.4 that,*

$$\mathbb{E}\left[d_{\text{act}}^{\text{mask}}\right] \le \mathbb{E}\left[d_{\text{act}}^{\text{sem}}\right]\mathbb{E} \le \left[d_{\text{act}}^{\text{uni}}\right], \quad \mathbb{E}\left[\mathcal{I}_1^{\text{mask}}\right] \le \mathbb{E}\left[\mathcal{I}_1^{\text{sem}}\right] \le \mathbb{E}\left[\mathcal{I}_1^{\text{uni}}\right] \tag{44}$$

B.4.1 Characterizing the local geometry with Hessian matrix

To describe the posterior approximation difficulty, we characterize the *local geometry* of the target posterior. Let $f_\theta(x_t)^i \in \mathbb{R}^{|V|}$ denote the predicted logits at position $i \in [1:L]$ such that $p_\theta^i(\cdot \mid x_t) = \mathrm{softmax}(f_\theta(x_t)^i)$. Assuming conditional factorization across positions given x_t, the asymptotic bias can be decomposed into local losses:

$$\mathcal{B}_{\text{asym}} = \mathbb{E}_{t,x_t}\left[\sum_{i=1}^{L} \ell_i\left(f_\theta(x_t)^i; x_t\right)\right], \qquad \ell_i(f_\theta; x_t) := D_{\mathrm{KL}}\big(q(x_0^i \mid x_t)\|\mathrm{softmax}(f_\theta)\big). \tag{45}$$

Fix a position i and a corrupted x_t, we have,

$$\ell_i(f; x_t) = D_{\mathrm{KL}}\big(q(x_0^i = \cdot \mid x_t) \,\|\, \mathrm{softmax}(f_\theta)\big) = \sum_{k \in V} q(x_0^i = k \mid x_t) \log \frac{q(x_0^i = k \mid x_t)}{p_\theta(x_0^i = k \mid x_t)}. \tag{46}$$

Since $\sum_k q(x_0^i = k \mid x_t) \log q(x_0^i = k \mid x_t)$ is constant w.r.t θ, minimizing $\ell_i(f; x_t)$ is equivalent to minimizing the cross-entropy

$$-\sum_{k \in V} q(x_0^i = k \mid x_t) \log p_\theta(x_0^i = k \mid x_t).$$

Recall that $\log p_\theta(x_0^i = k \mid x_t) = f_\theta(x_t)_k^i - \log \sum_{m \in V} e^{f_\theta(x_t)_m^i}$, when $f_\theta(x_t)_k^i$ denote the k-th logit at position i (we use f_k when there is no ambuguity), we obtain

$$\frac{\partial \log p_\theta(x_0^i = k \mid x_t)}{\partial f_r} = \delta_{kr} - p_\theta(x_0^i = r \mid x_t).$$

Therefore, the gradient is

$$\begin{aligned}
\frac{\partial \ell_i(f; x_t)}{\partial f_r} &= -\sum_{k \in V} q(x_0^i = k \mid x_t) \frac{\partial \log p_\theta(x_0^i = k \mid x_t)}{\partial f_{\theta,r}} \\
&= -\sum_{k \in V} q(x_0^i = k \mid x_t)(\delta_{kr} - p_\theta(x_0^i = r \mid x_t)) \\
&= -q(x_0^i = r \mid x_t) + p_\theta(x_0^i = r \mid x_t) \underbrace{\sum_{k \in V} q(x_0^i = k \mid x_t)}_{=1} \\
&= p_\theta(x_0^i = r \mid x_t) - q(x_0^i = r \mid x_t),
\end{aligned}$$

By the chain rule and the decomposition of the global loss,

$$\begin{aligned}\nabla_\theta \mathcal{L} &= \nabla_\theta \mathbb{E}_{t,x_t}\left[\sum_{i=1}^{L} \ell_i\big(f_\theta(x_t)^i; x_t\big)\right] \\ &= \mathbb{E}_{t,x_t}\left[\sum_{i=1}^{L}\sum_{k\in V} \frac{\partial \ell_i(f_\theta(x_t)^i; x_t)}{\partial f_\theta(x_t)^i_k} \cdot \nabla_\theta f_\theta(x_t)^i_k\right] \\ &= \mathbb{E}_{t,x_t}\left[\sum_{i=1}^{L}\sum_{k\in V} \Big(p_\theta(x_0^i = k \mid x_t) - q(x_0^i = k \mid x_t)\Big) \cdot \nabla_\theta f_\theta(x_t)^i_k\right], \end{aligned} \tag{47}$$

which is exactly Eq. (14).

Since q is irrelevant with f, differentiate Eq. (46) once more we obtain

$$\nabla_f^2 \ell(f; q) = \frac{\partial p}{\partial f}.$$

For softmax, the Jacobian is

$$\frac{\partial p_k}{\partial f_r} = p_k(\delta_{kr} - p_r),$$

hence

$$\nabla_f^2 \ell(f; q) = \mathrm{Diag}(p) - pp^\top. \tag{48}$$

At any optimum $f^\star(q)$ such that $\mathrm{softmax}(f^\star(q)) = q$, we obtain

$$\nabla_f^2 \ell(f^\star(q); q) = \mathrm{Diag}(q) - qq^\top = \Sigma(q). \tag{49}$$

Local Taylor expansion. By standard second-order expansion around $f^\star(q)$, and denote $\delta f \approx f - f^*(q)$ a small variation around f, using taylor expansion around f^* we have,

$$\ell(f^\star(q) + \delta f; q) = \ell(f^\star(q); q) + \underbrace{\nabla_f \ell(f^\star(q); q)}_{=0} \delta f + \frac{1}{2}\delta f^\top \Sigma(q) \delta f + o(\|\delta f\|^2), \tag{50}$$

which gives

$$\ell(f; q) - \ell(f^\star(q); q) = \frac{1}{2}\delta f^\top \Sigma(q) \delta f + o(\|\delta f\|^2). \tag{51}$$

hence, we obtain Eq. (15), where

$$\nabla_f \ell(f; q) = \Sigma(q)(f - f^*(q)) + o\left(\|f - f^*(q)\|\right), \text{with } \Sigma(q) \coloneqq \nabla_f^2 \ell(f; q) = \mathrm{Diag}(q) - qq^\top. \tag{52}$$

This suggest that the shape of the second order hessian will dominant the optimization dynamics of our diffusion LLMs.

B.4.2 Metrics for posterior approximation difficulty.

Through our Hessian matrix $\Sigma(q)$, we can then define a few metrics that is helpful in analyzing the posterior approximation difficulty in DLMs. Specifically,

$$d_{\mathrm{act}}(q) \coloneqq \mathrm{rank}\big(\Sigma(q)\big), \quad \mathcal{I}_1(q) \coloneqq \mathrm{tr}\big(\Sigma(q)\big), \quad \mathcal{I}_2(q) \coloneqq \mathrm{tr}\big(\Sigma(q)^2\big). \tag{53}$$

Specifically,

1. d_{act} counts the dimension of the tangent space on which the KL objective has non-zero curvature. Hence it is the number of independent logit directions that must be fitted simultaneously.
2. $\mathcal{I}_1$ measures the average curvature and therefore how much an isotropic local logit error increases the KL loss.
3. $\mathcal{I}_2$ measures the squared curvature and therefore the local gradient energy induced by the same logit error.

We provide some justification on what they mean in the optimization problem.

$d_{\mathrm{act}}(q)$ **measures the independent direction to be fitted.** Given that $d_{\mathrm{act}}(q) = \mathrm{rank}(\Sigma(q) = \mathrm{supp}(q)$. It directly measure the independent directions that is required to be fitted.

