WATCH YOUR (TIME-)STEP: ADAPTIVE SAMPLING TO IMPROVE DIFFUSION LANGUAGE MODEL TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

The success of *auto-regressive language models* (ARMs) has spurred interest in *diffusion language models* (DLMs), which can generate multiple tokens in parallel. Recent masked-diffusion language models often reuse bidirectional denoisers familiar from masked language modeling, but differ in how they distribute training across corruption levels. We focus on one common assumption in these models: uniform time-step sampling for noising sequences. We first view masked diffusion language modeling (MDLM) as risk estimation over a continuum of fixed-rate masked-language-modeling (MLM) denoising objectives. This perspective explains both why a fixed masking rate can be optimization-friendly, since it repeatedly trains the same corruption level, and why MDLM requires ELBO-dependent weights to estimate the full diffusion objective without bias. The same analysis shows that weighted losses can have time-step-dependent second moments, making uniform time-step sampling statistically inefficient. Motivated by this, we propose ***No**rmalized **2**nd **Mo**ment **Aware**ness* (`NoNoMoAware`), an adaptive strategy that maintains online estimates of the weighted-loss second moment, fits a Beta proposal over time-steps, and corrects with importance weights to preserve unbiased loss estimation. For empirical validation, we conduct experiments using different backbones during both pre-training and instruction-tuning, with our method showing downstream improvements on a number of different benchmarks. Our findings illustrate the need for more principled investigation into the training dynamics of DLMs and offer practical guidance for training such models.

1 INTRODUCTION

Large language models (LLMs) (Qwen Team, 2025; OpenAI, 2025; Llama Team, 2024; Guo et al., 2025) have reshaped machine learning in because of their strong generative and reasoning abilities (Yang et al., 2024; Shao et al., 2024). While most high-performing LLMs use an auto-regressive modeling structure, generating tokens one after another, this sequential generation process can underutilize the parallelism of modern hardware and create an important inference bottleneck (Leviathan et al., 2023; Zhang et al., 2023). Among the alternative approaches that have emerged, *diffusion* (Sohl-Dickstein et al., 2015; Song and Ermon, 2019; Ho et al., 2020) has gained traction because of its strong performance in other domains (Song et al., 2021a; Karras et al., 2022). This has motivated *diffusion language models* (DLMs) (Li et al., 2022; He et al., 2023; Nie et al., 2025b; Ye et al., 2025; Shi et al., 2024; Sahoo et al., 2024a; Ou et al., 2025), which generate text by iteratively converting noise or masks into tokens and can therefore 'generate' multiple tokens in parallel.

DLMs share a natural connection to masked language models (MLMs) such as BERT (Devlin et al., 2019): both train bidirectional denoisers on discrete masked inputs. The distinction is not architectural, as many recent masked-diffusion models retain bidirectional Transformer encoders (Nie et al., 2025b; Sahoo et al., 2024a), but rather how training is formulated. A classical MLM optimizes a fixed masking rate, so every minibatch repeatedly targets the same corruption level. This can make optimization stable by avoiding between-time-step variability, but it trains only one denoising subproblem. MDLMs instead optimize a weighted continuum of fixed-rate denoising objectives, which is better aligned with diffusion likelihood training but introduces time-step-dependent stochasticity.

This comparison motivates our method. If part of the appeal of fixed-rate MLM is that it concentrates optimization on a stable corruption level, then uniform MDLM training may be statistically inefficient

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

whenever the weighted denoising loss varies substantially across time-steps. Rather than abandoning the MDLM objective and choosing a single masking rate, we ask whether the same optimization benefit can be recovered through non-uniform time-step sampling. To address this, we introduce **No**rmalized **2**nd **Mo**ment **Aware**ness (`NoNoMoAware`), which maintains an unbiased risk estimator while reallocating samples toward high-second-moment regions of the corruption process. By validating our approach on multiple model backbones at different training stages (pre-training and post-training), we show that adaptive sampling can improve downstream performance relative to the traditional uniform sampling approach. Overall, our contributions are

- We formalize MDLM as a weighted continuum of fixed-rate MLM-style denoising risks, clarifying why fixed-rate MLM is optimization-friendly but targets a narrower objective.
- We show that uniform time-step sampling can be inefficient when the weighted denoising loss has non-uniform second moments across corruption levels.
- Using this analysis, we propose a variance-aware online adaptive sampling strategy that reallocates samples across time-steps while preserving unbiased estimation of the MDLM risk.
- We apply our method to train multiple backbone models during different language model training stages, validating our method through improvements on multiple and diverse downstream tasks.

2 Background and Preliminaries

First, denote scalar discrete variables with K categories as one-hot column vectors and $\mathcal{V} : \{\boldsymbol{x} \in \{0,1\}^K : \sum_{i=1}^K \boldsymbol{x}_i = 1\}$ as the set of all such vectors, and a sequence of discrete tokens as $\boldsymbol{X} = \left[\boldsymbol{x}^1, \ldots, \boldsymbol{x}^L\right]$, where $\boldsymbol{x}^\ell$ denotes the ℓ-th token. $\mathcal{V}$ is the vocabulary, K its size and each $\boldsymbol{x}^\ell$ is a one-hot vector corresponding to an element from $\mathcal{V}$.

Most state-of-the-art language models follow an *auto-regressive* paradigm, where the model parameterized by θ attempts to learn a probability function that predicts a token conditioned on the past, i.e. $f_\theta\left(\boldsymbol{x}^j \,|\, \boldsymbol{x}^{<j}\right)$. Over a sequence, the model learns using a cross-entropy loss over the predictions, i.e.

$$\mathcal{L}_{\text{CLM}} = \frac{1}{L} \sum_{\ell=1}^{L} -\log \left[f_\theta\left(\boldsymbol{x}^{<\ell}\right)\right]_{\boldsymbol{x}^\ell}, \tag{1}$$

where $f_\theta\left(\boldsymbol{x}^{<\ell}\right)$ is the model output (a discrete distribution over $\mathcal{V}$) by giving the first ℓ elements of $\boldsymbol{X}$ as input and $[\cdot]_{\boldsymbol{x}^\ell}$ indicates the probability of predicting token $\boldsymbol{x}^\ell$ at position ℓ. To compute the loss on all elements of the sequence in parallel, auto-regressive language models make use of causal masking with *attention* (Bahdanau et al., 2015; Vaswani et al., 2017; Brown et al., 2020). However, this dependency between a token and all elements before it leads to a sequential bottleneck at inference, where only a single token can be generated at a time.

Another class of models are *masked language models* (MLMs), which eliminate the causal mask and instead have access to both past and future tokens at any position. The learning objective changes as well: during training, $\boldsymbol{X}$ is modified such that a subset of indices, $\mathcal{I} \in [L]$, is replaced with a special `[MASK]` token, creating a masked/noisy input $\tilde{\boldsymbol{X}}$. The model learns, given $\tilde{\boldsymbol{X}}$, to correctly predict the original tokens in the masked positions, i.e.

$$\mathcal{L}_{\text{MLM}} = \frac{1}{|\mathcal{I}|} \sum_{\ell \in \mathcal{I}} -\log \left(\left[g_\theta\left(\tilde{\boldsymbol{X}}\right)\right]_{\boldsymbol{x}^\ell}\right), \tag{2}$$

where g produces a probability distribution over the vocabulary corresponding to what the model believes the correct element should be at each position, acting as a denoising model or text discriminator (Goodfellow et al., 2014; Clark et al., 2020). The masking rate in this objective is typically fixed in advance, such as the original 15% rate in BERT (Devlin et al., 2019), though later work has shown that the choice of masking rate can materially affect learning (Liu et al., 2019; Wettig et al., 2023). This makes MLM an important reference point for MDLM: fixing the masking rate removes between-rate variation during training, but it also turns the corruption level into a chosen hyperparameter rather than a Monte Carlo estimate of a diffusion objective over all corruption levels.

Discrete diffusion models have been introduced to apply diffusion processes (Ho et al., 2020; Sohl-Dickstein et al., 2015; Song et al., 2021b; Wang et al., 2023; Sahoo et al., 2024b) on discrete structures,

where *masked diffusion* (Nie et al., 2025a; Shi et al., 2024; He et al., 2023) has consistently achieved greater performance. To understand masked diffusion, we re-use our notation from auto-regressive language modeling. Further denote $\mathrm{Cat}(\cdot; \boldsymbol{\pi})$ as the categorical distribution over the K classes given by $\boldsymbol{\pi} \in \Delta^K$ (the K-simplex). We assume the K-th category to correspond to a special `[MASK]` token and $\boldsymbol{m} \in \mathcal{V}$ be its one-hot vector. From before, the marginal of $\mathbf{z}_t$ conditioned on $\boldsymbol{x}$ at time t is

$$q(\mathbf{z}_t|\boldsymbol{x}) = \mathrm{Cat}(\mathbf{z}; \alpha_t \boldsymbol{x} + (1 - \alpha_t)\, \boldsymbol{\pi})\,,$$

where $\alpha_t \equiv \alpha(t) \in [0, 1]$ is a strictly decreasing function with $\alpha_0 \approx 1$ and $\alpha_1 \approx 0$. At each noising step t, the input $\boldsymbol{x}$ transitions to a masked state $\boldsymbol{m}$ with some probability. When $\boldsymbol{x}$ transitions to $\boldsymbol{m}$ at time t', for any time $t > t'$ it remains in that state. Applied to language modeling over sequences $\boldsymbol{X}$ from a vocabulary $\mathcal{V}$, a common implicit assumption is that the forward noising process is applied independently across the sequence and the denoising process factorizes independently across tokens conditioned on the latents $\mathbf{Z}_t = \left[\mathrm{z}_t^1, \ldots, \mathrm{z}_t^L\right]$ of each token. Thus a single model can be used to denoise the input as $\boldsymbol{x}_\theta^\ell(\mathbf{Z}_t, t)$ for each ℓ from a masked sequence $\mathbf{Z}$, optimizing

$$\mathcal{L}_{\text{NELBO}}^{\infty} = \mathbb{E}_q \left[\int_0^1 \frac{\alpha_t'}{1 - \alpha_t} \sum_{\ell=1}^{L} \log \left\langle \boldsymbol{x}_\theta^\ell(\mathbf{Z}_t, t)\,, \boldsymbol{x}^\ell \right\rangle \mathrm{d}t \right]. \tag{3}$$

This process, *masked diffusion language modeling* (MDLM), can be interpreted as a weighted average over fixed-rate MLM-style losses (Devlin et al., 2019; Warner et al., 2025). For unmasked tokens, the $\log \left\langle \boldsymbol{x}_\theta^\ell(\mathbf{Z}_t, t)\,, \boldsymbol{x}^\ell \right\rangle$ term reduces to zero so they do not contribute to the loss. Unlike fixed-rate MLM, MDLM traverses the full corruption continuum by sampling t, often uniformly within the range $[0, 1]$, leading to an ELBO formulation of

$$\mathcal{L}_{\text{MDLM}} = \mathbb{E} \left[\int_0^1 \frac{1}{t} \sum_{\ell=1}^{L} \log \left\langle \boldsymbol{x}_\theta^\ell(\mathbf{Z}_t, t)\,, \boldsymbol{x}^\ell \right\rangle \mathrm{d}t \right]. \tag{4}$$

3 Motivation and Analysis

To understand the effectiveness of MDLM, we study it as risk minimization over corruption levels. Let $\alpha : [0, 1] \to [0, 1]$ be a monotonically decreasing schedule on t with ELBO weight

$$\lambda(t) := -\alpha'(t) / \left(1 - \alpha(t)\right).$$

Using the standard linear schedule $\alpha(t) = 1 - t$ (Sahoo et al., 2024a; Nie et al., 2025a). Define

$$h_\theta(\boldsymbol{X}, t) := \mathbb{E}_{\tilde{\boldsymbol{X}} \sim q(\cdot|\boldsymbol{X}, t)} \left[\sum_{\ell=1}^{L} \mathbb{1}\left(\tilde{\boldsymbol{x}}^\ell = \boldsymbol{m}\right) \cdot \mathcal{L}\Big(\boldsymbol{x}^\ell, p_\theta(\tilde{\boldsymbol{X}}, \ell)\Big) \right]. \tag{5}$$

Then the true MDLM risk over a training distribution (or dataset) $\mathcal{D}$ can be written as

$$\mathcal{R}(\theta) = \mathbb{E}_{\boldsymbol{X} \sim \mathcal{D}} \left[\int_0^1 \lambda(t) \cdot h_\theta(\boldsymbol{X}, t)\, \mathrm{d}t \right]. \tag{6}$$

Equivalently, if $t \sim \mathrm{Unif}[0, 1]$, then $\mathcal{R}(\theta) = \mathbb{E}_{\boldsymbol{X}, t}[\lambda(t)\, h_\theta(\boldsymbol{X}, t)]$, making explicit that MDLM is a weighted mixture of denoising objectives indexed by the corruption level, consistent with the view of diffusion training as optimizing a family of denoising tasks across levels (Kingma and Gao, 2023).

Assumption 3.1 (One-step hierarchical sampling). Fix a proposal density p for the current estimator. $\{(\boldsymbol{X}_i, t_i)\}_i^N$ are sampled independently from $\mathcal{D}$ and p respectively, with corrupted inputs $\{\tilde{\boldsymbol{X}}_i\}_{i=1}^N$ independently sampled as $\tilde{\boldsymbol{X}}_i \sim q\left(\cdot|\boldsymbol{X}_i, t_i\right)$. If a proposal is adapted online, the same assumption is understood conditionally on the pre-batch history that determines the current proposal.