2) $\mathcal{I}_1(q)$ **measures local error sensitivity.** Using $\mathrm{tr}(qq^\top) = \|q\|_2^2$,

$$\mathcal{I}_1(q) = \mathrm{tr}(\Sigma(q)) = \mathrm{tr}(\mathrm{Diag}(q)) - \mathrm{tr}(qq^\top) = 1 - \|q\|_2^2. \tag{54}$$

Around the optimal logit $f^\star(q)$, $\mathbb{E}[\delta f] = 0$ and $\mathbb{E}[\delta f \delta f^\top] = \sigma^2 I$, The Taylor expansion gives

$$\mathbb{E}\big[\ell(f^\star(q) + \delta f; q) - \ell(f^\star(q); q)\big] = \frac{\sigma^2}{2}\,\mathrm{tr}(\Sigma(q)) + o(\sigma^2) = \frac{\sigma^2}{2}\mathcal{I}_1(q) + o(\sigma^2). \tag{55}$$

Therefore $\mathcal{I}_1(q)$ is exactly the first-order coefficient converting local logit mismatch into excess posterior KL.

$\mathcal{I}_2(q)$ **measures gradient energy.** The local gradient linearization gives $\nabla_f \ell(f^\star(q) + \delta f; q) = \Sigma(q)\delta f + o(\|\delta f\|)$. Hence

$$\mathbb{E}\|\nabla_f \ell(f^\star(q) + \delta f; q)\|_2^2 = \sigma^2\,\mathrm{tr}(\Sigma(q)^2) + o(\sigma^2) = \sigma^2 \mathcal{I}_2(q) + o(\sigma^2). \tag{56}$$

Thus $\mathcal{I}_2(q)$ measures the local gradient energy created by posterior fitting. A larger $\mathcal{I}_2$ means that small logit errors induce larger gradients and can contribute to noisier optimization.

Together, d_{act}, $\mathcal{I}_1$, and $\mathcal{I}_2$ summarize three aspects of posterior fitting: how many logit directions are active, how strongly local mismatch becomes KL error, and how much gradient energy the mismatch creates. Smaller values indicate an easier local posterior approximation problem.

Approximation difficulty across diffusion variants. We next show how the transition kernel changes these quantities. For simplicity, we fix one position and suppress the position index and the context. Let $q_{\mathrm{data}}(x_0 = k)$ be the clean-token data prior, and define the support as $S := \{k : q_{\mathrm{data}}(x_0 = k) > 0\}$.We denote the forward kernel as $q_t(j \mid k) := q(x_t = j \mid x_0 = k)$ and gives the posterior

$$q(x_0 = k \mid x_t = j) = \frac{q_{\mathrm{data}}(x_0 = k) q_t(j \mid k)}{\sum_{r \in S} q_{\mathrm{data}}(x_0 = r) q_t(j \mid r)}. \tag{57}$$

Thus the support for the posterior is,

$$\mathrm{supp}\, q(x_0 = \cdot \mid x_t = j) = S \cap \{k : q_t(j \mid k) > 0\}. \tag{58}$$

This candidate set controls d_{act}, while its concentration controls $\mathcal{I}_1 = 1 - \|q\|_2^2$.

We then discuss different diffusion variants. Please note that all the following discussion we use $q_t(j \mid k) := q(x_t = j \mid x_0 = k)$ for simplicity.

Masking diffusion. For the absorbing kernel,

$$q_t^{\mathrm{mask}}(j \mid k) = \alpha_t \mathbb{I}\{j = k\} + (1 - \alpha_t)\mathbb{I}\{j = [\mathtt{MASK}]\}.$$

If $j \neq [\mathtt{MASK}]$, the clean token must be j, so the posterior is δ_j and $d_{\mathrm{act}} = \mathcal{I}_1 = 0$. If $j = [\mathtt{MASK}]$, the corrupted token gives no token-level information and the posterior is $q_{\mathrm{data}}(x_0 = \cdot)$, so $d_{\mathrm{act}} = |S| - 1$ and $\mathcal{I}_1 = 1 - \sum_{k \in S} q_{\mathrm{data}}(x_0 = k)^2$. Since the mask event has probability $1 - \alpha_t$, we can take expectation and get

$$\mathbb{E}[d_{\mathrm{act}}^{\mathrm{mask}}] = (1 - \alpha_t)(|S| - 1), \qquad \mathbb{E}[\mathcal{I}_1^{\mathrm{mask}}] = (1 - \alpha_t)\left(1 - \sum_{k \in S} q_{\mathrm{data}}(x_0 = k)^2\right). \tag{59}$$

Uniform diffusion. For the uniform kernel,

$$q_t^{\mathrm{uni}}(j \mid k) = \alpha_t \mathbb{I}\{j = k\} + |\mathcal{V}|^{-1}(1 - \alpha_t).$$

When $0 < \alpha_t < 1$, $q_t^{\mathrm{uni}}(j \mid k) > 0$ for every $k \in S$. Hence every observed j leaves all clean tokens in S possible:

$$d_{\mathrm{act}}^{\mathrm{uni}}(j) = |S| - 1. \tag{60}$$

Uniform corruption is therefore harder than masking with more active dimension. It is also less informative such that

$$\mathbb{E}[\mathcal{I}_1^{\mathrm{mask}}] \leq \mathbb{E}[\mathcal{I}_1^{\mathrm{uni}}]. \tag{61}$$

Semantic diffusion. For a localized semantic kernel,

$$q_t^{\mathrm{sem}}(j \mid k) = \alpha_t \mathbb{I}\{j = k\} + (1 - \alpha_t) s_t^{\mathrm{sem}}(j \mid k),$$

the possible clean explanations of j are

$$S_{\mathrm{sem}}(j) := S \cap \left(\{j\} \cup \{k : s_t^{\mathrm{sem}}(j \mid k) > 0\}\right).$$

Then

$$d_{\mathrm{act}}^{\mathrm{sem}}(j) = |S_{\mathrm{sem}}(j)| - 1. \tag{62}$$

If the semantic neighborhoods are genuinely local, then

$$\{j\} \cap S \subseteq S_{\mathrm{sem}}(j) \subseteq S,$$

so semantic diffusion activates a local set of clean explanations: larger than a copied masking token, but smaller than the full clean support used by uniform diffusion. This gives

$$\mathbb{E}[d_{\mathrm{act}}^{\mathrm{mask}}] \le \mathbb{E}[d_{\mathrm{act}}^{\mathrm{sem}}] \le \mathbb{E}[d_{\mathrm{act}}^{\mathrm{uni}}]. \tag{63}$$

For $\mathcal{I}_1$, recall that $\mathcal{I}_1(q) = 1 - \|q\|_2^2$ increases as the posterior becomes less concentrated. Masking gives delta posteriors on visible tokens, uniform keeps the broadest set of explanations active, and semantic diffusion lies between them by restricting explanations to semantic neighborhoods. Therefore, under the same intermediate-kernel condition,

$$\mathbb{E}[\mathcal{I}_1^{\mathrm{mask}}] \le \mathbb{E}[\mathcal{I}_1^{\mathrm{sem}}] \le \mathbb{E}[\mathcal{I}_1^{\mathrm{uni}}]. \tag{64}$$

A simple sufficient case is a locally flat posterior over each candidate set: if q is approximately uniform over m candidates, then $\mathcal{I}_1(q) = 1 - 1/m$, which grows with m.

Finally, averaging equation 63 and equation 64 over t yields the main-text comparison:

$$\mathbb{E}\left[d_{\mathrm{act}}^{\mathrm{mask}}\right] \le \mathbb{E}\left[d_{\mathrm{act}}^{\mathrm{sem}}\right] \le \mathbb{E}\left[d_{\mathrm{act}}^{\mathrm{uni}}\right], \qquad \mathbb{E}\left[\mathcal{I}_1^{\mathrm{mask}}\right] \le \mathbb{E}\left[\mathcal{I}_1^{\mathrm{sem}}\right] \le \mathbb{E}\left[\mathcal{I}_1^{\mathrm{uni}}\right]. \tag{65}$$

Dense semantic proposals. If s_t^{sem} is implemented by a dense softmax over semantic scores, then the exact support can become the full vocabulary, making the exact rank equal to the uniform rank. In that case the same argument should be read in terms of the ϵ-effective support

$$\mathrm{supp}_\epsilon(q) := \{k : q_k \ge \epsilon\}, \qquad d_{\mathrm{act},\epsilon}(q) := |\mathrm{supp}_\epsilon(q)| - 1,$$

or equivalently in the low-temperature regime where most posterior mass lies in the semantic neighborhood. This is the practical sense in which semantic diffusion reduces the active posterior geometry relative to uniform diffusion while retaining more active directions than masking.

B.5 Proof for Exposure-bias Propagation

We restate proposition 4 and give proof here.

Proposition 7 (Exposure Bias Propagation)**.** *With mild conditions, the propagation satisfies:*

$$\mathcal{B}_{t-1} \le \eta_t \mathcal{B}_t + \rho_t,$$

where ρ_t upper-bounds the step-wise error: $\rho_t \ge \sup_{x_t} D_{\mathrm{KL}}\big(q_t(x_{t-1} \mid x_t) \| p_t^\theta(x_{t-1} \mid x_t)\big)$ and $\eta_t^{\mathrm{KL}} \in [0, 1]$ is the step-wise error propagation coefficient. In particular,

$$\eta_{t,\mathrm{mask}} = 1, \quad \eta_{t,\mathrm{rm}} \le 1 - r_t \lambda_t^{\mathrm{ref}}, \quad \eta_{t,\mathrm{uni}} \le 1 - \lambda_t^{\mathrm{uni}} < 1, \quad \eta_{t,\mathrm{sem}} \le 1 - \lambda_t^{\mathrm{sem}} < 1.$$

*Consequently, for **semantic and uniform diffusion**, $\rho_t \le \rho$ and $\eta_t \le \eta < 1$, then*

$$\mathcal{B}_{\mathrm{exp}} = \sum_{t=0}^{T} \mathcal{B}_t = \mathcal{O}\left(\frac{T\,\rho}{1 - \eta}\right);$$

*whereas for **masking diffusion** $\eta_t^{\mathrm{KL}} \approx 1$ and*

$$\mathcal{B}_{\mathrm{exp}} = \mathcal{O}\big(T^2 \rho\big).$$

Please note that, in the main text, we use $t - dt$ instead of $t - 1$ in the recursion. This is just a re-indexing of the t from exact continuous time to sampling step T, thus have no impact in our conclusion. In this proof, we follow $t = [0 : T]$ notation as this can simplify the proof.

Proof. For one reverse step $t \to t - 1$, write

$$Q_t(y \mid x) \coloneqq q(x_{t-1} = y \mid x_t = x), \qquad P_t^\theta(y \mid x) \coloneqq p_\theta(x_{t-1} = y \mid x_t = x). \tag{66}$$

Thus $q_{t-1} = q_t Q_t$, $p_{t-1} = p_t P_t^\theta$, and $\mathcal{B}_t = D_{\mathrm{KL}}(q_t \| p_t)$. We measure the repair ability of the true reverse kernel by its KL contraction coefficient

$$\eta_t \coloneqq \sup_{\mu,\nu} \frac{D_{\mathrm{KL}}(\mu Q_t \| \pi Q_t)}{D_{\mathrm{KL}}(\mu \| \pi)} \in [0, 1], \tag{67}$$

where the supremum is over pairs with finite nonzero denominator. η_t measures under true transition kernel Q_t, what will be the error propagated.

We then use ρ_t to measure the error between true kernel Q_t and learned kernel P_t^θ,

$$\rho_t \geq \sup_{x,y} \log \frac{Q_t(y \mid x)}{P_t^\theta(y \mid x)}.$$

This gives

$$Q_t(y \mid x) \leq e^{\rho_t} P_t^\theta(y \mid x), \qquad \forall x, y. \tag{68}$$

With the definition of η_t and ρ_t, we can derive the one-step exposure bias recursion.

Theorem B.1 (One-step exposure-bias recursion)**.** *Under Eq.* (67) *and Eq.* (68)*, we have*

$$\mathcal{B}_{t-1} \leq \eta_t \mathcal{B}_t + \rho_t. \tag{69}$$

Proof. We expand the per-step exposure bias as

$$\mathcal{B}_{t-1} = D_{\mathrm{KL}}(q_{t-1} \| p_{t-1}^\theta) \text{ and } \mathcal{B}_{t-1} = D_{\mathrm{KL}}(q_t \| p_t^\theta)$$

By Eq. (68), the mixed distributions also satisfy

$$(p_t Q_t)(y) = \sum_x p_t(x) Q_t(y \mid x) \leq e^{\rho_t} \sum_x p_t(x) P_t^\theta(y \mid x) = e^{\rho_t} (p_t P_t^\theta)(y).$$

Therefore, with $q_{t-1} = q_t Q_t$ and $p_{t-1} = p_t P_t^\theta$

$$\begin{aligned} D_{\mathrm{KL}}(q_{t-1} \| p_{t-1}^\theta) = D_{\mathrm{KL}}(q_t Q_t \| p_t P_t^\theta) &= \sum_y (q_t Q_t)(y) \log \frac{(q_t Q_t)(y)}{(p_t P_t^\theta)(y)} \\ &\leq \sum_y (q_t Q_t)(y) \log \frac{(q_t Q_t)(y)}{(p_t Q_t)(y)} + \rho_t \\ &= D_{\mathrm{KL}}(q_t Q_t \| p_t Q_t) + \rho_t \\ &\leq \eta_t D_{\mathrm{KL}}(q_t \| p_t^\theta) + \rho_t. \end{aligned} \tag{70}$$

This is exactly Eq. (69). □

The recursion shows that exposure bias has two sources. The term ρ_t is the local denoising error of this step. The coefficient η_t decides whether previous rollout mismatch is preserved or contracted.

When is $\eta_t < 1$? The useful case is when the reverse kernel contains a shared component (for instance, global jumping). If there exist $\lambda_t > 0$ and a distribution ν_t such that

$$Q_t(\cdot \mid x) \geq \lambda_t \nu_t(\cdot) \quad \forall x, \text{ such that } \eta_t \leq 1 - \lambda_t \tag{71}$$

We can Eq. (71) allows us to rewrite the transition

$$Q_t(\cdot \mid x) = \lambda_t \nu_t(\cdot) + (1 - \lambda_t) \widetilde{Q}_t(\cdot \mid x)$$

for another Markov kernel $\widetilde{Q}_t$. This is because for any μ, ν, joint convexity of KL gives

$$D_{\mathrm{KL}}(\mu Q_t \| \nu Q_t) \leq (1 - \lambda_t) D_{\mathrm{KL}}(\mu \widetilde{Q}_t \| \nu \widetilde{Q}_t),$$

and ordinary data processing gives

$$D_{\mathrm{KL}}(\mu \widetilde{Q}_t \| \nu \widetilde{Q}_t) \leq D_{\mathrm{KL}}(\mu \| \nu).$$

Thus, a source-independent refresh mass (for instance, the global jumping in uniform and semantic diffusion) can make the reverse step reduce exposure bias.

Instantiating the coefficient for diffusion variants. We now apply the above criterion to the transition kernels discussed in the main text. Let $a_t \in (0,1)$ denote the one-step probability of retaining the current token in the forward kernel.

Masking diffusion. For the absorbing kernel, observing a visible token $x_t = j \neq [\texttt{MASK}]$ forces $x_{t-1} = j$. Hence, on the visible-token subspace, $Q_t^{\mathrm{mask}}(\cdot \mid j) = \delta_j$. The reverse step acts as the identity on that subspace, so one can choose two visible-token distributions μ, ν for which

$$D_{\mathrm{KL}}(\mu Q_t^{\mathrm{mask}} \| \nu Q_t^{\mathrm{mask}}) = D_{\mathrm{KL}}(\mu \| \nu).$$

Therefore $\eta_{t,\mathrm{mask}} = 1$. This reflects the "early commitment" effect: once a wrong visible token appears, the standard masking reverse kernel has no mechanism to repair it.

Remasking. If a remasking sampler has the effective kernel $Q_t^{\mathrm{rm}} = (1 - r_t)Q_t^{\mathrm{mask}} + r_t Q_t^{\mathrm{ref}}$, and the refresh part satisfies

$$Q_t^{\mathrm{ref}}(\cdot \mid x) \geq \lambda_t^{\mathrm{ref}} \nu_t(\cdot), \qquad \forall x,$$

then $Q_t^{\mathrm{rm}}(\cdot \mid x) \geq r_t \lambda_t^{\mathrm{ref}} \nu_t(\cdot)$. Hence $\eta_{t,\mathrm{rm}} \leq 1 - r_t \lambda_t^{\mathrm{ref}}$. Remasking therefore helps precisely through the amount of shared refresh it injects.