Lemma 3.2 (Conditional i.i.d. reduction). *Under Assumption 3.1, the triples $Z_i = (\boldsymbol{X}_i, t_i, \tilde{\boldsymbol{X}}_i)$ are i.i.d. for the current estimator. More generally, if a proposal $\hat{p}_k$ is chosen from past history $\mathcal{F}_{k-1}$, then conditional on $\mathcal{F}_{k-1}$ the current-step triples $Z_{k,i} = (\boldsymbol{X}_{k,i}, t_{k,i}, \tilde{\boldsymbol{X}}_{k,i})$ are i.i.d.*

This also clarifies the relation between MDLM and MLM approaches: by replacing the continuum in Equation 6 with a chosen t_0, the inter-time-step component of the stochastic objective is removed, changing the target risk from the MDLM mixture to a single denoising subproblem. Viewing the empirical MDLM and MLM objectives as different risk estimators, we state the following.

Proposition 3.3. *For each fixed $t_0 \in (0,1)$, the integrand of the MDLM risk at t_0 is the corresponding fixed-rate MLM-style denoising risk $\mathcal{R}_{\mathrm{MLM}}(\theta; t_0)$ multiplied by the non-negative ELBO weight $\lambda(t_0)$. As such, MDLM optimizes a non-negative mixture of MLM-style sub-objectives indexed by t.*

Proposition 3.4. *The loss-weighted empirical MDLM estimator is unbiased for the risk $\mathcal{R}(\theta)$.*

Proposition 3.5. *The corresponding unweighted MLM-style estimator targets a different risk and is therefore, in general, biased for $\mathcal{R}(\theta)$ whenever $\lambda(t)$ is non-constant.*

Remark 3.6. The role of the MDLM weight $\lambda(t)$ is therefore statistical as well as variational: once a masking schedule is fixed, the correct importance weight recovers the desired ELBO risk, whereas removing that weight changes the target objective.

From this perspective, we can also characterize its stochastic variability.

Lemma 3.7. *Under Assumption 3.1 with uniform time-step sampling, let $Z = (\boldsymbol{X}, t, \tilde{\boldsymbol{X}})$ with $\boldsymbol{X} \sim \mathcal{D}$, $t \sim \mathrm{Unif}[0,1]$, and $\tilde{\boldsymbol{X}} \sim q(\cdot|\boldsymbol{X}, t)$. Define*

$$g_\theta(Z) := \sum_{\ell=1}^{L} \mathbb{1}(\tilde{\boldsymbol{x}}^\ell = \boldsymbol{m}) \cdot \mathcal{L}\Big(\boldsymbol{x}^\ell, p_\theta(\tilde{\boldsymbol{X}}, \ell)\Big), \qquad Y_\theta(Z) := \lambda(t) \cdot g_\theta(Z),$$
$$m_\theta(t) := \mathbb{E}[Y_\theta(Z) \mid t], \qquad V_\theta(t) := \mathbb{E}\Big[Y_\theta(Z)^2 \mid t\Big]. \tag{7}$$

Then

$$\mathrm{Var}\Big(\widehat{\mathcal{R}}_N(\theta)\Big) = \frac{1}{N}\left[\int_0^1 V_\theta(t)\ \mathrm{d}t - \mathcal{R}(\theta)^2\right] = \frac{1}{N}\left[\mathbb{E}_t[\mathrm{Var}(Y_\theta(Z) \mid t)] + \mathrm{Var}_t(m_\theta(t))\right]. \tag{8}$$

Likewise,

$$m_\theta^{\mathrm{unw}}(t) := \mathbb{E}[g_\theta(Z)|t]\,,\ V_\theta^{\mathrm{unw}}(t) := \mathbb{E}\Big[g_\theta(Z)^2\Big|t\Big]$$
$$\Longrightarrow \mathrm{Var}\Big(\widehat{\mathcal{R}}_N^{\mathrm{unw}}(\theta)\Big) = \frac{1}{N}\left[\int_0^1 V_\theta^{\mathrm{unw}}(t)\ \mathrm{d}t - \mathcal{R}_{\mathrm{unw}}(\theta)^2\right]. \tag{9}$$

Lemma 3.7 shows that the relevant source of estimator noise is the heterogeneity of the weighted denoising loss across time-steps. The first term in Equation 8 captures within-time-step stochasticity from data and mask sampling, while the second term captures between-time-step mismatch in expected difficulty. A fixed-rate MLM removes this second term by conditioning on one t_0, but does so by changing the objective. $\lambda(t) = 1/t$ amplifies some regions more than others, so a highly non-uniform second-moment profile $V_\theta(t)$ can lead to noisy gradients and inefficient optimization.

4 NoNoMoAware: Non-Uniform Time-step Sampling for MDLM

The previous analysis presents a dilemma. Fixed-rate MLM-style training can be stable because it repeatedly optimizes the same corruption level, but it changes the target risk. Standard MDLM training preserves the target ELBO risk, but its stochastic loss need not be equally variable across time-steps. Diffusion training can be viewed as optimizing a continuum of coupled denoising sub-objectives indexed by t (Kingma and Gao, 2023). Uniform sampling allocates the same number of training examples to every value of t, regardless of how much that region contributes to the variance of the risk estimator. This is precisely the setting in which importance sampling can improve optimization by reallocating effort toward high-variance regions while preserving unbiasedness (Katharopoulos and Fleuret, 2018). Because the proposal is updated online, the clean formal object is the current-step estimator conditional on the pre-batch history that determines the proposal. In this section, we formalize this idea and use it to motivate a non-uniform time-step sampling strategy for MDLMs.

4.1 Optimization Stability

We again retain $\alpha(t) = 1 - t$ for $t \in [0,1]$. Define

$$\ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t) := \lambda(t) \sum_{\ell=1}^{L} \mathbb{1}(\tilde{\boldsymbol{x}}^\ell = \boldsymbol{m}) \cdot \mathcal{L}\Big(\boldsymbol{x}^\ell, p_\theta(\tilde{\boldsymbol{X}}, \ell)\Big), \tag{10}$$

as the weighted per-example MDLM loss so that, under uniform time-step sampling,

$$\mathcal{R}(\theta) = \int_0^1 \mathbb{E}_{\boldsymbol{X}\sim\mathcal{D},\tilde{\boldsymbol{X}}\sim q(\cdot|\boldsymbol{X},t)} \left[\ell_\theta(\boldsymbol{X},\tilde{\boldsymbol{X}},t)\right] \mathrm{d}t. \tag{11}$$

Assumption 4.1 (Finite conditional second moments). $V_\theta : (0,1) \to [0,\infty)$, defined by

$$V_\theta(t) := \mathbb{E}_{\boldsymbol{X}\sim\mathcal{D},\tilde{\boldsymbol{X}}\sim q(\cdot|\boldsymbol{X},t)} \left[\ell_\theta(\boldsymbol{X},\tilde{\boldsymbol{X}},t)^2\right] \tag{12}$$

with $\int_0^1 V_\theta(t)\,\mathrm{d}t < \infty$, $\int_0^1 \sqrt{V_\theta(t)}\mathrm{d}t < \infty$, is measurable.

$V_\theta(t)$, the timestep-wise second moment of the weighted denoising loss, aggregates randomness from data sampling, masking, and prediction uncertainty at corruption level t. We call a density p on $[0,1]$ *admissible* if $p(t) > 0$ whenever $V_\theta(t) > 0$ and $\int_0^1 V_\theta(t)\,/p(t)\ \mathrm{d}t < \infty$. For highly non-uniform $V_\theta(t)$, uniform time-step sampling under-samples the most variable denoising sub-objectives.

Lemma 4.2. *Let p be any admissible proposal density on $[0,1]$, and define the estimator*

$$\widehat{\mathcal{R}}_{N,p}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{\ell_\theta(\boldsymbol{X}_i,\tilde{\boldsymbol{X}}_i,t_i)}{p(t_i)}, \tag{13}$$

where $\boldsymbol{X}_i \sim \mathcal{D}, t_i \sim p, \tilde{\boldsymbol{X}}_i \sim q\left(\cdot|\boldsymbol{X}_i,t_i\right)$. Then

$$\mathbb{E}\left[\widehat{\mathcal{R}}_{N,p}(\theta)\right] = \mathcal{R}(\theta)\,, \mathrm{Var}\left(\widehat{\mathcal{R}}_{N,p}(\theta)\right) = \frac{1}{N}\left[\int_0^1 \frac{V_\theta(t)}{p(t)}\mathrm{d}t - \mathcal{R}(\theta)^2\right]. \tag{14}$$

if Assumption 3.1 holds for the current minibatch with proposal p.

If the proposal $\hat{p}_k$ at step k is chosen from the pre-batch history $\mathcal{F}_{k-1}$, then provided each current-step triple $\left(\boldsymbol{X}_{k,i},t_{k,i},\tilde{\boldsymbol{X}}_{k,i}\right)$ has conditional marginal distribution $\boldsymbol{X}_{k,i}\sim\mathcal{D}$, $t_{k,i}\sim\hat{p}_k$, and $\tilde{\boldsymbol{X}}_{k,i}\sim q(\cdot\mid \boldsymbol{X}_{k,i},t_{k,i})$, the expectation identity remains valid conditional on $\mathcal{F}_{k-1}$. If the conditional version of Assumption 3.1 also holds, then the same variance formula holds conditional on $\mathcal{F}_{k-1}$ with p replaced by $\hat{p}_k$. In particular, adaptive proposals remain unbiased for the same MDLM objective.

Theorem 4.3. *Under Assumption 4.1 and 3.1, the variance in Equation 14 is minimized as*

$$p^\star_\theta(t) = \sqrt{V_\theta(t)}\Big/\int_0^1 \sqrt{V_\theta(u)}\,\mathrm{d}u, \qquad \mathrm{Var}\left(\widehat{\mathcal{R}}_{N,p^\star_\theta}(\theta)\right) = \frac{1}{N}\left[\left(\int_0^1 \sqrt{V_\theta(t)}\,\mathrm{d}t\right)^2 - \mathcal{R}(\theta)^2\right]. \tag{15}$$

Corollary 4.4. *Under Assumption 4.1 and 3.1, Let $p_{\mathrm{uni}}(t) = 1$ be the uniform density on $[0,1]$. Then*

$$\mathrm{Var}\left(\widehat{\mathcal{R}}_{N,p^\star_\theta}(\theta)\right) \le \mathrm{Var}\left(\widehat{\mathcal{R}}_{N,p_{\mathrm{uni}}}(\theta)\right), \tag{16}$$

with equality if and only if $V_\theta(t)$ is constant almost everywhere.

These show that for non-uniform $V_\theta(t)$, i.e. some corruption levels induce larger stochastic losses than others, allocating samples to those regions reduces variance but leaves target risk unchanged.

Assumption 4.5 (Gradient-loss coupling). There exists $C_{\mathrm{grad}} > 0$ such that for all $t \in [0,1]$,

$$\mathbb{E}_{\boldsymbol{X},\tilde{\boldsymbol{X}}|t}\left[\left\|\nabla_\theta\,\ell_\theta(\boldsymbol{X},\tilde{\boldsymbol{X}},t)\right\|_2^2\right] \le C_{\mathrm{grad}}\cdot V_\theta(t)\,. \tag{17}$$

Proposition 4.6. *Under Assumption 3.1, assume differentiation and expectation can be interchanged for the weighted loss. For any admissible proposal density p, the gradient estimator induced by $\widehat{\mathcal{R}}_{N,p}$ is unbiased, and under Assumption 4.5 satisfies*

$$\mathbb{E}\left[\left\|\nabla_\theta\,\widehat{\mathcal{R}}_{N,p}(\theta) - \nabla_\theta\,\mathcal{R}(\theta)\right\|_2^2\right] \le C_{\mathrm{grad}}/N\int_0^1 V_\theta(t)/p(t)\mathrm{d}t. \tag{18}$$

Reducing $\int_0^1 V_\theta(t)\,/p(t)\ \mathrm{d}t$ thus not only lowers the scalar risk estimator variance, but also a gradient noise upper bound. Consequently, a good proposal should sample more from time-steps where weighted denoising losses exhibit large second moments. We justify this in Appendix A.10.

4.2 Beta Sampling for Step Reduction

The exact optimum $p_\theta^\star$ in Equation 15 depends on the current model state and is costly to estimate. We therefore seek a low-dimensional surrogate on $[0,1]$ flexible enough to emphasize different parts of the corruption process and for which sampling and re-weighting remain simple.

Assumption 4.7 (Boundary-aware power-law second moment). For the current model state, we model the second moment profile by the boundary-aware surrogate

$$V_\theta(t) \approx C \cdot t^a (1-t)^b, \qquad C > 0, \;\; a, b > -2. \tag{19}$$

This is motivated by two recurring observations in literature. *(1)* Time-step effects are often organized through low-dimensional noise-level or log-SNR parameterizations, making scalar laws across t effective in practice (Karras et al., 2022; Kingma and Gao, 2023; Hang and Gu, 2024). *(2)* Work on non-uniform timestep weighting and sampling reports timestep importance to be highly non-uniform and often concentrated toward boundary regimes, which families such as Beta or SNR-based rules can capture (Hang et al., 2023; Lee et al., 2025; Kim et al., 2025). $a, b > -2$ ensures that the induced proposal is normalizable and includes $a = b = 0$ as a degenerate special case.