Uniform diffusion. For a one-step uniform forward kernel on a vocabulary of size V, $q(x_t = j \mid x_{t-1} = y) = a_t \delta_{jy} + V^{-1}(1 - a_t)$. Bayes' rule gives

$$Q_t^{\mathrm{uni}}(y \mid j) = \frac{\left(a_t \delta_{jy} + \frac{1-a_t}{V}\right) q_{t-1}(y)}{q_t(j)} \geq \frac{1 - a_t}{V q_t(j)} q_{t-1}(y).$$

Thus Q_t^{uni} has the shared component q_{t-1} with mass

$$\lambda_t^{\mathrm{uni}} := \min_j \frac{1 - a_t}{V q_t(j)},$$

assuming $q_t(j) > 0$. By Eq. (71),

$$\eta_{t,\mathrm{uni}} \leq 1 - \lambda_t^{\mathrm{uni}} < 1. \tag{72}$$

Uniform diffusion can therefore contract rollout mismatch, which is the sampling-side repair advantage emphasized in the main text.

Semantic diffusion and SemDLM+. For a semantic forward kernel

$$q(x_t = j \mid x_{t-1} = y) = a_t \delta_{jy} + (1 - a_t) s_t^{\mathrm{sem}}(j \mid y),$$

similarly, if the semantic diffusion is equiped with a global transition ν_t as a shared component:

$$s_t^{\mathrm{sem}}(j \mid y) \geq \alpha_t^{\mathrm{sem}} \nu_t(j), \qquad \forall y, j.$$

Then

$$Q_t^{\mathrm{sem}}(y \mid j) \geq \frac{(1 - a_t)\alpha_t^{\mathrm{sem}} \nu_t(j)}{q_t(j)} q_{t-1}(y),$$

so

$$\eta_{t,\mathrm{sem}} \leq 1 - \lambda_t^{\mathrm{sem}} < 1, \qquad \lambda_t^{\mathrm{sem}} := \min_j \frac{(1 - a_t)\alpha_t^{\mathrm{sem}} \nu_t(j)}{q_t(j)}. \tag{73}$$

This serves another important factor that we need global transition in Eq. (22).

Cumulated Exposure Bias with Propagation. We then extend the one-step analysis to the overall analysis of exposure bias.

Assume $\rho_t \leq \rho$ for all t and $\eta_t \leq \eta < 1$ (this is the case for semantic and uniform diffusion), then iterating $\mathcal{B}_{t-1} \leq \eta \mathcal{B}_t + \rho$ from time $t = [0 : T]$ gives

$$\mathcal{B}_t \leq \eta^{T-t} \mathcal{B}_T + \frac{1 - \eta^{T-t}}{1 - \eta} \rho. \tag{74}$$

Summing over all the time steps $t = 0, \ldots, T$ yields

$$\sum_{t=0}^{T} \mathcal{B}_t \leq \frac{\mathcal{B}_T}{1 - \eta} + \frac{T\rho}{1 - \eta}. \tag{75}$$

If $q_T = p_T$, we get $\mathcal{B}_T = 0$, and therefore

$$\sum_{t=0}^{T} \mathcal{B}_t = \mathcal{O}\Big(\frac{T\rho}{1-\eta}\Big).$$

For masking diffusion, $\eta_t = 1$. The same recursion becomes $\mathcal{B}_{t-1} \le \mathcal{B}_t + \rho$, so

$$\mathcal{B}_t \le \mathcal{B}_T + (T-t)\rho$$

and hence

$$\sum_{t=0}^{T} \mathcal{B}_t \le (T+1)\mathcal{B}_T + \frac{T(T+1)}{2}\rho. \tag{76}$$

When $q_T = p_T$, this reduces to

$$\sum_{t=0}^{T} \mathcal{B}_t = \mathcal{O}(T^2\rho).$$

This proves the comparison in Proposition 4: kernels with shared support, such as uniform diffusion and SemDLM+ with global jumping, contract exposure bias across steps, while masking can accumulate it quadratically in the number of reverse steps.

B.6 Transition Kernel Dispersion and Variance

B.6.1 Connecting and unifying sampling and approximation variance

As pointed out in sec. 3.1, we will show that sampling variance can be controlled by the approximation variance, such that in the main text we only consider the approximation variance in analysis.

Final-sample risk and posterior approximation. To recall, the rollout risk $\mathcal{R}$ the rollout risk at the final time $t = 0$, where $\hat{p}_S(x_0)$ is the distribution of the generated samples. We can then show the expected risk can be upper bounded by the approximation approximation error,

$$\begin{aligned}
&\mathbb{E}_S[\mathcal{R}(\hat{\theta}_S)] = \mathbb{E}_S D_{\mathrm{KL}}(q(x_0)\|\hat{p}_S(x_0)) = \mathcal{B}_0 + \mathcal{V}_0 \\
&\le \underbrace{D_{\mathrm{KL}}(q(x_T)\|p(x_T))}_{\text{Initial mismatch}} + \sum_{t=1}^{T} \mathbb{E}_q\Big[\underbrace{D_{\mathrm{KL}}(q(x_0 \mid x_t)\|\bar{p}(x_0 \mid x_t))}_{\text{Approximation bias}} + \underbrace{\mathcal{V}(x_t)}_{\text{Approximation variance}}\Big].
\end{aligned} \tag{77}$$

Thus cumulative approximation bias and variance control an upper bound on the risk. Thus, the above inequality makes explicit why we study approximation variance in the main text.

Proof:

We follow the notation of the main text: With optimization randomness S, $\hat{p}_S(x_0 \mid x_t)$ denotes the learned clean-data predictor, $\hat{p}_S(x_t)$ the actual rollout marginal, and $\bar{p}$ their respective expectations over training-data.

Consider a fixed finite sampling grid $t = 0, \dots, T$, using the same trained model S at every reverse step. The x_0-prediction parameterization gives

$$\hat{p}_S(x_{t-1} \mid x_t) = \sum_{x_0} q(x_{t-1} \mid x_t, x_0)\hat{p}_S(x_0 \mid x_t). \tag{78}$$

Here $q(x_{t-1} \mid x_t, x_0)$ is pre-specified forward process. With

$$\hat{p}_S(x_{t-1}) = \sum_{x_t} \hat{p}_S(x_t)\hat{p}_S(x_{t-1} \mid x_t).$$

and assuming a common initial distribution $\hat{p}_S(x_T) = p(x_T)$ leads to $\bar{p}(x_0 \mid x_t) \coloneqq \mathbb{E}_S \hat{p}_S(x_0 \mid x_t)$ and $\bar{p}(x_t) \coloneqq \mathbb{E}_S \hat{p}_S(x_t)$. Recall the defined approximation variance $\mathcal{V}(x_t)$ and the rollout quantities $\mathcal{B}_t, \mathcal{V}_t$:

$$\mathcal{V}(x_t) \coloneqq \mathbb{E}_{q(x_0|x_t)}[\log \bar{p}(x_0 \mid x_t) - \mathbb{E}_S \log \hat{p}_S(x_0 \mid x_t)], \tag{79}$$

$$\mathcal{B}_t \coloneqq D_{\mathrm{KL}}(q(x_t)\|\bar{p}(x_t)), \tag{80}$$

$$\mathcal{V}_t \coloneqq \mathbb{E}_{q(x_t)}[\log \bar{p}(x_t) - \mathbb{E}_S \log \hat{p}_S(x_t)]. \tag{81}$$

The per-step approximation variance is $\mathbb{E}_{q(x_t)}[\mathcal{V}(x_t)]$, whereas $\mathcal{V}_t$ is the per-step rollout variance. We can then have the folliwng proposition

Proposition 8 (Rollout risk controlled by approximation error)**.** *For every $t \in \{0, \dots, T\}$,*

$$\mathcal{B}_t + \mathcal{V}_t \le D_{\mathrm{KL}}(q(x_T)\|p(x_T)) + \sum_{s=t+1}^{T} \mathbb{E}_{q(x_s)}\Big[D_{\mathrm{KL}}(q(x_0 \mid x_s)\|\bar{p}(x_0 \mid x_s)) + \mathcal{V}(x_s)\Big]. \tag{82}$$

If additionally $p(x_T) = q(x_T)$ and $\bar{p}(x_0 \mid x_s) = q(x_0 \mid x_s)$ for $q(x_s)$-almost every x_s and every $s > t$, then

$$\mathcal{V}_t \le \mathcal{B}_t + \mathcal{V}_t \le \sum_{s=t+1}^{T} \mathbb{E}_{q(x_s)}[\mathcal{V}(x_s)]. \tag{83}$$

Proof. For each fixed x_t, we have proved in the previous sections that

$$D_{\mathrm{KL}}(q(x_{t-1} \mid x_t)\|\hat{p}_S(x_{t-1} \mid x_t)) \le D_{\mathrm{KL}}(q(x_0 \mid x_t)\|\hat{p}_S(x_0 \mid x_t)).$$

where the equality holds when the forward process is assumed to be equivalemnt. Then the joint distribution $q(x_t)q(x_{t-1} \mid x_t)$ and $\hat{p}_S(x_t)\hat{p}_S(x_{t-1} \mid x_t)$ have marginals $q(x_{t-1})$ and $\hat{p}_S(x_{t-1})$, respectively. Therefore, marginalization and the KL chain rule therefore yield

$$\begin{aligned} &D_{\mathrm{KL}}(q(x_{t-1})\|\hat{p}_S(x_{t-1})) \\ &\quad \le D_{\mathrm{KL}}(q(x_t)\|\hat{p}_S(x_t)) + \mathbb{E}_{q(x_t)} D_{\mathrm{KL}}(q(x_{t-1} \mid x_t)\|\hat{p}_S(x_{t-1} \mid x_t)) \\ &\quad \le D_{\mathrm{KL}}(q(x_t)\|\hat{p}_S(x_t)) + \mathbb{E}_{q(x_t)} D_{\mathrm{KL}}(q(x_0 \mid x_t)\|\hat{p}_S(x_0 \mid x_t)). \end{aligned} \tag{84}$$

Iterating from T to t and taking $\mathbb{E}_S$ gives

$$\begin{aligned} \mathbb{E}_S D_{\mathrm{KL}}(q(x_t)\|\hat{p}_S(x_t)) \le{}& D_{\mathrm{KL}}(q(x_T)\|p(x_T)) \\ &+ \sum_{s=t+1}^{T} \mathbb{E}_{q(x_s)} \mathbb{E}_S D_{\mathrm{KL}}(q(x_0 \mid x_s)\|\hat{p}_S(x_0 \mid x_s)). \end{aligned}$$

The bias–variance decompositions in the main text identify the left-hand side as $\mathcal{B}_t + \mathcal{V}_t$ and each summand as

$$\mathbb{E}_{q(x_s)} D_{\mathrm{KL}}(q(x_0 \mid x_s)\|\bar{p}(x_0 \mid x_s)) + \mathbb{E}_{q(x_s)}[\mathcal{V}(x_s)].$$

This proves the general bound. □

With such a bound, we motivate an analysis on approximation variance (and this removes the necessaries in analyzing sampling variance) as a training-side contribution to rollout risk.

B.6.2 The gradient guided variance

To connect the variance term to a more classical estimation-variance picture, we rewrite the KL divergence as the combination of negative log-likelihood and entropy.

$$\mathcal{L} = -\mathbb{E}_{x \sim p_{\mathrm{data}}(x)} [\log p_\theta(x)] - \text{Entropy} \tag{85}$$

And define the per-sample loss $\ell_t(\theta) := -\log p_\theta(x_0 \mid x_t)$, per-step risks $\mathcal{L}_t(\theta) := \mathbb{E}_{x_t}[\ell_t(\theta)]$ and use ∇_θ as the gradient operator. Let $\theta^\star$ minimize the population risk $\mathcal{L}(\theta) = \mathbb{E}_t[\mathcal{L}_t(\theta)]$, so that $\mathbb{E}_t[g_t(\theta^\star)] = 0$. At $\theta^\star$, the law of total variance yields

$$\begin{aligned} &\mathrm{Var}_{t,x_t}\big(\nabla \ell_t(\theta^\star)\big) \\ =&\mathbb{E}_t\left[\mathrm{Var}_{x_t}\big(\nabla \ell_t(\theta^\star)\big)\right] + \mathrm{Var}_t\left(\mathbb{E}_{x_t}\left[\nabla \ell_t(\theta^\star)\right]\right), \\ =&\underbrace{\mathbb{E}_t\left[\mathrm{Var}_{x_t}\big(\nabla \ell_t(\theta^\star)\big)\right]}_{\text{within-step noise}} + \underbrace{\mathrm{Var}_t\left(\nabla \mathcal{L}_t(\theta^\star)\right)}_{\text{between-step heterogeneity}}. \end{aligned} \tag{86}$$

The second term $\mathrm{Var}_t\left(\nabla \mathcal{L}_t(\theta^\star)\right)$ is a *kernel-induced heterogeneity*. It quantifies how the gradient varies across different diffusion steps t.

Remark. Crucially, $\nabla \mathcal{L}_t(\theta^\star)$ depends on Q_t via sampling x_t, hence a larger dispersion of Q_t will induce larger between-t heterogeneity. Other than the theoretical results, Fig. 4 empirically valid that the gradient variance is strongly correlated with the dispersion of the transition kernel across time t.

Now we aim to rigorously derive that as Dispersion Q_t increases, Between-t Heterogeneity increases.

We define the between-t heterogeneity as:

$$\mathcal{V}(Q) \coloneqq \mathrm{Var}_t\big(\nabla\mathcal{L}_t(\theta^\star;Q)\big) = \mathbb{E}_t\big[\|\nabla\mathcal{L}_t(\theta^\star;Q)\|^2\big], \tag{87}$$

where the equality holds because $\mathbb{E}_t[\nabla\mathcal{L}_t(\theta^\star;Q)] = \mathbf{0}$.

We define the posterior entropy as $h(t;Q) \coloneqq \mathbb{E}_{x_t\sim q_t(\cdot;Q_t)}\big[H\big(q(x_0 \mid x_t;Q_t)\big)\big]$, where $H(\cdot)$ is Shannon entropy. Then we define the kernel dispersion as $\mathcal{D}(Q) \coloneqq \mathrm{Var}_t\big(h(t;Q)\big) = \mathbb{E}_t\big[h(t;Q)^2\big] - \big(\mathbb{E}_t[h(t;Q)]\big)^2$. This quantifies how posterior uncertainty fluctuates across diffusion steps. Uniform diffusion yields large $\mathcal{D}(Q)$ (entropy spans $[0,\log|\mathcal{V}|]$); semantic diffusion yields small $\mathcal{D}(Q)$ (entropy concentrated near $\log k$), and Masking diffusion has the lowest dispersion.

We make the following assumptions:

1) the approximation error scales with posterior entropy, such that there exists $c_1 > 0$ such that for all t,

$$\mathbb{E}_{x_t}\big[\|p_{\theta^\star}(\cdot \mid x_t) - q(\cdot \mid x_t;Q_t)\|_1\big] \geq c_1 \cdot h(t;Q). \tag{88}$$

This is induced by the fact that high-entropy posteriors (induced by dispersed Q_t) are harder to approximate with finite-capacity models

2) The gradients are bounded and stable. $\|\nabla_\theta f_{\theta^\star}(x_t)\| \leq M$ for all x_t, and the direction of $\nabla_\theta f_{\theta^\star}(x_t)$ varies smoothly over semantically coherent neighborhoods.