Proposition 4.8. *Under Theorem 4.3 and Assumption 4.7,*

$$p_\theta^\star(t) \propto t^{a/2}(1-t)^{b/2} \approx \mathrm{Beta}(t; \mu, \nu), \text{ where } \mu = 1 + a/2, \nu = 1 + b/2. \tag{20}$$

Proposition 4.8 gives a direct justification for our parameterization. Under a simple structural approximation of the timestep-wise second moment, it is the natural low-parameter approximation to the variance-optimal distribution. Moreover, the parameters (μ, ν) have an interpretable effect: $\mu < 1$ emphasizes smaller t, $\nu < 1$ emphasizes larger t, and $\mu, \nu > 1$ concentrate mass in the interior.

4.3 Technical Considerations

Because computing $V_\theta(t)$ exactly for every $t \in [0,1]$ is infeasible, we estimate it online during training (Figure 1). We partition the interval $[0,1]$ into T evenly sized bins and maintain a running estimate of the weighted loss second moment in each bin. For a training example with sampled time-step t in bin j, we observe the weighted loss $\ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)$ from Equation 10 and update

$$\widehat{V}_j \leftarrow (1-\rho) \cdot \widehat{V}_j + \rho \cdot \ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)^2, \tag{21}$$

where $\rho \in (0,1]$ is a smoothing coefficient. This produces a cheap online approximation to the timestep-wise second moment profile. We fit the log-linear model

$$\log \widehat{V}(t) \approx c + a \log(t) + b \log(1-t), \tag{22}$$

where t is the bin center. Once $(\hat{a}, \hat{b})$ are estimated, we set $\hat{\mu} = 1 + \hat{a}/2$, $\hat{\nu} = 1 + \hat{b}/2$, and clip them to a small positive minimum to preserve full support on $(0,1)$. We then sample $t_i \sim \hat{p} = \mathrm{Beta}(\cdot; \hat{\mu}, \hat{\nu})$ for the next batch, re-weighting each example to ensure un-biasedness relative to Equation 3

$$\widehat{\ell}_i = \ell_\theta(\boldsymbol{X}_i, \tilde{\boldsymbol{X}}_i, t_i) / \hat{p}(t_i). \tag{23}$$

Since the target distribution in MDLM is uniform on $[0,1]$, Equation 23 is the complete correction term. In particular, conditional on the fitted proposal $\hat{p}$ used for a given batch, Lemma 4.2 implies that the resulting risk estimator remains unbiased for the same MDLM objective.

We refer to this adaptive method as **No**rmalized 2nd **Mo**ment **Aware**ness (**No²MoAware**/**N₂M**[1]): as $\widehat{V}(t)$ evolves during training, $(\hat{\mu}, \hat{\nu})$ shifts sampling mass toward levels with greater stochastic variability. This provides a practical approximation to the optimal proposal in Theorem 4.3 with small overhead, since only scalar loss statistics and distribution parameters need to be maintained online.

5 Experimental Results

5.1 Initial Tuning

We initially conduct an initial experiment on a smaller sized MDLM in order to determine appropriate hyper-parameters to set for our model. In particular, we use a **LLaDA** (Nie et al., 2025b) model with

[1] Read as NoNoMoAware, a reference to concept of *mono no aware*, i.e. the appreciation of impermanence.

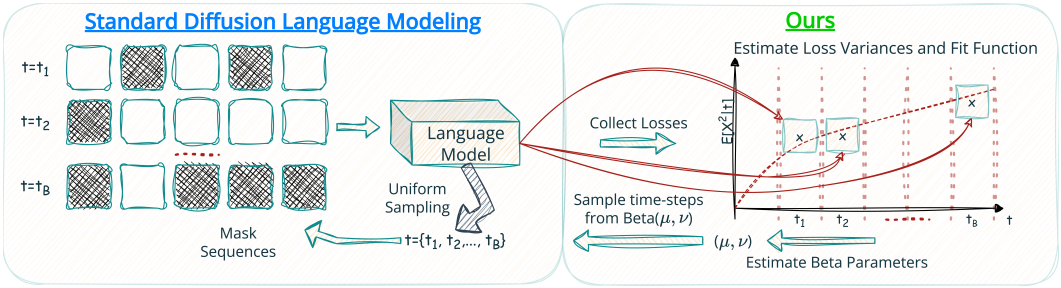

Figure 1: We use sample losses to estimate the 2nd moment as a function of t, then fit it to a (log-)linear model and Beta distribution (Equation 20) used to sample time-steps for the next batch.

approximately 400M non-embedding parameters trained on 10B tokens from Li et al. (2024) with a context length of 2048 tokens. We use AdamW with a peak learning rate of 1e-4, weight decay of 0.1, and gradient clipping of 1.0. We use a cosine annealing schedule with a warm-up period of ≈100M tokens and a total batch size of 0.5M tokens. We run each experiment with 3 seeds.

We set an initial version where we use $T = 10$ bins, $\rho = 0.9$, $a, b > 1e-3$ and fit after each step, comparing against alternative hyper-parameter choices through performance on downstream tasks.

Table 1: Hyperparameter sensitivity check on a 400M LLaDA model.

Strategy	MMLU acc	BBH acc	Hella. acc	ARC-c acc	TQA acc	Wino. acc	PIQA acc	GSM8K acc	Math acc	GPQA acc	HE pass@1	MBPP pass@1	Avg.
Uniform	$46.35_{\pm 0.82}$	$35.25_{\pm 1.53}$	$34.13_{\pm 0.54}$	$52.65_{\pm 0.61}$	$35.42_{\pm 0.55}$	$55.43_{\pm 0.76}$	$52.35_{\pm 0.54}$	$53.32_{\pm 1.45}$	$20.43_{\pm 1.43}$	$29.34_{\pm 1.92}$	$29.43_{\pm 1.53}$	$31.85_{\pm 1.65}$	39.66
N²M (*Ours*)	$47.47_{\pm 0.97}$	$35.46_{\pm 1.12}$	$33.41_{\pm 0.38}$	$52.45_{\pm 0.57}$	$34.34_{\pm 0.72}$	$54.26_{\pm 0.59}$	$52.23_{\pm 0.52}$	$53.62_{\pm 1.43}$	$21.34_{\pm 1.24}$	$29.96_{\pm 2.31}$	$29.94_{\pm 1.42}$	$31.94_{\pm 1.45}$	39.70
+ ($T = 20$)	$48.45_{\pm 1.03}$	$35.65_{\pm 1.34}$	$33.91_{\pm 0.42}$	$53.25_{\pm 0.53}$	$36.74_{\pm 0.68}$	$54.95_{\pm 0.65}$	$52.53_{\pm 0.62}$	$54.72_{\pm 1.24}$	$22.14_{\pm 1.24}$	$30.53_{\pm 2.54}$	$30.54_{\pm 1.56}$	$32.34_{\pm 1.34}$	40.48
+ ($T = 50$)	$48.96_{\pm 1.11}$	$35.71_{\pm 1.54}$	$33.96_{\pm 0.54}$	$53.31_{\pm 0.61}$	$36.84_{\pm 0.84}$	$55.11_{\pm 0.76}$	$52.57_{\pm 0.58}$	$54.88_{\pm 1.21}$	$22.19_{\pm 1.22}$	$30.65_{\pm 2.56}$	$30.49_{\pm 1.60}$	$32.32_{\pm 1.54}$	40.58
+ ($\rho = 0.7$)	$47.54_{\pm 1.34}$	$33.75_{\pm 1.54}$	$32.12_{\pm 0.56}$	$51.75_{\pm 0.45}$	$35.64_{\pm 0.47}$	$53.37_{\pm 0.76}$	$52.23_{\pm 0.64}$	$54.14_{\pm 1.23}$	$20.86_{\pm 1.32}$	$29.46_{\pm 2.36}$	$28.65_{\pm 1.76}$	$31.46_{\pm 1.23}$	39.25
+ ($a, b > 1e-1$)	$48.89_{\pm 1.33}$	$35.35_{\pm 1.62}$	$33.43_{\pm 0.64}$	$53.37_{\pm 0.75}$	$36.78_{\pm 0.86}$	$54.23_{\pm 0.78}$	$52.12_{\pm 0.85}$	$54.65_{\pm 1.65}$	$22.64_{\pm 1.84}$	$30.13_{\pm 2.85}$	$30.12_{\pm 1.84}$	$32.53_{\pm 1.74}$	40.35
+ (Fit 5 Steps)	$46.76_{\pm 1.15}$	$35.53_{\pm 1.64}$	$33.64_{\pm 0.53}$	$52.63_{\pm 0.45}$	$36.34_{\pm 0.79}$	$54.35_{\pm 0.90}$	$52.34_{\pm 0.94}$	$52.53_{\pm 1.93}$	$21.16_{\pm 1.23}$	$29.53_{\pm 2.45}$	$29.35_{\pm 1.95}$	$31.87_{\pm 1.78}$	39.67

Our observations are consistent with some intuition; more consistent updates leads to greater performance, likely due to more accurate loss statistics. Furthermore, more granularity (more bins) as well as greater distributional range (a, b) lead to some performance benefits as well.

5.2 Pre-Training

We use two types of models, **LLaDA** and **ModernBERT** (Warner et al., 2025), with the same general setup as the initial tuning. Following our initial investigation, we use 20 evenly sized intervals for fitting and $\rho = 0.9$. For computational costs, we only run each experiment with a single seed. We set hyper-parameters found based on our previous investigation. We compare against standard MLM with a masking rate of 50% (to preserve the same average number of masking tokens), MDLM with uniform time-step sampling and weighting, as well as without loss weighting.

Table 2: Comparison of our time-step sampling approach during pre-training against uniform sampling of t with and without re-weighting (MLM).

	Strategy	MMLU acc	BBH acc	Hella. acc	ARC-c acc	TQA acc	Wino. acc	PIQA acc	GSM8K acc	Math acc	GPQA acc	HE pass@1	MBPP pass@1	Avg.
LLaDA	MLM (50%)	46.24	41.49	57.97	61.08	56.81	56.30	54.86	53.14	22.39	24.97	30.46	32.64	44.86
	No Weighting	47.84	41.51	57.76	61.65	56.79	57.57	55.43	53.32	22.44	25.02	30.52	32.35	45.18
	Uniform	48.02	41.61	57.30	62.04	56.93	**57.90**	55.92	53.59	22.26	25.10	31.02	32.86	45.38
	N²M (*Ours*)	**49.38**	**42.34**	**58.34**	**62.64**	**56.97**	57.86	**56.04**	**54.05**	**22.86**	**25.93**	**31.22**	**33.17**	**45.90**
mBERT	MLM (50%)	45.34	40.14	52.66	60.17	56.97	52.96	55.32	53.21	22.16	25.10	28.34	29.42	43.48
	No Weighting	46.76	40.21	53.54	**61.26**	57.09	53.33	55.93	53.41	32.66	25.08	28.22	29.24	44.73
	Uniform	47.93	40.34	53.94	60.29	**57.16**	**53.80**	56.96	53.44	22.90	25.07	28.57	30.03	44.20
	N²M (*Ours*)	**48.37**	**41.29**	**54.59**	60.10	56.99	53.59	**57.29**	**53.84**	**23.12**	**25.26**	**29.02**	**30.53**	**44.50**

Table 2 shows that adaptive time-step sampling yields consistent gains after pre-training on a range of downstream tasks, especially on general reasoning and commonsense benchmarks. The gains are smaller on math and code tasks, which is unsurprising because those capabilities are less directly targeted by broad pre-training alone. The main takeaway is that replacing uniform time-step sampling improves the quality of the learned representation without changing the underlying MDLM objective.

5.3 Supervised Fine-Tuning

We next conduct supervised fine-tuning (SFT) experiments where examples consist of a prompt-response pair $\boldsymbol{x} = (\boldsymbol{p}, \boldsymbol{r})$. Similar to pre-training, tokens are selectively masked based on the noise schedule, but only in the response. We follow the setup by Lambert et al. (2024) and further add `UltraChat` (Ding et al., 2023) and `Alpaca` (Dubois et al., 2023) as tuning datasets. Due to the smaller size of `Alpaca`, we train for more epochs (5× more) when using it.

Table 3: Comparison of approaches during SFT.