Proposition 9. *There exists a constant $c = c_1/M > 0$ such that:*

$$\mathcal{V}(Q) \geq c^2 \cdot \mathcal{D}(Q). \tag{89}$$

Consequently, if two generators satisfy $\mathcal{D}(Q^A) > \mathcal{D}(Q^B)$, then $\mathcal{V}(Q^A) > \mathcal{V}(Q^B)$.

proof: We previously derived the gradient derivation for cross-entropy loss, and we re-formula it here. (in Sec. B.4)

$$\begin{aligned}\nabla\mathcal{L}_t(\theta^\star;Q) &= \mathbb{E}_{x_t}\Big[\mathbb{E}_{x_0|x_t}\big[(p_{\theta^\star}(x_0 \mid x_t) - q(x_0 \mid x_t;Q_t)) \odot \nabla_\theta f_{\theta^\star}(x_t)\big]\Big] \\ &= \mathbb{E}_{x_t}\big[\boldsymbol{\delta}_t(x_t;Q)\big],\end{aligned} \tag{90}$$

where $\ell_t(x_t;Q)$ denotes the per-sample gradient signal. By Hölder's inequality:

$$\begin{aligned}\|\nabla\mathcal{L}_t(\theta^\star;Q)\| &\geq \big|\mathbb{E}_{x_t}\big[\|\boldsymbol{\delta}_t(x_t;Q)\|\big]\big| \\ &\geq c_1 \cdot \mathbb{E}_{x_t}\big[H(q(x_0 \mid x_t;Q_t))\big] \cdot \frac{1}{M} \\ &= c \cdot h(t;Q), \quad c \coloneqq c_1/M > 0.\end{aligned} \tag{91}$$

Squaring both sides and taking expectation over t:

$$\begin{aligned}\mathcal{V}(Q) &= \mathbb{E}_t\big[\|\nabla\mathcal{L}_t(\theta^\star;Q)\|^2\big] \\ &\geq c^2 \cdot \mathbb{E}_t\big[h(t;Q)^2\big] \\ &= c^2 \cdot \Big(\mathrm{Var}_t\big(h(t;Q)\big) + \big(\mathbb{E}_t[h(t;Q)]\big)^2\Big) \\ &\geq c^2 \cdot \mathrm{Var}_t\big(h(t;Q)\big) \\ &= c^2 \cdot \mathcal{D}(Q).\end{aligned} \tag{92}$$

The directly completes the proof.

Remark: $\mathcal{D}(Q)$ explicitly captures how Q_t's structural dispersion, such as support size, entropy trajectory, propagates to gradient statistics. This suggests that minimizing kernel dispersion $\mathcal{D}(Q)$ during design can reduce between-t gradient heterogeneity. This inspires us to design SemDLM.

C More about Diffusion as CTMC

In Sec. 2, we mentioned that the local transition $q_{t|t-dt}$, the cumulative forward $q_{t|0}$, and the generator Q_t are equivalent representations of the same forward process. The equivalence is obtained via Bayesian rule: $q\left(x_{t-dt} \mid x_t, x_0\right) \propto q\left(x_t \mid x_{t-dt}\right) q(x_{t-dt} \mid x_0)$.

We thus present other details for the process. We first detail the forward process and reverse process in CTMC. The forward process is derived by the *infinitesimal generator* Q_t, i.e., $\frac{dq_t}{dt} = Q_t q_t, 0 \le t \le 1$, where

$$\text{Forward Process via Euler Sampling: } q\left(x_{t+dt} = y \mid x_t = z\right) = \delta_{zy} + Q_t(z, y)dt + O\left(dt^2\right) \tag{93}$$

and the corresponding reverse process is

$$\text{Reverse-time Euler Step: } q(x_{t-dt} = z \mid x_t = y) = \delta_{yz} + \bar{Q}_t(y, z)dt + O(dt^2),$$

With the reverse-time infinitesimal generator

$$\bar{Q}_t(y, z) = Q_t(z, y)\frac{q_t(z)}{q_t(y)}, \quad z \neq y, \qquad \bar{Q}_t(y, y) = -\sum_{z \neq y} \bar{Q}_t(y, z).$$

where the density ratio $q_t(z)/q_t(y)$ is typically approximated by a learned score or posterior model in practice.

Thus, it is clear that the local transition $q_{t|t-dt}$, the cumulative forward $q_{t|0}$, and the generator Q_t are equivalent representations of the same forward/reverse process. For completeness, we list the other representation of masking, uniform and semantic diffusion.

Absorbing (Mask) Transition defines an absorbing token $[\texttt{MASK}]$ to make $Q_t([\texttt{MASK}], y) = 0$ and

$$Q_t(z, y) = \begin{cases} \lambda(t), & z \neq [\texttt{MASK}], y = [\texttt{MASK}], \\ -\lambda(t), & z \neq [\texttt{MASK}], y = z. \end{cases} \tag{94}$$

where $\lambda(t)$ is the time schedule. This induces a cumulated kernel directly from x_0 to x_t:

$$q(x_t = j \mid x_0 = i) = \alpha_t \delta_{ij} + (1 - \alpha_t)\delta_{j,[\texttt{MASK}]}, \tag{95}$$

where $\alpha_t = \exp\left(-\int_0^t \lambda(s)ds\right)$ and δ_{ij} is the Kronecker delta.

Uniform Transition is another choice that has been shown to have favorable scaling behavior [36]:

$$Q_t(z, y) = \begin{cases} \frac{\lambda(t)}{K}, & y \neq z, \\ -\lambda(t)\left(1 - \frac{1}{K}\right), & y = z. \end{cases} \tag{96}$$

which has the following cumulated kernel:

$$q(x_t = j \mid x_0 = i) = \alpha_t \delta_{ij} + \frac{1 - \alpha_t}{K}, \quad \forall i, j \in \mathcal{V}, \tag{97}$$

with the same decay factor $\alpha_t = \exp\left(-\int_0^t \lambda(s)ds\right)$.

Semantic Transition Let $\mathrm{KNN}(z)$ be the k nearest neighbors of z and $w(z, y) \ge 0$ be edge weights, the transition kernel is then,

$$Q_t(z, y) = \begin{cases} \lambda(t)\frac{w(z,y)}{\sum_{y' \in \mathrm{KNN}(z)} w(z,y')}, & y \in \mathrm{KNN}(z), \\ -\lambda(t), & y = z, \\ 0, & \text{otherwise.} \end{cases} \tag{98}$$

This gives a forward kernel as,

$$q(x_t = j \mid x_0 = i) = \alpha_t \delta_{ij} + \frac{1 - \alpha_t}{k_t}\mathbb{I}\left(j \in \mathcal{N}_k(i)\right), \tag{99}$$

where $\mathcal{N}_k(i)$ is the top-k semantic neighborhood of token i.

D Additional Design for Semantic Diffusion

Algorithm 1 Training of SemDLM

Require: Dataset $\mathcal{D} \sim p_0$
Ensure: Trained model $f_\theta(x_t, t)$
1: Initialize parameters θ
2: **while** not converged **do**
3: Sample a minibatch $\{x_0^{(i)}\}_{i=1}^B \sim \mathcal{D}$
4: Sample $t \sim \mathcal{U}(0,1)$
5: Sample noisy states $x_t^{(i)} \sim p(x_t \mid x_0^{(i)})$ via Eq. (22)
6: Compute $p_{0|t}^\theta(\cdot \mid x_t^{(i)}) = f_\theta(x_t^{(i)}, t)$
7: Update θ by minimizing Eq. (3)
8: **end while**

Algorithm 2 Sampling from SemDLM

Require: Reference distribution p_1, trained model $f_\theta(x_t, t)$, step size Δt
Ensure: Generated sample $\hat{x}_0$
1: Sample initial state $\hat{x}_1 \sim p_1$
2: **for** $t = 1, 1 - \Delta t, \ldots, \Delta t$ **do**
3: Compute $p_{0|t}^\theta(\cdot \mid \hat{x}_t) = f_\theta(\hat{x}_t, t)$
4: Sample $\tilde{x}_0 \sim p_{0|t}^\theta(\cdot \mid \hat{x}_t)$
5: Compute generator $Q_\theta(\hat{x}_t \mid \tilde{x}_0)$
6: Sample $\hat{x}_{t-\Delta t} \sim \hat{x}_t + Q_\theta(\hat{x}_t \mid \tilde{x}_0)\,\Delta t$
7: **end for**
8: **return** $\hat{x}_0$