		PT	SFT	MMLU	MMLU-Pro	Hella.	ARC-c	GPQA	GSM8K	Math	HE	MBPP	Avg.
UltraChat	LLaDA	Unif	Unif	49.46	36.43	52.68	63.58	24.85	66.48	26.58	37.93	33.47	43.50
		Unif	N²M	49.83	37.38	53.14	**63.83**	25.64	67.28	26.84	38.01	33.84	43.98
		N²M	N²M	**50.21**	**37.77**	**53.28**	63.76	**26.64**	**67.58**	**26.89**	**38.83**	**34.42**	44.38
	mBERT	Unif	Unif	48.38	35.02	**56.84**	62.57	25.20	64.28	24.38	37.84	33.29	43.09
		Unif	N²M	48.74	**35.33**	55.95	62.84	**26.28**	64.57	24.20	37.85	**33.51**	43.25
		N²M	N²M	**49.38**	35.28	56.60	**63.01**	26.02	**65.02**	**24.88**	**38.42**	33.48	43.57
Tulu3Mixture	LLaDA	Unif	Unif	51.53	37.58	64.98	64.49	25.34	67.83	27.92	**39.73**	35.92	46.15
		Unif	N²M	51.46	38.96	**64.64**	**65.92**	26.48	68.12	28.34	39.62	36.02	46.62
		N²M	N²M	**51.78**	**39.21**	66.38	65.38	**26.95**	**68.48**	**28.49**	39.72	**36.64**	47.00
	mBERT	Unif	Unif	48.58	**35.38**	67.85	**63.85**	25.75	67.34	27.32	39.72	35.86	45.74
		Unif	N²M	49.28	35.27	**68.55**	63.62	25.62	67.78	28.01	**39.84**	36.78	46.08
		N²M	N²M	**49.85**	35.25	67.72	63.48	**26.73**	**68.32**	**28.73**	39.57	**37.25**	46.32
Alpaca	LLaDA	Unif	Unif	48.56	34.29	68.28	62.48	25.38	64.28	22.85	32.34	32.79	43.47
		Unif	N²M	48.60	34.39	**68.78**	62.35	25.66	64.43	23.03	32.65	32.76	43.63
		N²M	N²M	**49.58**	**35.42**	68.68	**62.75**	**25.83**	**64.65**	**23.21**	**33.20**	**33.28**	44.07
	mBERT	Unif	Unif	48.02	**33.48**	64.02	60.38	24.84	64.49	23.08	29.12	30.34	41.97
		Unif	N²M	**48.46**	33.43	63.92	**60.84**	24.59	**64.62**	**23.54**	29.32	30.91	42.18
		N²M	N²M	48.11	34.47	**64.11**	60.34	**25.76**	64.58	23.52	**30.04**	**31.23**	42.46

Table 3 shows that adaptive sampling remains helpful during post-training. Using it during either pre-training or SFT typically improves downstream performance, and applying it in both stages is often strongest overall. Even on the smaller `Alpaca` setting, where limited data constrains absolute gains, the adaptive strategy remains competitive across both `LLaDA` and `ModernBERT`.

5.4 Additional Analysis

Different t for each token. Since standard MDLM uses a single time-step t for the full sequence. We test whether assigning a separate t to each token is beneficial. Table 4 shows that per-token time-steps are consistently worse, both for uniform sampling and for our adaptive strategy. A likely reason is that token-wise variation introduces additional heterogeneity into the ELBO weights, making the training signal noisier without providing a corresponding benefit.

Table 4: Varying t for each token.

	MMLU	Hella.	ARC-c	GPQA
Uniform	48.42	57.70	62.44	25.10
+ Per Token Rate	46.31	56.68	61.47	24.95
N²M (*Ours*)	49.38	58.34	62.64	25.93
+ Per Token Rate	46.75	57.39	61.68	25.58

Fixing Beta**.** We also test whether the gains relate to using a non-uniform distribution, without adaptation. Table 5 shows that fixed Beta distributions can underperform uniform sampling, matches intuition in Section 4. The useful proposal aligns with the *current* profile $V_\theta(t)$, while a static choice can over-concentrate on the wrong corruption levels and waste training effort.

Table 5: Adaptive and fixed Beta.

Distribution	MMLU	Hella.	ARC-c	GPQA
$\mathrm{Beta}(0.5, 0.5)$	47.94	58.13	62.85	25.73
$\mathrm{Beta}(2, 2)$	46.14	58.07	60.98	25.43
$\mathrm{Beta}(5, 1)$	48.95	57.68	62.53	25.79
$\mathrm{Beta}(2, 5)$	47.85	56.84	62.58	25.39
N²M (*Ours*)	29.38	38.34	32.64	25.93

Table 6: Varying fitting function.

Function	MMLU	Hella.	ARC-c	GPQA
Linear	43.34	45.39	54.29	25.09
Quadratic	44.85	46.38	54.48	25.14
Log-Linear	49.38	58.34	62.64	25.93
Log-Quadratic	49.42	58.22	62.58	25.87

Determining the proper fit. Another design choice is how to fit the Beta parameters from the estimated second-moment profile. Table 6 shows that log-linear and log-quadratic fitting clearly outperform direct polynomial fits. This is consistent with Section 4: the Beta family becomes linear in the basis $\{\log t, \log(1-t)\}$, whereas direct polynomial fits need not respect positivity or boundary behavior and can therefore yield unstable proposals.

Change in estimated Beta function. It is also useful to understand how the learned proposal evolves during training. Figure 2 shows a clear pattern: early in training, the fitted proposal is close to uniform, which is consistent with broad uncertainty across all corruption levels. As training proceeds,

probability mass shifts toward larger t, indicating that heavily corrupted examples dominate the weighted loss second moment due to greater variability within samples at intermediate stages. Later, the proposal stabilizes with a preference for higher-noise regimes, but avoids extremely corrupted examples due to a likely inability to further optimize for such examples. This bears close resemblance

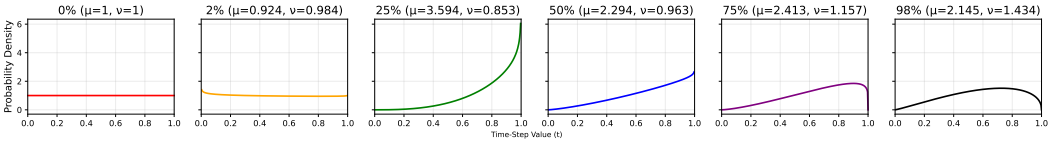

Figure 2: Beta evolution over training; see Figure 3 for individual variance values.

to related work on curriculum learning (Bengio et al., 2009) as well as noisy label learning (Yao et al., 2020; Yuan et al., 2023; 2025). In particular, such works often observe a necessity in progressively learning in a easy-to-difficult approach (motivated by psychology) while also identifying noisy or unlearn-able examples. Figure 2 shows a similar trajectory in our setting; in particular, the models are prioritized to learn at lower masking rates initially, but at later stages they also reject highly noised samples likely due to a realization in the lack of useful learning signals from such models due to generally low variance in individual token-wise losses.

6 Related Work

Non-uniform Time-Step Sampling. Most diffusion training pipelines still use uniform time-step sampling. Recent work in continuous image diffusion has begun to question that choice: Lee et al. (2025) transform uniform samples through the CDF of a fixed Beta distribution, while Kim et al. (2025) learn an adaptive sampler with gradient-based updates. Other approaches use heuristic or re-sampling rules to accelerate learning (Wang et al., 2025; Hang et al., 2023; Choi et al., 2022). Our setting differs in two ways, where we study masked diffusion for language and build the proposal directly from the timestep-wise second moment of the weighted denoising loss, which keeps the resulting estimator aligned with the same ELBO objective.

Optimal Masking Rates. BERT (Devlin et al., 2019) used a 15% fixed masking rate across all samples. RoBERTa (Liu et al., 2019) improved on this with a 30% rate, leading to various questions regarding exactly how much masking was necessary. Wettig et al. (2023) conducted a larger scale study showing that even higher masking rates ($> 50\%$) could retain comparable performance without changes in learning efficiency. Vision Transformers (ViTs) (Dosovitskiy et al., 2021) meanwhile use much larger masking rates ($\approx 80\%$) for images, highlighting how large rates could be better for robustness in transformer model training. Zhang et al. (2025) relate this to a possible need for aggressive corruption in order to properly learn intra-class features.

7 Conclusion

We studied masked diffusion language modeling from the perspective of risk estimation across a continuum of corruption levels. This viewpoint clarifies two points. First, MDLM is a weighted mixture of fixed-rate MLM-style denoising objectives, so fixed-rate training can reduce between-time-step variation but targets a narrower risk. Second, under the standard linear schedule, uniform time-step sampling is not statistically neutral: when the weighted loss second moment varies substantially with t, uniform sampling yields unnecessarily noisy risk and gradient estimates. Motivated by this analysis, we proposed an adaptive non-uniform time-step sampling strategy that preserves unbiasedness through importance weighting while reallocating effort toward high-second-moment regions. The resulting Beta-based parameterization is both theoretically motivated and practical to fit online. Empirically, it improves pre-training and supervised fine-tuning performance across multiple backbones and datasets, while introducing only modest computational overhead.

8 Limitations, Ethics And Impact

This work studies a training-time sampling strategy for masked diffusion language models. While improving the efficiency and downstream quality of generative language models can indirectly contribute to the same broad risks associated with capable language models, such as misuse or unreliable generated content, these risks do not stem specifically from the sampling estimator proposed here. The method changes how training time-steps are allocated, not the deployment setting, data source, or safety behavior of the resulting model. While our method improves the general performance of such models, the ethical considerations are largely consistent with those of general-purpose language modeling. Potential risks include misuse in generating harmful or misleading content, reinforcement of biases present in training corpora, and environmental concerns arising from large-scale training, which are not specific to our work and thus unnecessary to analyze here.

The primary limitation of our current work is that the scale and types of models on which experiments are conducted may not represent the full spectrum of diffusion language models. However, our analysis is intended to be as general as possible for masked diffusion models, presenting a potential avenue for use in other such models. Another limitation is that the Beta proposal is a low-dimensional approximation to the variance-optimal density. If the true second-moment profile is highly multi-modal or changes too quickly for the online estimates to track, the method should remain unbiased under the importance correction, but the variance reduction may be smaller. A further limitation is the direct application of such methods to reinforcement learning and alternative alignment objectives, which often rely on scoring model generations rather than token-level denoising losses. Nevertheless, the importance of the pre-training and instruction-tuning stages means that the findings of this work can still be used to improve the models used in such settings.

9 Reproducibility and AI Use Statement

We include many details of our direct implementation in Section 5 and Appendix C. We further provide many components of our code in the supplementary material. We commit to ensuring that the code will be fully released upon acceptance of this work alongside a camera-ready version. Though some small variability may exist due to differences in the exact computational environment (GPUs, CPUs, etc.), we will further clarify the exact devices and environment used alongside the publicly available code.

The use of large language models was limited to LaTeX formatting as well as anonymization of possibly revealing details within the supplementary material (primarily code). Research design, execution, results collection and presentation were done by authors themselves. All technical content and claims are the responsibility of the authors.

References

J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. van den Berg. Structured denoising diffusion models in discrete state-spaces. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 17981–17993, 2021. URL https://openreview.net/forum?id=h7-XixPCAL.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In A. P. Danyluk, L. Bottou, and M. L. Littman, editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM, 2009. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

A. Campbell, J. Benton, V. D. Bortoli, T. Rainforth, G. Deligiannidis, and A. Doucet. A continuous time framework for discrete denoising models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL https://openreview.net/forum?id=DmT862YAieY.

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

J. Choi, J. Lee, C. Shin, S. Kim, H. Kim, and S. Yoon. Perception prioritized training of diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 11462–11471. IEEE, 2022. URL https://doi.org/10.1109/CVPR52688.2022.01118.

K. Clark, M. Luong, Q. V. Le, and C. D. Manning. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=r1xMH1BtvB.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/n19-1423.

S. Dieleman, L. Sartran, A. Roshannai, N. Savinov, Y. Ganin, P. H. Richemond, A. Doucet, R. Strudel, C. Dyer, C. Durkan, C. Hawthorne, R. Leblond, W. Grathwohl, and J. Adler. Continuous diffusion for categorical data, 2022. URL http://arxiv.org/abs/2211.15089.

N. Ding, Y. Chen, B. Xu, Y. Qin, S. Hu, Z. Liu, M. Sun, and B. Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.183. URL https://aclanthology.org/2023.emnlp-main.183/.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Y. Dubois, C. X. Li, R. Taori, T. Zhang, I. Gulrajani, J. Ba, C. Guestrin, P. Liang, and T. B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL https://openreview.net/forum?id=4hturzLcKX.

G. Folland. *Real analysis: modern techniques and their applications*. Pure and applied mathematics. Wiley, 2 edition, 1999. URL http://www.loc.gov/catdir/toc/onix03/98037260.html.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.

I. Gulrajani and T. B. Hashimoto. Likelihood-based diffusion language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL https://openreview.net/forum?id=e2MCL6hObn.

D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Xu, H. Ding, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Chen, J. Yuan, J. Tu, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. You, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Zhou, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, S. S. Li, S. Zhou, S. Wu, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. Q. Li, X. Jin, X. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Xiong, Y. Luo, Y. You, Y. Liu, Y. Zhou, Y. X. Zhu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu, Z. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, and Z. Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nat.*, 645(8081):633–638, 2025. doi: 10.1038/S41586-025-09422-Z. URL https://doi.org/10.1038/s41586-025-09422-z.

X. Han, S. Kumar, and Y. Tsvetkov. SSD-LM: semi-autoregressive simplex-based diffusion language model for text generation and modular control. In A. Rogers, J. L. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 11575–11596. Association for Computational Linguistics, 2023. URL https://doi.org/10.18653/v1/2023.acl-long.647.

T. Hang and S. Gu. Improved noise schedule for diffusion training, 2024. URL https://arxiv.org/abs/2407.03297.

T. Hang, S. Gu, C. Li, J. Bao, D. Chen, H. Hu, X. Geng, and B. Guo. Efficient diffusion training via min-snr weighting strategy. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 7407–7417. IEEE, 2023. URL https://doi.org/10.1109/ICCV51070.2023.00684.

Z. He, T. Sun, Q. Tang, K. Wang, X. Huang, and X. Qiu. Diffusionbert: Improving generative masked language models with diffusion models. In A. Rogers, J. L. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4521–4534. Association for Computational Linguistics, 2023. URL https://doi.org/10.18653/v1/2023.acl-long.248.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020,*

December 6-12, 2020, virtual, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL https://openreview.net/forum?id=k7FuTOWMOc7.

A. Katharopoulos and F. Fleuret. Not all samples are created equal: Deep learning with importance sampling. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2525–2534. PMLR, 2018. URL https://proceedings.mlr.press/v80/katharopoulos18a.html.

M. Kim, D. Ki, S. Shim, and B. Lee. Adaptive non-uniform timestep sampling for accelerating diffusion model training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 2513–2522. Computer Vision Foundation / IEEE, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Kim_Adaptive_Non-Uniform_Timestep_Sampling_for_Accelerating_Diffusion_Model_Training_CVPR_2025_paper.html.

D. P. Kingma and R. Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL https://openreview.net/forum?id=NnMEadcdyD.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

N. Lambert, J. Morrison, V. Pyatkin, S. Huang, H. Ivison, F. Brahman, L. J. V. Miranda, A. Liu, N. Dziri, X. Lyu, Y. Gu, S. Malik, V. Graf, J. D. Hwang, J. Yang, R. L. Bras, O. Tafjord, C. Wilhelm, L. Soldaini, N. A. Smith, Y. Wang, P. Dasigi, and H. Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. In *The Second Conference on Language Modeling, COLM 2025, Montreal, QC, Canada, October 7-10, 2025*. OpenReview.net, 2024. URL https://openreview.net/forum?id=i1uGbfHHpH.

H. Lee, H. Lee, S. Gye, and J. Kim. Beta sampling is all you need: Efficient image generation strategy for diffusion models using stepwise spectral analysis. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2025, Tucson, AZ, USA, February 26 - March 6, 2025*, pages 4215–4224. IEEE, 2025. URL https://doi.org/10.1109/WACV61041.2025.00414.

Y. Leviathan, M. Kalman, and Y. Matias. Fast inference from transformers via speculative decoding. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

J. Li, A. Fang, G. Smyrnis, M. Ivgi, M. Jordan, S. Y. Gadre, H. Bansal, E. K. Guha, S. S. Keh, K. Arora, S. Garg, R. Xin, N. Muennighoff, R. Heckel, J. Mercat, M. F. Chen, S. Gururangan, M. Wortsman, A. Albalak, Y. Bitton, M. Nezhurina, A. Abbas, C. Hsieh, D. Ghosh, J. Gardner, M. Kilian, H. Zhang, R. Shao, S. M. Pratt, S. Sanyal, G. Ilharco, G. Daras, K. Marathe, A. Gokaslan, J. Zhang, K. R. Chandu, T. Nguyen, I. Vasiljevic, S. M. Kakade, S. Song, S. Sanghavi, F. Faghri, S. Oh, L. Zettlemoyer, K. Lo, A. El-Nouby, H. Pouransari, A. Toshev, S. Wang, D. Groeneveld, L. Soldaini, P. W. Koh, J. Jitsev, T. Kollar, A. Dimakis, Y. Carmon, A. Dave, L. Schmidt, and V. Shankar. Datacomp-lm: In search of the next generation of training sets for language models. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL https://openreview.net/forum?id=CNWdWn47IE.

X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. B. Hashimoto. Diffusion-lm improves controllable text generation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL https://openreview.net/forum?id=3s9IrEsjLyk.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized BERT pretraining approach, 2019. URL http://arxiv.org/abs/1907.11692.

Llama Team. The llama 3 herd of models, 2024. URL https://ai.meta.com/research/publications/the-llama-3-herd-of-models/.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

A. Lou, C. Meng, and S. 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.

J. Lovelace, V. Kishore, C. Wan, E. Shekhtman, and K. Q. Weinberger. Latent diffusion for language generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL https://openreview.net/forum?id=NKdtztladR.

S. Nie, F. Zhu, C. Du, T. Pang, Q. Liu, G. Zeng, M. Lin, and C. Li. Scaling up masked diffusion models on text. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a. URL https://openreview.net/forum?id=WNvvwK0tut.

S. Nie, F. Zhu, Z. You, X. Zhang, J. Ou, J. Hu, J. Zhou, Y. Lin, J. Wen, and C. Li. Large language diffusion models. In D. Belgrave, C. Zhang, L. N. Montoya, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, N. Chen, I. V. M. Ruíz, and A. Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025b. URL https://openreview.net/forum?id=KnqiC0znVF.

OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://openai.com/index/introducing-gpt-oss/.

J. Ou, S. Nie, K. Xue, F. Zhu, J. Sun, Z. Li, and C. Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=sMyXP8Tanm.

Qwen Team. Qwen3 technical report, 2025. URL https://qwenlm.github.io/blog/qwen3/.

S. S. Sahoo, M. Arriola, Y. Schiff, A. Gokaslan, E. Marroquin, J. T. Chiu, A. Rush, and V. Kuleshov. Simple and effective masked diffusion language models. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024a. URL https://openreview.net/forum?id=L4uaAR4ArM.

S. S. Sahoo, A. Gokaslan, C. D. Sa, and V. Kuleshov. Diffusion models with learned adaptive noise. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024b. URL https://openreview.net/forum?id=loMa99A4p8.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. K. Li, Y. Wu, and D. Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL http://arxiv.org/abs/2402.03300.

J. Shi, K. Han, Z. Wang, A. Doucet, and M. K. Titsias. Simplified and generalized masked diffusion for discrete data. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL https://openreview.net/forum?id=xcqSOfHt4g.

J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In F. R. Bach and D. M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2256–2265. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/sohl-dickstein15.html.

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

Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11895–11907, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/3001ef257407d5a371a96dcd947c7d93-Abstract.html.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

R. Strudel, C. Tallec, F. Altché, Y. Du, Y. Ganin, A. Mensch, W. Grathwohl, N. Savinov, S. Dieleman, L. Sifre, and R. Leblond. Self-conditioned embedding diffusion for text generation, 2022. URL http://arxiv.org/abs/2211.04236.

H. Sun, L. Yu, B. Dai, D. Schuurmans, and H. Dai. Score-based continuous-time discrete diffusion models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=BYWWwSY2G5s.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

K. Wang, M. Shi, Y. Zhou, Z. Li, Z. Yuan, Y. Shang, X. Peng, H. Zhang, and Y. You. A closer look at time steps is worthy of triple speed-up for diffusion model training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 12934–12944. Computer Vision Foundation / IEEE, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_A_Closer_Look_at_Time_Steps_is_Worthy_of_Triple_CVPR_2025_paper.html.

Y. Wang, Y. Schiff, A. Gokaslan, W. Pan, F. Wang, C. D. Sa, and V. Kuleshov. Infodiffusion: Representation learning using information maximizing diffusion models. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 36336–36354. PMLR, 2023. URL https://proceedings.mlr.press/v202/wang23ah.html.

B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen, G. T. Adams, J. Howard, and I. Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 2526–2547. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.127. URL https://doi.org/10.18653/v1/2025.acl-long.127.

A. Wettig, T. Gao, Z. Zhong, and D. Chen. Should you mask 15% in masked language modeling? In A. Vlachos and I. Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 2977–2992. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EACL-MAIN.217. URL https://doi.org/10.18653/v1/2023.eacl-main.217.

A. Yang, B. Zhang, B. Hui, B. Gao, B. Yu, C. Li, D. Liu, J. Tu, J. Zhou, J. Lin, K. Lu, M. Xue, R. Lin, T. Liu, X. Ren, and Z. Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024. URL http://arxiv.org/abs/2409.12122.

Q. Yao, H. Yang, B. Han, G. Niu, and J. T. Kwok. Searching to exploit memorization effect in learning with noisy labels. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10789–10798. PMLR, 2020. URL http://proceedings.mlr.press/v119/yao20b.html.

J. Ye, Z. Xie, L. Zheng, J. Gao, Z. Wu, X. Jiang, Z. Li, and L. Kong. Dream 7b: Diffusion large language models, 2025. URL http://arxiv.org/abs/2508.15487.

S. Yuan, L. Feng, and T. Liu. Late stopping: Avoiding confidently learning from mislabeled examples. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 16033–16042. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01473. URL https://doi.org/10.1109/ICCV51070.2023.01473.

S. Yuan, L. Feng, B. Han, and T. Liu. Enhancing sample selection against label noise by cutting mislabeled easy examples. In D. Belgrave, C. Zhang, L. N. Montoya, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, N. Chen, I. V. M. Ruíz, and A. Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL https://openreview.net/forum?id=OfIUAlo2hJ.

H. Zhang, M. Dang, N. Peng, and G. V. den Broeck. Tractable control for autoregressive language generation. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 40932–40945. PMLR, 2023. URL https://proceedings.mlr.press/v202/zhang23g.html.

Q. Zhang, Y. Wang, and Y. Wang. An augmentation overlap theory of contrastive learning. *J. Mach. Learn. Res.*, 26:228:1–228:42, 2025. URL https://jmlr.org/papers/v26/22-1009.html.

Appendix

A Mathematical Details **18**

A.1 Proof of Lemma 3.2 . . . 18
A.2 Proof of Proposition 3.3 . . . 18
A.3 Proof of Proposition 3.4 . . . 18
A.4 Proof of Proposition 3.5 . . . 19
A.5 Proof of Lemma 3.7 . . . 19
A.6 Proof of Lemma 4.2 . . . 21
A.7 Proof of Theorem 4.3 . . . 22
A.8 Proof of Corollary 4.4 . . . 23
A.9 Proof of Assumption 4.5 . . . 23
A.10 Interchangability of Differentiation and Expectation . . . 24
A.10.1 Statement . . . 24
A.10.2 Proof . . . 25
A.11 Proof of Proposition 4.6 . . . 25
A.12 Proof of Proposition 4.8 . . . 26

B Modeling Considerations **28**

B.1 Boundary-Aware Power-Law Model . . . 28
B.2 Beta Proposal . . . 28
B.3 Fitted $\sqrt{\hat{V}_j}$ . . . 28

C Additional Experimental Details and Results **30**

C.1 Technical Details For Reproducibility . . . 30
C.2 Results on Pre-Trained Checkpoints . . . 30
C.2.1 SFT on `LLaDA-8B` . . . 30
C.2.2 SFT on `ModernBERT-Large` . . . 30
C.3 Ablation on Number of Masked Tokens . . . 30
C.4 Ablation on Form of Distribution . . . 31
C.5 Comparison to Sampling with Respect to Probabilities . . . 31
C.6 Training Dynamics . . . 32

D Extended Background **33**

D.1 Diffusion Language Modeling . . . 33

A Mathematical Details

A.1 Proof of Lemma 3.2

Lemma 3.2. Under Assumption 3.1, the triples $Z_i = (\boldsymbol{X}_i, t_i, \tilde{\boldsymbol{X}}_i)$ are i.i.d. for the current estimator. More generally, if a proposal $\hat{p}_k$ is chosen from past history $\mathcal{F}_{k-1}$, then conditional on $\mathcal{F}_{k-1}$ the current-step triples $Z_{k,i} = (\boldsymbol{X}_{k,i}, t_{k,i}, \tilde{\boldsymbol{X}}_{k,i})$ are i.i.d.

Proof. We first prove the one-step claim. Under Assumption 3.1, the pairs $(\boldsymbol{X}_i, t_i)$ are independent across i sampled from $\mathcal{D} \otimes p$, and conditional on $\{(\boldsymbol{X}_i, t_i)\}_{i=1}^N$, the variables $\tilde{\boldsymbol{X}}_i$ are independent with common kernel $q(\cdot \mid \boldsymbol{X}_i, t_i)$. Therefore for any measurable sets $A_1, \ldots, A_N$,

$$\mathbb{P}(Z_1 \in A_1, \ldots, Z_N \in A_N) = \mathbb{E}\left[\prod_{i=1}^N \mathbb{P}(Z_i \in A_i | \boldsymbol{X}_i, t_i)\right] \tag{24}$$

$$= \prod_{i=1}^N \mathbb{E}[\mathbb{P}(Z_i \in A_i | \boldsymbol{X}_i, t_i)], \tag{25}$$

which is exactly $\prod_{i=1}^N \mathbb{P}(Z_i \in A_i)$. Hence $Z_1, \ldots, Z_N$ are i.i.d.

For the adaptive statement, let $\hat{p}_k$ be $\mathcal{F}_{k-1}$-measurable. Conditional on $\mathcal{F}_{k-1}$, the same argument applies with p replaced by $\hat{p}_k$, so $Z_{k,1}, \ldots, Z_{k,N}$ are conditionally i.i.d.

A.2 Proof of Proposition 3.3

Proposition 3.3. For each fixed $t_0 \in (0, 1)$, the integrand of the MDLM risk at t_0 is the corresponding fixed-rate MLM-style denoising risk $\mathcal{R}_{\text{MLM}}(\theta; t_0)$ multiplied by the non-negative ELBO weight $\lambda(t_0)$. As such, MDLM optimizes a non-negative mixture of MLM-style sub-objectives indexed by t.

Proof. For each fixed $t \in (0, 1)$, define the fixed-rate denoising risk

$$\mathcal{R}_{\text{MLM}}(\theta; t) := \mathbb{E}_{\boldsymbol{X} \sim \mathcal{D}}\left[\mathbb{E}_{\tilde{\boldsymbol{X}} \sim q(\cdot|\boldsymbol{X}, t)}\left[\sum_{\ell=1}^L \mathbb{1}(\tilde{\boldsymbol{x}}^\ell = \boldsymbol{m}) \cdot \mathcal{L}\Big(\boldsymbol{x}^\ell, p_\theta(\tilde{\boldsymbol{X}}, \ell)\Big)\right]\right]. \tag{26}$$

This is exactly the MLM-style reconstruction objective induced by masking at corruption level t, or equivalently by the fixed masking rate $\beta(t) = 1 - \alpha(t)$.