D.1 Analysis of Semantic Basins

Posterior logit decomposition. Again, for a fixed position i, let the current sampling state be $\hat{x}_t$, and write $\hat{x}_t^i = j$. The ideal denoising posterior satisfies

$$q(x_0^i = k \mid \hat{x}_t) \propto p_{\text{data}}(x_0^i = k \mid \hat{x}_t^{-i}) q_t(j \mid x_0^i = k, \hat{x}_t^{-i}). \tag{100}$$

Thus, up to a normalization constant, the ideal logits $l_i^\star(k; \hat{x}_t)$

$$l_i^\star(k; \hat{x}_t) = \underbrace{\log p_{\text{data}}(x_0^i = k \mid \hat{x}_t^{-i})}_{\text{contextual prior}} + \underbrace{\log q_t(\hat{x}_t^i \mid x_0^i = k, \hat{x}_t^{-i})}_{\text{local forward likelihood}} + \text{const}. \tag{101}$$

The first term is the contextual prior, while the second term is the local forward likelihood. During sampling, the context $\hat{x}_t^{-i}$ is generated by the model itself. Hence the actual model logit can be written as

$$l_{\theta,i}(k; \hat{x}_t) = l_i^\star(k; \hat{x}_t) + \Delta_{\text{roll},i}(k; \hat{x}_t) + \epsilon_i(k), \tag{102}$$

where $\Delta_{\text{roll},i}$ denotes rollout-induced logit bias. The rollout bias can be locally approximated as

$$\Delta_{\text{roll},i}(k; \hat{x}_t) \approx \sum_{l \in V} A(k,l) n_i^{(W)}(l), \text{ with } A(k,l) > 0.$$

Why do semantically similar tokens induce positive contextual contribution? $A(k,l)$ measures contextual association: whether the occurrence of token l in the recent context provides positive evidence for predicting token k at position i.

At the data-distribution level, one can define the contextual contribution as a log-prior shift:

$$A_i(k,l) := \log \frac{p_{\text{data}}(x_0^i = k \mid x_t^r = l, \text{rest})}{p_{\text{data}}(x_0^i = k \mid x_t^r = \text{neutral}, \text{rest})}. \tag{103}$$

Thus, $A_i(k,l) > 0$ means that observing l in the context increases the conditional prior probability of k. More generally, tokens in the same semantic cluster tend to have positive pointwise contextual association under coherent natural language contexts.

At the model level, the same effect can be understood through a local linearization of the Transformer logits. Suppose the sampling logit is

$$l_{\theta,i}(k; \hat{x}_t) = u_k^\top h_i(\hat{x}_t) + b_k,$$

where $h_i(\hat{x}_t)$ is the hidden state at position i and u_k is the output vector for token k. Around a neutral baseline $\bar{x}_t$, the hidden state can be locally approximated as

$$h_i(\hat{x}_t) \approx h_i(\bar{x}_t) + \sum_{r \in W} B_{ir} e_{\hat{x}_t^r},$$

where $e_{\hat{x}_t^r}$ is the embedding of the token at position r, and B_{ir} summarizes the local attention and feed-forward influence from position r to position i. Substituting this into the logit gives

$$l_{\theta,i}(k;\hat{x}_t) - l_{\theta,i}(k;\bar{x}_t) \approx \sum_{r\in W} u_k^\top B_{ir} e_{\hat{x}_t^r}. \tag{104}$$

Hence, the contextual contribution of token l to the logit of k can be approximated by

$$A_{\theta,i}(k,l) \approx u_k^\top B_{ir} e_l. \tag{105}$$

If k and l are semantically related, their embedding and output directions are often aligned in language models, making A contribution more likely to be positive.

Rollout-induced positive feedback. The rollout-induced logit bias can be locally approximated as

$$\Delta_{\mathrm{roll},i}(k;\hat{x}_t) \approx \sum_{r\in W} A(k,\hat{x}_t^r) = \sum_{l\in V} A(k,l) n_i^{(W)}(l). \tag{106}$$

This approximation states that repeated contextual contributions accumulate in the logit. If $A(k,l) > 0$ for semantically related tokens $k, l \in C$, then over-production of cluster C increases the logits of tokens in C. Thus,

$$n_i^{(W)}(C) \uparrow \Rightarrow \Delta_{\mathrm{roll},i}(k) \uparrow \text{ for } k \in C \Rightarrow p_\theta(x_0^i \in C \mid \hat{x}_t) \uparrow .$$

This is the rollout-induced positive feedback that drives semantic basin formation.

Likelihood amplification. Let $C \subseteq V$ be a semantic cluster and suppose the current token $j = \hat{x}_t^i$ belongs to C. For a semantic kernel, we have $q_t(j \mid k)$ large for $k \in C$ and $q_t(j \mid k)$ small for $k \notin C$. Define $a_{\mathrm{in}} := \inf_{k\in C} q_t(j \mid k)$ and $a_{\mathrm{out}} := \sup_{k\notin C} q_t(j \mid k)$. If $a_{\mathrm{in}} > a_{\mathrm{out}}$, then the posterior odds of cluster C are amplified relative to the contextual prior odds:

$$\frac{q(x_0^i \in C \mid \hat{x}_t)}{q(x_0^i \notin C \mid \hat{x}_t)} = \frac{\sum_{k\in C} p_{\mathrm{data}}(k \mid \hat{x}_t^{-i}) q_t(j \mid k)}{\sum_{k\notin C} p_{\mathrm{data}}(k \mid \hat{x}_t^{-i}) q_t(j \mid k)} \geq \frac{a_{\mathrm{in}}}{a_{\mathrm{out}}} \cdot \frac{p_{\mathrm{data}}(x_0^i \in C \mid \hat{x}_t^{-i})}{p_{\mathrm{data}}(x_0^i \notin C \mid \hat{x}_t^{-i})}. \tag{107}$$

Thus, the semantic likelihood itself increases the cluster-level log-odds. For a highly local semantic kernel, a_{out} can be very small, so the likelihood amplification can be large.

Combining the semantic likelihood amplification in Eq. (107) and the rollout-induced feedback in Eq. (106), semantic diffusion has two significant sources of cluster amplification, which makes semantic basins more severe in SemDLM.

D.2 Global Jumping as Basin Escape

We show why the global jumping kernel can mitigate semantic basin. Consider a kernel,

$$q_t(j \mid k, c) = \alpha_t \delta_{kj} + \beta_t \nu_t(j) + (1 - \alpha_t - \beta_t) s_t^{\mathrm{sem}}(j \mid k, c). \tag{108}$$

The global component gives $q_t(j \mid k, c) \geq \beta_t \nu_t(j)$ As a result, even if k is outside the semantic neighborhood of j, it still receives non-zero likelihood.

Let $j \in C$, $k_{\mathrm{in}} \in C$ and $k_{\mathrm{out}} \notin C$. With global jumping, $q_t(j \mid k_{\mathrm{out}}, c) \geq \beta_t \nu_t(j)$, and thus the likelihood ratio is bounded by

$$\frac{q_t(j \mid k_{\mathrm{in}}, c)}{q_t(j \mid k_{\mathrm{out}}, c)} \leq \frac{\alpha_t \delta_{k_{\mathrm{in}} j} + \beta_t \nu_t(j) + (1 - \alpha_t - \beta_t) s_t^{\mathrm{sem}}(j \mid k_{\mathrm{in}}, c)}{\beta_t \nu_t(j)}. \tag{109}$$

Therefore, global jumping caps the log-likelihood advantage of the local semantic cluster, which prevents the positive feedback introduced in Sec. 4.1.