Using the definition of $h_\theta(\boldsymbol{X}, t)$ from Equation 5, the MDLM risk can be written as

$$\mathcal{R}(\theta) = \mathbb{E}_{\boldsymbol{X} \sim \mathcal{D}}\left[\int_0^1 \lambda(t)\ h_\theta(\boldsymbol{X}, t)\, \mathrm{d}t\right] \tag{27}$$

$$= \int_0^1 \lambda(t)\ \mathbb{E}_{\boldsymbol{X} \sim \mathcal{D}}[h_\theta(\boldsymbol{X}, t)]\, \mathrm{d}t \tag{28}$$

$$= \int_0^1 \lambda(t)\ \mathcal{R}_{\text{MLM}}(\theta; t)\, \mathrm{d}t. \tag{29}$$

Since $\lambda(t) \geq 0$ for absorbing schedules, $\mathcal{R}(\theta)$ is a non-negative mixture of fixed-rate MLM-style objectives. In particular, for any fixed $t_0 \in (0, 1)$, the integrand evaluated at t_0 is $\lambda(t_0)\mathcal{R}_{\text{MLM}}(\theta; t_0)$.

A.3 Proof of Proposition 3.4

Proposition 3.4. The loss-weighted empirical MDLM estimator is unbiased for the risk $\mathcal{R}(\theta)$.

Proof. The empirical MDLM ELBO risk estimator is

$$\widehat{\mathcal{R}}_{\text{MDLM}}(\theta) = \frac{1}{N}\sum_{i=1}^N \lambda(t_i) \sum_{\ell=1}^L \mathbb{1}(\tilde{\boldsymbol{x}}_i^\ell = \boldsymbol{m})\, \mathcal{L}\Big(\boldsymbol{x}_i^\ell, p_\theta(\tilde{\boldsymbol{X}}_i, \ell)\Big), \tag{30}$$

where $\boldsymbol{X}_i \sim \mathcal{D},\ t_i \sim \text{Unif}[0, 1],\ \tilde{\boldsymbol{X}}_i \sim q(\cdot \mid \boldsymbol{X}_i, t_i)$.

For any fixed i, conditioning on $(\boldsymbol{X}_i, t_i)$ gives

$$\mathbb{E}\left[\lambda(t_i)\sum_{\ell}\mathbb{1}(\tilde{\boldsymbol{x}}_i^\ell = \boldsymbol{m})\,\mathcal{L}(\boldsymbol{x}_i^\ell, p_\theta(\tilde{\boldsymbol{X}}_i, \ell))\right] = \mathbb{E}_{\boldsymbol{X}}\left[\mathbb{E}_t\left[\lambda(t)\; h_\theta(\boldsymbol{X}, t)\right]\right] \tag{31}$$

$$= \mathbb{E}_{\boldsymbol{X}}\left[\int_0^1 \lambda(t)\; h_\theta(\boldsymbol{X}, t)\,\mathrm{d}t\right] \tag{32}$$

$$= \mathcal{R}(\theta)\,. \tag{33}$$

By linearity of expectation,

$$\mathbb{E}\left[\widehat{\mathcal{R}}_{\mathrm{MDLM}}(\theta)\right] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\lambda(t_i)\sum_{\ell}\mathbb{1}(\tilde{\boldsymbol{x}}_i^\ell = \boldsymbol{m})\,\mathcal{L}(\boldsymbol{x}_i^\ell, p_\theta(\tilde{\boldsymbol{X}}_i, \ell))\right] \tag{34}$$

$$= \mathcal{R}(\theta)\,. \tag{35}$$

Hence

$$\mathrm{Bias}\left(\widehat{\mathcal{R}}_{\mathrm{MDLM}}(\theta)\right) = \mathbb{E}[\widehat{\mathcal{R}}_{\mathrm{MDLM}}(\theta)] - \mathcal{R}(\theta) = 0.$$

A.4 Proof of Proposition 3.5

Proposition 3.5. The corresponding unweighted MLM-style estimator targets a different risk and is therefore, in general, biased for $\mathcal{R}(\theta)$ whenever $\lambda(t)$ is non-constant.

Proof. Suppose instead we use the *unweighted* estimator

$$\widehat{\mathcal{R}}_{\mathrm{unw}}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\sum_{\ell=1}^{L}\mathbb{1}(\tilde{\boldsymbol{x}}_i^\ell = \boldsymbol{m})\,\mathcal{L}\Big(\boldsymbol{x}_i^\ell, p_\theta(\tilde{\boldsymbol{X}}_i, \ell)\Big)\,. \tag{36}$$

By linearity of expectation and the fact that each summand has the same marginal distribution,

$$\mathbb{E}\Big[\widehat{\mathcal{R}}_{\mathrm{unw}}(\theta)\Big] = \mathbb{E}_{\boldsymbol{X},t,\tilde{\boldsymbol{X}}}\left[\sum_{\ell}\mathbb{1}(\tilde{\boldsymbol{x}}^\ell = \boldsymbol{m})\,\mathcal{L}(\boldsymbol{x}^\ell, p_\theta(\tilde{\boldsymbol{X}}, \ell))\right] \tag{37}$$

$$= \mathbb{E}_{\boldsymbol{X}}\left[\mathbb{E}_t\left[h_\theta(\boldsymbol{X}, t)\right]\right]. \tag{38}$$

Hence the unweighted estimator targets the risk

$$\mathcal{R}_{\mathrm{unw}}(\theta) := \mathbb{E}_{\boldsymbol{X}}\left[\int_0^1 h_\theta(\boldsymbol{X}, t)\,\mathrm{d}t\right]. \tag{39}$$

By definition,

$$\mathrm{Bias}\Big(\widehat{\mathcal{R}}_{\mathrm{unw}}(\theta)\Big) = \mathbb{E}\Big[\widehat{\mathcal{R}}_{\mathrm{unw}}(\theta)\Big] - \mathcal{R}(\theta) \tag{40}$$

$$= \mathcal{R}_{\mathrm{unw}}(\theta) - \mathcal{R}(\theta) \tag{41}$$

$$= \mathbb{E}_{\boldsymbol{X}}\left[\int_0^1 (1-\lambda(t))\,h_\theta(\boldsymbol{X}, t)\,\mathrm{d}t\right]. \tag{42}$$

Therefore the unweighted estimator is unbiased for $\mathcal{R}(\theta)$ only if either $\lambda(t) \equiv 1$ almost everywhere or the integral above cancels for the particular function $h_\theta(\boldsymbol{X}, t)$. When $\lambda(t)$ is non-constant, such cancellation is not guaranteed and the estimator targets the different risk $\mathcal{R}_{\mathrm{unw}}(\theta)$ instead. This is exactly the sense in which the unweighted estimator is generally biased for the MDLM risk.

A.5 Proof of Lemma 3.7

Lemma 3.7. Under Assumption 3.1 with uniform time-step sampling, let $Z = (\boldsymbol{X}, t, \tilde{\boldsymbol{X}})$ with $\boldsymbol{X} \sim \mathcal{D}$, $t \sim \mathrm{Unif}[0,1]$, and $\tilde{\boldsymbol{X}} \sim q\,(\cdot|\boldsymbol{X}, t)$. Define

$$g_\theta(Z) := \sum_{\ell=1}^{L}\mathbb{1}(\tilde{\boldsymbol{x}}^\ell = \boldsymbol{m})\cdot\mathcal{L}\Big(\boldsymbol{x}^\ell, p_\theta(\tilde{\boldsymbol{X}}, \ell)\Big)\,, \qquad Y_\theta(Z) := \lambda(t)\cdot g_\theta(Z)\,,$$

$$m_\theta(t) := \mathbb{E}[Y_\theta(Z)\mid t]\,, \qquad V_\theta(t) := \mathbb{E}\Big[Y_\theta(Z)^2 \mid t\Big]\,.$$

Then

$$\mathrm{Var}\Big(\widehat{\mathcal{R}}_N(\theta)\Big) = \frac{1}{N}\left[\int_0^1 V_\theta(t)\ \mathrm{d}t - \mathcal{R}(\theta)^2\right] = \frac{1}{N}\left[\mathbb{E}_t[\mathrm{Var}(Y_\theta(Z)\mid t)] + \mathrm{Var}_t(m_\theta(t))\right].$$

Likewise,

$$m_\theta^{\mathrm{unw}}(t) := \mathbb{E}[g_\theta(Z)|t]\,,\ V_\theta^{\mathrm{unw}}(t) := \mathbb{E}\Big[g_\theta(Z)^2\Big|t\Big]$$
$$\Longrightarrow \mathrm{Var}\Big(\widehat{\mathcal{R}}_N^{\mathrm{unw}}(\theta)\Big) = \frac{1}{N}\left[\int_0^1 V_\theta^{\mathrm{unw}}(t)\ \mathrm{d}t - \mathcal{R}_{\mathrm{unw}}(\theta)^2\right].$$

Proof. Let

$$Y_i := Y_\theta(Z_i) = \lambda(t_i)\, g_\theta(Z_i)\,, \qquad Z_i = (\boldsymbol{X}_i, t_i, \tilde{\boldsymbol{X}}_i).$$

Then

$$\widehat{\mathcal{R}}_N(\theta) = \frac{1}{N}\sum_{i=1}^N Y_i.$$

Under uniform time-step sampling, Lemma 3.2 implies that the current-step variables Z_i are i.i.d. Hence the Y_i are also i.i.d., and therefore

$$\mathrm{Var}\Big(\widehat{\mathcal{R}}_N(\theta)\Big) = \frac{1}{N}\mathrm{Var}(Y_1)\,. \tag{43}$$

Now condition on the sampled time-step t. By definition of $V_\theta(t)$,

$$\mathbb{E}[Y_1^2] = \int_0^1 \mathbb{E}\Big[Y_\theta(Z)^2 \mid t\Big]\,\mathrm{d}t \tag{44}$$
$$= \int_0^1 V_\theta(t)\ \mathrm{d}t. \tag{45}$$

Since $\mathbb{E}[Y_1] = \mathcal{R}(\theta)$ by Proposition 3.4,

$$\mathrm{Var}(Y_1) = \mathbb{E}[Y_1^2] - \mathbb{E}[Y_1]^2 = \int_0^1 V_\theta(t)\ \mathrm{d}t - \mathcal{R}(\theta)^2\,, \tag{46}$$

which proves the first equality in Equation 8.

For the second equality, apply the law of total variance to $Y_\theta(Z)$:

$$\mathrm{Var}(Y_\theta(Z)) = \mathbb{E}_t[\mathrm{Var}(Y_\theta(Z)\mid t)] + \mathrm{Var}_t(\mathbb{E}[Y_\theta(Z)\mid t])\,. \tag{47}$$

Using $m_\theta(t) := \mathbb{E}[Y_\theta(Z)\mid t]$ and dividing by N yields

$$\mathrm{Var}\Big(\widehat{\mathcal{R}}_N(\theta)\Big) = \frac{1}{N}\left[\mathbb{E}_t[\mathrm{Var}(Y_\theta(Z)|t)] + \mathrm{Var}_t(m_\theta(t))\right]. \tag{48}$$

The unweighted case is identical with Y_i replaced by $g_\theta(Z_i)$. Defining

$$\widehat{\mathcal{R}}_N^{\mathrm{unw}}(\theta) = \frac{1}{N}\sum_{i=1}^N g_\theta(Z_i)$$

gives

$$\mathrm{Var}\Big(\widehat{\mathcal{R}}_N^{\mathrm{unw}}(\theta)\Big) = \frac{1}{N}\mathrm{Var}(g_\theta(Z)) = \frac{1}{N}\left[\int_0^1 V_\theta^{\mathrm{unw}}(t)\ \mathrm{d}t - \mathcal{R}_{\mathrm{unw}}(\theta)^2\right],$$

which is equivalent to Equation 9.

A.6 Proof of Lemma 4.2

Lemma 4.2. Let p be any admissible proposal density on $[0,1]$, and define the estimator

$$\widehat{\mathcal{R}}_{N,p}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\frac{\ell_\theta(\boldsymbol{X}_i, \tilde{\boldsymbol{X}}_i, t_i)}{p(t_i)},$$

where $\boldsymbol{X}_i \sim \mathcal{D}, t_i \sim p, \tilde{\boldsymbol{X}}_i \sim q\left(\cdot|\boldsymbol{X}_i, t_i\right)$. Then

$$\mathbb{E}\left[\widehat{\mathcal{R}}_{N,p}(\theta)\right] = \mathcal{R}(\theta)\,, \mathrm{Var}\left(\widehat{\mathcal{R}}_{N,p}(\theta)\right) = \frac{1}{N}\left[\int_0^1 \frac{V_\theta(t)}{p(t)}\mathrm{d}t - \mathcal{R}(\theta)^2\right].$$

if Assumption 3.1 holds for the current minibatch with proposal p.