D.3 Semantic-Frequency Penalty as Negative Feedback

We rewrite the semantic frequency penalty term:

$$m_i^{(W)}(k) = \sum_{l\in V} S_+(k,l) n_i^{(W)}(l), \tag{110}$$

where $S_+(k,l) \geq 0$ measures positive semantic association. with the semantic frequency penalty, the new logits are

$$\tilde{l}_{\theta,i}(k) = l_{\theta,i}(k) - \lambda_{\text{freq}}\psi(n_i^{(W)}(k)) - \lambda_{\text{sem}}\psi(m_i^{(W)}(k)), \qquad \psi(u) = \log(1+u). \tag{111}$$

For two tokens k and l, the corrected log-ratio satisfy

$$\begin{aligned}\log\frac{\tilde{p}_\theta(k)}{\tilde{p}_\theta(l)} = \log\frac{p_\theta(k)}{p_\theta(l)} &- \lambda_{\text{freq}}\Big[\psi(n_i^{(W)}(k)) - \psi(n_i^{(W)}(l))\Big] \\ &- \lambda_{\text{sem}}\Big[\psi(m_i^{(W)}(k)) - \psi(m_i^{(W)}(l))\Big].\end{aligned}$$

Therefore, a token receives lower relative odds if either the token itself has been over-produced, or its semantic neighborhood has been over-produced. We can connect this penalty to the rollout-induced positive feedback. Suppose the semantic feedback satisfies

$$\bar{\Delta}_{\text{roll}}(C) - \bar{\Delta}_{\text{roll}}(C^c) \leq \eta_{\text{sem}}\Big[\psi(m_i^{(W)}(C)) - \psi(m_i^{(W)}(C^c))\Big], \tag{112}$$

where $\bar{\Delta}_{\text{roll}}(C)$ denotes the average rollout-induced logit shift for tokens in C. After applying the semantic penalty, the net cluster-level semantic feedback is bounded by

$$\begin{aligned}&\big[\bar{\Delta}_{\text{roll}}(C) - \bar{\Delta}_{\text{roll}}(C^c)\big] - \lambda_{\text{sem}}\Big[\psi(m_i^{(W)}(C)) - \psi(m_i^{(W)}(C^c))\Big] \\ &\leq -(\lambda_{\text{sem}} - \eta_{\text{sem}})\Big[\psi(m_i^{(W)}(C)) - \psi(m_i^{(W)}(C^c))\Big].\end{aligned}$$

Thus, when $\lambda_{\text{sem}} \geq \eta_{\text{sem}}$, the semantic penalty cancels the positive semantic feedback.

Pure frequency penalty as a special case. The exact frequency penalty is recovered by setting $\lambda_{\text{sem}} = 0$ in Eq. (111), or equivalently by choosing $S_+(k,l) = \mathbb{I}\{k = l\}$. In this case,

$$m_i^{(W)}(k) = n_i^{(W)}(k),$$

and the semantic penalty reduces to an exact-token repetition penalty.

E Additional Experimental Details

E.1 Practical Designs

Model setup. Following Arriola et al. [1] and Sahoo et al. [29], we use a Transformer backbone with rotary positional embeddings. We adopt the small-scale architecture of Arriola et al. [1], resulting in a 110M-parameter model.

Practical SemDLM+ kernel. Since a full-vocabulary semantic softmax in Eq. (22) is expensive for large vocabularies, we use a top-k nearest neighborhoods as a practical approximation, i.e.

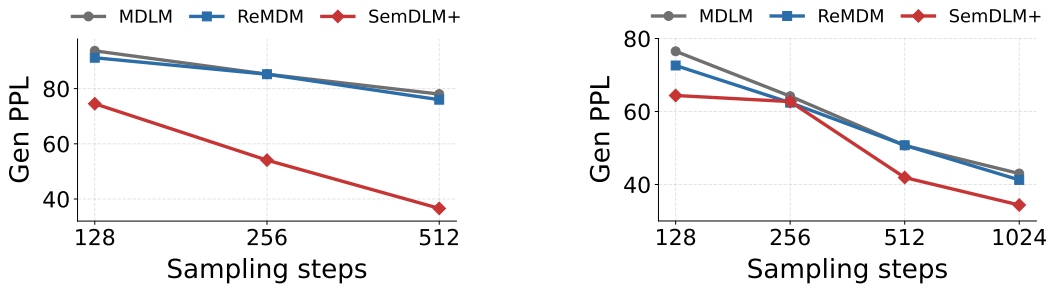
$$q\big(x_t = j \mid x_0 = i\big) = \alpha_t\delta_{ij} + \beta_t\nu_t(j) + \frac{1-\alpha_t-\beta_t}{k_t}\mathbb{I}\big(j \in \mathcal{N}_{k_t}(i)\big), \tag{113}$$

where $\mathcal{N}_{k_t}(i)$ is the top-k_t semantic neighborhood of token i. We exclude i from $\mathcal{N}_{k_t}(i)$ so that α_t exclusively controls self-retention. A simple schedule is

$$\alpha_t = \alpha_{\min} + (1-\alpha_{\min})(1-t)^\beta, \qquad \beta_t = \beta_{\max}t^\eta, \qquad k_t = 1 + (k_{\max}-1)t^\gamma, \tag{114}$$

with $\alpha_{\min} \in (0,1)$ and $\beta, \eta, \gamma > 0$. Early diffusion ($t \to 1$) uses larger neighborhoods and a stronger shared transition term for exploration and overlap; late diffusion ($t \to 0$) increases self-retention, shrinks the neighborhood, and can anneal β_t downward for precise refinement.

This kernel keeps the main properties of SemDLM+: the semantic neighborhood keeps the posterior concentrated, while the global transition branch provides shared support and improves sampling-side repair. In implementation, we store only a finite top-$k_{\max}$ neighbor table and absorb the remaining probability mass into the global transition branch, which keeps memory usage manageable.



(a) LM1B.

(b) OpenWebText.

Figure 8: Generation PPL as the number of sampling steps increases. SemDLM+ shows stronger improvement with additional refinement steps, indicating better sampling-side repair.

Dataset-specific global transition. We also test the unigram global transition For LM1B, a simple uniform transition is sufficient. For OpenWebText, we use a unigram transition where the transition weights is determined by the word frequency in the dataset:

$$\nu_\tau^{\mathrm{owt}}(j) = \frac{\mathbb{I}\{j \in \mathcal{V}_{\mathrm{valid}}^{\mathrm{owt}}\}\,\hat{\nu}(j)^\tau}{\sum_{k \in \mathcal{V}_{\mathrm{valid}}^{\mathrm{owt}}} \hat{\nu}(k)^\tau}, \qquad 0 < \tau \le 1, \tag{115}$$

where $\hat{\nu}$ is the empirical unigram distribution and $\mathcal{V}_{\mathrm{valid}}^{\mathrm{owt}}$ removes special symbols, continuation wordpieces, punctuation-only tokens, and extremely rare tokens. This avoids the unnatural proposals produced by uniform transition on open-domain text.

Changed-aware training objective. In practice, many positions remain unchanged under the forward kernel. If they are weighted equally, the model can overuse a trivial copy shortcut. We therefore upweight corrupted positions during training. Let $m_i \in \{0, 1\}$ indicate whether position i was changed by the forward kernel. We optimize

$$\mathcal{L}_{\mathrm{train}} = \frac{\sum_i w_i \ell_i}{\sum_i w_i}, \qquad w_i = \lambda_{\mathrm{chg}} \mathbb{I}\{m_i = 1\} + \lambda_{\mathrm{same}} \mathbb{I}\{m_i = 0\}, \tag{116}$$

with $\lambda_{\mathrm{chg}} \gg \lambda_{\mathrm{same}}$. This preserves the semantic denoising signal while keeping optimization stable.

Blockwise predictor sampler. For generation, we use a semi-autoregressive blockwise sampler. At each step, the model predicts $p_\theta(x_0 \mid x_t)$, samples a truncated clean proposal, projects it to the next lower-noise level using the same practical forward kernel, and updates only low-confidence positions. The replacement probability is approximated by

$$\Pr(\text{replace at position } i) \approx \frac{p_t - p_s}{p_t}\Big(1 - p_\theta(x_t^i \mid x_t)\Big)^\rho, \tag{117}$$

optionally with a freeze threshold for already confident tokens.

E.2 Addtional Figures

Figure 8 draws the impact of sampling steps on generation quality.

E.3 Computational Cost

The experiments were run on 8x NVIDIA A100-SXM4-80GB GPUs. For LM1B experiments, the training time is typically 72 hours and for OWT the training time is 144 Hours. The sampling takes approximately 5 mins in the same infrastructure.