Proof. For convenience, define

$$Y_i := \frac{\ell_\theta(\boldsymbol{X}_i, \tilde{\boldsymbol{X}}_i, t_i)}{p(t_i)}.$$

Then

$$\widehat{\mathcal{R}}_{N,p}(\theta) = \frac{1}{N}\sum_{i=1}^{N} Y_i.$$

We first compute the expectation of one term. Using the law of total expectation and $t_i \sim p$,

$$\mathbb{E}[Y_i] = \int_0^1 \mathbb{E}_{\boldsymbol{X}\sim\mathcal{D},\tilde{\boldsymbol{X}}\sim q(\cdot|\boldsymbol{X},t)}\left[\frac{\ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)}{p(t)}\right] p(t)\ \mathrm{d}t \tag{49}$$

$$= \int_0^1 \mathbb{E}_{\boldsymbol{X}\sim\mathcal{D},\tilde{\boldsymbol{X}}\sim q(\cdot|\boldsymbol{X},t)}\left[\ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)\right] \mathrm{d}t \tag{50}$$

$$= \mathcal{R}(\theta)\,. \tag{51}$$

Hence, by linearity of expectation,

$$\mathbb{E}\left[\widehat{\mathcal{R}}_{N,p}\left(\theta\right)\right] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[Y_i] = \mathcal{R}(\theta)\,, \tag{52}$$

which proves un-biasedness.

For the variance identity, assume Assumption 3.1. By Lemma 3.2, the current-step variables $(\boldsymbol{X}_i, t_i, \tilde{\boldsymbol{X}}_i)$ are i.i.d., so the random variables $\{Y_i\}_{i=1}^N$ are i.i.d. Next, because p is admissible,

$$\mathbb{E}[Y_i^2] = \int_0^1 \mathbb{E}_{\boldsymbol{X},\tilde{\boldsymbol{X}}|t}\left[\frac{\ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)^2}{p(t)^2}\right] p(t)\ \mathrm{d}t \tag{53}$$

$$= \int_0^1 \frac{1}{p(t)}\mathbb{E}_{\boldsymbol{X},\tilde{\boldsymbol{X}}|t}\left[\ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)^2\right] \mathrm{d}t \tag{54}$$

$$= \int_0^1 \frac{V_\theta(t)}{p(t)}\mathrm{d}t. \tag{55}$$

Therefore,

$$\mathrm{Var}(Y_i) = \mathbb{E}[Y_i^2] - \mathbb{E}[Y_i]^2 = \int_0^1 \frac{V_\theta(t)}{p(t)}\mathrm{d}t - \mathcal{R}(\theta)^2\,. \tag{56}$$

Since the Y_i are i.i.d.,

$$\mathrm{Var}\left(\widehat{\mathcal{R}}_{N,p}(\theta)\right) = \mathrm{Var}\left(\frac{1}{N}\sum_{i=1}^{N} Y_i\right) = \frac{1}{N}\mathrm{Var}(Y_1)\,, \tag{57}$$

which gives

$$\mathrm{Var}\left(\widehat{\mathcal{R}}_{N,p}(\theta)\right) = \frac{1}{N}\left[\int_0^1 \frac{V_\theta(t)}{p(t)}\mathrm{d}t - \mathcal{R}(\theta)^2\right].$$

The conditional adaptive version is identical after conditioning on the pre-batch history $\mathcal{F}_{k-1}$: the proposal $\hat{p}_k$ is then fixed, the marginal-law assumption yields the conditional unbiasedness identity, and the conditional version of Assumption 3.1 yields the same variance formula with p replaced by $\hat{p}_k$.

A.7 Proof of Theorem 4.3

Theorem 4.3. Under Assumption 3.1 and 4.1,, the variance in Equation 14 is minimized by

$$p^\star_\theta(t) = \sqrt{V_\theta(t)} \Big/ \int_0^1 \sqrt{V_\theta(u)}\,\mathrm{d}u$$

with corresponding minimum variance

$$\mathrm{Var}\left(\widehat{\mathcal{R}}_{N,p^\star_\theta}(\theta)\right) = \frac{1}{N}\left[\left(\int_0^1 \sqrt{V_\theta(t)}\,\mathrm{d}t\right)^2 - \mathcal{R}(\theta)^2\right].$$

Proof. By Lemma 4.2, for fixed θ and N, minimizing the variance of $\widehat{\mathcal{R}}_{N,p}(\theta)$ over proposal densities p is equivalent to minimizing

$$\int_0^1 \frac{V_\theta(t)}{p(t)}\mathrm{d}t$$

subject to $\int_0^1 p(t)\ \mathrm{d}t = 1$ and $p(t) > 0$ wherever $V_\theta(t) > 0$.

For any such p, Cauchy–Schwarz gives

$$\left(\int_0^1 \sqrt{V_\theta(t)}\,\mathrm{d}t\right)^2 = \left(\int_0^1 \sqrt{\frac{V_\theta(t)}{p(t)}}\,\sqrt{p(t)}\,\mathrm{d}t\right)^2 \tag{58}$$

$$\leq \left(\int_0^1 \frac{V_\theta(t)}{p(t)}\mathrm{d}t\right)\left(\int_0^1 p(t)\ \mathrm{d}t\right) \tag{59}$$

$$= \int_0^1 \frac{V_\theta(t)}{p(t)}\mathrm{d}t. \tag{60}$$

Thus every admissible p satisfies

$$\int_0^1 \frac{V_\theta(t)}{p(t)}\mathrm{d}t \geq \left(\int_0^1 \sqrt{V_\theta(t)}\,\mathrm{d}t\right)^2. \tag{61}$$

Equality in Cauchy–Schwarz holds if and only if

$$\sqrt{\frac{V_\theta(t)}{p(t)}} = c\,\sqrt{p(t)}$$

for some constant $c > 0$, or equivalently

$$p(t)^2 = \frac{V_\theta(t)}{c^2}.$$

Hence the minimizer must satisfy $p(t) \propto \sqrt{V_\theta(t)}$. Normalizing this probability distribution such that it integrates to 1 then yields

$$p^\star_\theta(t) = \frac{\sqrt{V_\theta(t)}}{\int_0^1 \sqrt{V_\theta(u)}\,\mathrm{d}u}.$$

Substituting $p^\star_\theta$ into the left-hand side of Equation 61 gives

$$\int_0^1 \frac{V_\theta(t)}{p^\star_\theta(t)}\mathrm{d}t = \int_0^1 V_\theta(t)\ \frac{\int_0^1 \sqrt{V_\theta(u)}\,\mathrm{d}u}{\sqrt{V_\theta(t)}}\mathrm{d}t \tag{62}$$

$$= \left(\int_0^1 \sqrt{V_\theta(u)}\,\mathrm{d}u\right)\left(\int_0^1 \sqrt{V_\theta(t)}\,\mathrm{d}t\right) \tag{63}$$

$$= \left(\int_0^1 \sqrt{V_\theta(t)}\,\mathrm{d}t\right)^2. \tag{64}$$

Applying Lemma 4.2 then yields

$$\mathrm{Var}\left(\widehat{\mathcal{R}}_{N,p^\star_\theta}(\theta)\right) = \frac{1}{N}\left[\left(\int_0^1 \sqrt{V_\theta(t)}\,\mathrm{d}t\right)^2 - \mathcal{R}(\theta)^2\right].$$

A.8 Proof of Corollary 4.4

Corollary 4.4. Under Assumption 3.1 and 4.1, Let $p_{\text{uni}}(t) = 1$ be the uniform density on $[0, 1]$. Then

$$\text{Var}\Big(\widehat{\mathcal{R}}_{N,p^\star_\theta}(\theta)\Big) \le \text{Var}\Big(\widehat{\mathcal{R}}_{N,p_{\text{uni}}}(\theta)\Big),$$

with equality if and only if $V_\theta(t)$ is constant almost everywhere.

Proof. Applying Lemma 4.2 with the uniform distribution $p_{\text{uni}}(t) = 1$ on $[0, 1]$ gives

$$\text{Var}\Big(\widehat{\mathcal{R}}_{N,p_{\text{uni}}}(\theta)\Big) = \frac{1}{N}\left[\int_0^1 V_\theta(t)\ \mathsf{d}t - \mathcal{R}(\theta)^2\right]. \tag{65}$$

Theorem 4.3 gives

$$\text{Var}\Big(\widehat{\mathcal{R}}_{N,p^\star_\theta}(\theta)\Big) = \frac{1}{N}\left[\left(\int_0^1 \sqrt{V_\theta(t)}\,\mathsf{d}t\right)^2 - \mathcal{R}(\theta)^2\right]. \tag{66}$$

Then by comparing

$$\left(\int_0^1 \sqrt{V_\theta(t)}\,\mathsf{d}t\right)^2 \le \left(\int_0^1 V_\theta(t)\ \mathsf{d}t\right)\left(\int_0^1 1\,\mathsf{d}t\right) = \int_0^1 V_\theta(t)\ \mathsf{d}t$$

using Cauchy-Schwarz, we arrive at

$$\text{Var}\Big(\widehat{\mathcal{R}}_{N,p^\star_\theta}(\theta)\Big) \le \text{Var}\Big(\widehat{\mathcal{R}}_{N,p_{\text{uni}}}(\theta)\Big).$$

Equality in Cauchy–Schwarz holds if and only if $\sqrt{V_\theta(t)}$ is constant almost everywhere, which is equivalent to $V_\theta(t)$ being constant almost everywhere.

A.9 Proof of Assumption 4.5

Assumption 4.5. There exists $C_{\text{grad}} > 0$ such that for all $t \in [0, 1]$,

$$\mathbb{E}_{\boldsymbol{X},\tilde{\boldsymbol{X}}|t}\left[\left\|\nabla_\theta\, \ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)\right\|_2^2\right] \le C_{\text{grad}} \cdot V_\theta(t).$$

We prove this through the following sufficient condition.

Proposition A.1 (Sufficient condition for Assumption 4.5). *Assume that the per-token denoising loss is cross-entropy, and let $z_\theta(\tilde{\boldsymbol{X}}, \ell)$ denote the pre-softmax logits used to predict token $\boldsymbol{x}^\ell$. Suppose there exist constants $B_z, m_0 > 0$ such that along the training regime of interest:*

1. *For every masked position ℓ, the logit Jacobian satisfies*

$$\left\|\nabla_\theta\, z_\theta(\tilde{\boldsymbol{X}}, \ell)\right\|_{\text{op}} \le B_z;$$

2. *For every masked position ℓ, the cross-entropy obeys*

$$\mathcal{L}\Big(\boldsymbol{x}^\ell, p_\theta(\tilde{\boldsymbol{X}}, \ell)\Big) \ge m_0.$$

Then Assumption 4.5 holds with $C_{\text{grad}} = 2B_z^2/m_0$.

Proof. Let $M(\tilde{\boldsymbol{X}}) = \{\ell : \tilde{\boldsymbol{x}}^\ell = \boldsymbol{m}\}$ be the masked positions, and write

$$\mathcal{L}_\ell := \mathcal{L}\Big(\boldsymbol{x}^\ell, p_\theta(\tilde{\boldsymbol{X}}, \ell)\Big), \qquad r_\ell := \nabla_{z_\theta(\tilde{\boldsymbol{X}},\ell)}\mathcal{L}_\ell.$$

Then

$$\ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t) = \lambda(t) \sum_{\ell \in M(\tilde{\boldsymbol{X}})} \mathcal{L}_\ell$$

and

$$\nabla_\theta \, \ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t) = \lambda(t) \sum_{\ell \in M(\tilde{\boldsymbol{X}})} \left(\nabla_\theta \, z_\theta \left(\tilde{\boldsymbol{X}}, \ell \right) \right)^\top r_\ell.$$

Using $\| \sum_j a_j \|_2^2 \le |M| \sum_j \|a_j\|_2^2$ and the Jacobian bound,

$$\left\| \nabla_\theta \, \ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t) \right\|_2^2 \le \lambda(t)^2 \, |M(\tilde{\boldsymbol{X}})| \sum_{\ell \in M(\tilde{\boldsymbol{X}})} \left\| \left(\nabla_\theta \, z_\theta \left(\tilde{\boldsymbol{X}}, \ell \right) \right)^\top r_\ell \right\|_2^2 \tag{67}$$

$$\le \lambda(t)^2 \, |M(\tilde{\boldsymbol{X}})| \, B_z^2 \sum_{\ell \in M(\tilde{\boldsymbol{X}})} \|r_\ell\|_2^2. \tag{68}$$

For one-hot cross-entropy, $\mathcal{L}_\ell = D_{\mathrm{KL}}(e_{\boldsymbol{x}^\ell} \| p_\theta(\tilde{\boldsymbol{X}}, \ell))$, so Pinsker's inequality gives

$$\|r_\ell\|_2^2 = \left\| p_\theta(\tilde{\boldsymbol{X}}, \ell) - e_{\boldsymbol{x}^\ell} \right\|_2^2 \le \left\| p_\theta(\tilde{\boldsymbol{X}}, \ell) - e_{\boldsymbol{x}^\ell} \right\|_1^2 \le 2\mathcal{L}_\ell.$$

Therefore

$$\left\| \nabla_\theta \, \ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t) \right\|_2^2 \le 2\lambda(t)^2 \, |M(\tilde{\boldsymbol{X}})| \, B_z^2 \sum_{\ell \in M(\tilde{\boldsymbol{X}})} \mathcal{L}_\ell.$$

By the loss-floor assumption, $\sum_{\ell \in M(\tilde{\boldsymbol{X}})} \mathcal{L}_\ell \ge m_0 \, |M(\tilde{\boldsymbol{X}})|$, hence

$$|M(\tilde{\boldsymbol{X}})| \le \frac{1}{m_0} \sum_{\ell \in M(\tilde{\boldsymbol{X}})} \mathcal{L}_\ell.$$

Substituting this into the previous display yields the pointwise bound

$$\left\| \nabla_\theta \, \ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t) \right\|_2^2 \le \frac{2B_z^2}{m_0} \lambda(t)^2 \left(\sum_{\ell \in M(\tilde{\boldsymbol{X}})} \mathcal{L}_\ell \right)^2 = \frac{2B_z^2}{m_0} \, \ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)^2.$$

Taking conditional expectations given t proves Assumption 4.5 with $C_{\mathrm{grad}} = 2B_z^2/m_0$.

A.10 Interchangability of Differentiation and Expectation

In Proposition 4.6, we mention the interchangeability of differentiation and expectation in the weighted loss; for succinctness, we formalize this in more detail to justify our proposition.

A.10.1 Statement

Assumption A.2 (Gradient-loss coupling). There exists $C_{\mathrm{grad}} > 0$ such that for all $t \in [0, 1]$,

$$\mathbb{E}_{\boldsymbol{X}, \tilde{\boldsymbol{X}} | t} \left[\left\| \nabla_\theta \, \ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t) \right\|_2^2 \right] \le C_{\mathrm{grad}} \cdot V_\theta(t) \, . \tag{69}$$

Assumption A.3 (Dominated differentiability of the weighted loss). Fix an admissible proposal density p and a parameter value θ. There exists an open neighborhood U_θ of θ such that, for almost every $(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)$, the map $\vartheta \mapsto \ell_\vartheta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)$ is differentiable on U_θ, and there is an integrable envelope $H_{\theta,p}(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)$ satisfying

$$\sup_{\vartheta \in U_\theta} \frac{\|\nabla_\vartheta \, \ell_\vartheta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)\|_2}{p(t)} \le H_{\theta,p}(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t), \qquad \mathbb{E}_{\boldsymbol{X} \sim \mathcal{D}, \, t \sim p, \, \tilde{\boldsymbol{X}} \sim q(\cdot | \boldsymbol{X}, t)} \left[H_{\theta,p}(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t) \right] < \infty. \tag{70}$$

Proposition A.4. *Under Assumption 3.1, fix any admissible proposal density p for which Assumption A.3 holds at the current parameter value θ. Then the gradient estimator induced by $\widehat{\mathcal{R}}_{N,p}$ is unbiased, and under Assumption 4.5 satisfies*

$$\mathbb{E} \left[\left\| \nabla_\theta \, \widehat{\mathcal{R}}_{N,p}(\theta) - \nabla_\theta \, \mathcal{R}(\theta) \right\|_2^2 \right] \le \frac{C_{\mathrm{grad}}}{N} \int_0^1 \frac{V_\theta(t)}{p(t)} \mathrm{d}t. \tag{71}$$

Because p is independent of θ, the only nontrivial step is moving ∇_θ through the expectation over data, corruption, and sampled time. Assumption A.3 is a standard dominated-differentiation condition for this weighted loss: it requires almost-sure local differentiability of the per-example objective together with an integrable envelope for the weighted gradient. Proposition A.5 makes this precise and shows that the interchange follows by dominated convergence; see, e.g., Folland (1999).

Reducing $\int_0^1 V_\theta(t)\,/p(t)\;\mathrm{d}t$ thus not only lowers the scalar risk estimator variance, but also a gradient noise upper bound. Consequently, a good proposal should sample more from time-steps where weighted denoising losses exhibit large second moments.

A.10.2 Proof

Proposition A.5 (Sufficient condition for Assumption A.3). *Fix an admissible proposal density p and a parameter value θ. Suppose there exists an open neighborhood U_θ of θ such that, for almost every $(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)$, the map $\vartheta \mapsto \ell_\vartheta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)$ is differentiable on U_θ, and*

$$\sup_{\vartheta \in U_\theta} \frac{\|\nabla_\vartheta\, \ell_\vartheta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)\|_2}{p(t)} \le H_{\theta,p}(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)$$

for some integrable envelope $H_{\theta,p}$ with

$$\mathbb{E}_{\boldsymbol{X}\sim\mathcal{D},\; t\sim p,\; \tilde{\boldsymbol{X}}\sim q(\cdot|\boldsymbol{X},t)}\Big[H_{\theta,p}(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)\Big] < \infty.$$

Then

$$\nabla_\theta \mathbb{E}\left[\frac{\ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)}{p(t)}\right] = \mathbb{E}\left[\frac{\nabla_\theta\, \ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)}{p(t)}\right].$$

Proof. Let e_j denote the j-th coordinate basis vector of the parameter space, and write

$$Z := (\boldsymbol{X}, \tilde{\boldsymbol{X}}, t).$$

For h small enough that the line segment $\{\theta + she_j : s \in [0,1]\} \subset U_\theta$, the fundamental theorem of calculus gives

$$\left|\frac{\ell_{\theta+he_j}(Z) - \ell_\theta(Z)}{h\,p(t)}\right| = \left|\frac{1}{p(t)}\int_0^1 \left\langle \nabla_\vartheta \ell_{\theta+she_j}(Z), e_j\right\rangle \mathrm{d}s\right| \tag{72}$$

$$\le \sup_{\vartheta\in U_\theta} \frac{\|\nabla_\vartheta\, \ell_\vartheta(Z)\|_2}{p(t)} \tag{73}$$

$$\le H_{\theta,p}(Z). \tag{74}$$

For almost every Z, the difference quotient converges as $h \to 0$ to

$$\frac{\partial_{\theta_j} \ell_\theta(Z)}{p(t)}.$$

Since it is dominated by the integrable envelope $H_{\theta,p}$, dominated convergence (Folland, 1999) yields

$$\partial_{\theta_j} \mathbb{E}\left[\frac{\ell_\theta(Z)}{p(t)}\right] = \mathbb{E}\left[\frac{\partial_{\theta_j} \ell_\theta(Z)}{p(t)}\right].$$

Applying the same argument coordinate-wise proves the vector identity.

A.11 Proof of Proposition 4.6

We prove Proposition A.4 as a more detailed version of Proposition 4.6.

Proposition A.4. Under Assumption 3.1, fix any admissible proposal density p for which Assumption A.3 holds at the current parameter value θ. Then the gradient estimator induced by $\widehat{\mathcal{R}}_{N,p}$ is unbiased, and under Assumption 4.5 satisfies

$$\mathbb{E}\left[\left\|\nabla_\theta \widehat{\mathcal{R}}_{N,p}(\theta) - \nabla_\theta \mathcal{R}(\theta)\right\|_2^2\right] \le \frac{C_{\mathrm{grad}}}{N}\int_0^1 \frac{V_\theta(t)}{p(t)}\mathrm{d}t.$$

Proof. Define

$$G_i := \frac{\nabla_\theta \, \ell_\theta(\boldsymbol{X}_i, \tilde{\boldsymbol{X}}_i, t_i)}{p(t_i)}, \qquad \bar{G} := \nabla_\theta \, \mathcal{R}(\theta) \, .$$

Then

$$\nabla_\theta \, \widehat{\mathcal{R}}_{N,p}(\theta) = \frac{1}{N} \sum_{i=1}^{N} G_i.$$

By Assumption A.3, the weighted loss satisfies the hypotheses of Proposition A.5; hence differentiation and expectation can be interchanged for the weighted loss. Together with the same conditioning argument used in Lemma 4.2,

$$\mathbb{E}[G_i] = \int_0^1 \mathbb{E}_{\boldsymbol{X}, \tilde{\boldsymbol{X}}|t} \left[\frac{\nabla_\theta \, \ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)}{p(t)} \right] p(t) \, \mathrm{d}t \tag{75}$$

$$= \int_0^1 \mathbb{E}_{\boldsymbol{X}, \tilde{\boldsymbol{X}}|t} \left[\nabla_\theta \, \ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t) \right] \mathrm{d}t \tag{76}$$

$$= \nabla_\theta \int_0^1 \mathbb{E}_{\boldsymbol{X}, \tilde{\boldsymbol{X}}|t} \left[\ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t) \right] \mathrm{d}t \tag{77}$$

$$= \nabla_\theta \, \mathcal{R}(\theta) = \bar{G}. \tag{78}$$

Hence $\nabla_\theta \, \widehat{\mathcal{R}}_{N,p}(\theta)$ is unbiased.

Next, by Lemma 3.2 the G_i are i.i.d., and hence independent and centered after subtracting $\bar{G}$,

$$\mathbb{E}\left[\left\| \nabla_\theta \, \widehat{\mathcal{R}}_{N,p}(\theta) - \nabla_\theta \, \mathcal{R}(\theta) \right\|_2^2 \right] = \mathbb{E}\left[\left\| \frac{1}{N} \sum_{i=1}^{N} (G_i - \bar{G}) \right\|_2^2 \right] \tag{79}$$

$$= \frac{1}{N^2} \sum_{i=1}^{N} \mathbb{E}\big[\|G_i - \bar{G}\|_2^2 \big] \tag{80}$$

$$= \frac{1}{N} \mathbb{E}\big[\|G_1 - \bar{G}\|_2^2 \big] \tag{81}$$

$$\leq \frac{1}{N} \mathbb{E}\big[\|G_1\|_2^2 \big] \, . \tag{82}$$

Expanding the remaining expectation gives

$$\mathbb{E}\big[\|G_1\|_2^2 \big] = \int_0^1 \mathbb{E}_{\boldsymbol{X}, \tilde{\boldsymbol{X}}|t} \left[\frac{\|\nabla_\theta \, \ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)\|_2^2}{p(t)^2} \right] p(t) \, \mathrm{d}t \tag{83}$$

$$= \int_0^1 \frac{1}{p(t)} \mathbb{E}_{\boldsymbol{X}, \tilde{\boldsymbol{X}}|t} \left[\|\nabla_\theta \, \ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)\|_2^2 \right] \mathrm{d}t \tag{84}$$

$$\leq C_{\mathrm{grad}} \int_0^1 \frac{V_\theta(t)}{p(t)} \mathrm{d}t, \tag{85}$$

where the last step uses Assumption 4.5. Combining the above displays proves

$$\mathbb{E}\left[\left\| \nabla_\theta \, \widehat{\mathcal{R}}_{N,p}(\theta) - \nabla_\theta \, \mathcal{R}(\theta) \right\|_2^2 \right] \leq \frac{C_{\mathrm{grad}}}{N} \int_0^1 \frac{V_\theta(t)}{p(t)} \mathrm{d}t.$$

A.12 Proof of Proposition 4.8

Proposition 4.8. Under Theorem 4.3 and Assumption 4.7,

$$p_\theta^\star(t) \propto t^{a/2}(1-t)^{b/2} \approx \mathrm{Beta}(t; \mu, \nu) \, , \text{where } \mu = 1 + a/2, \nu = 1 + b/2.$$

Proof. Under Assumption 4.7,

$$V_\theta(t) \approx C \cdot t^a (1-t)^b$$

for constants $C > 0$ and $a, b > -2$. Substituting this approximation into $p^\star$ from Theorem 4.3 gives

$$p^\star_\theta(t) \approx \frac{\sqrt{C \cdot t^a (1-t)^b}}{\int_0^1 \sqrt{C \cdot u^a (1-u)^b}\, \mathrm{d}u} = \frac{t^{a/2}(1-t)^{b/2}}{\int_0^1 u^{a/2}(1-u)^{b/2}\, \mathrm{d}u}. \tag{86}$$

Because $a, b > -2$, we have $1 + a/2 > 0$ and $1 + b/2 > 0$, so the denominator is finite and the normalized density is well-defined on $(0, 1)$.

Now recall that the Beta density with parameters (μ, ν) is

$$\mathrm{Beta}(t; \mu, \nu) = \frac{1}{B(\mu, \nu)} t^{\mu-1}(1-t)^{\nu-1}, \qquad t \in (0, 1),$$

where $B(\mu, \nu)$ is the Beta function. Matching exponents gives

$$\mu - 1 = \frac{a}{2}, \qquad \nu - 1 = \frac{b}{2} \iff \mu = 1 + \frac{a}{2}, \qquad \nu = 1 + \frac{b}{2}.$$

Therefore the approximating optimal proposal is exactly a Beta density with these parameters.

B Modeling Considerations

B.1 Boundary-Aware Power-Law Model

Section 4 models the *weighted* timestep-wise second moment

$$V_\theta(t) = \mathbb{E}\Big[\ell_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}, t)^2\Big]$$

rather than the variance of an unweighted token loss. This distinction matters because the ELBO weight $\lambda(t) = 1/t$ is already absorbed into ℓ_θ, so the relevant profile for optimal sampling is the full second moment of the weighted MDLM loss. In the broader diffusion literature, timestep dependence is often expressed through low-dimensional functions of the noise level or log-SNR, and simple schedule parameterizations already account for a substantial amount of training behavior (Karras et al., 2022; Kingma and Gao, 2023; Hang and Gu, 2024).

Under the linear masking schedule, each token is masked independently with probability t. For a sequence of length L, the number of masked positions therefore satisfies

$$K_t \sim \mathrm{Binomial}(L, t)\,.$$

If the total denoising loss is the sum of contributions over masked positions, then both its expectation and its raw second moment inherit factors from the first two moments of K_t, namely Lt and $Lt(1-t)$. This does not force an exact polynomial law, but it does suggest that the dependence on t should be smooth, low-dimensional, and especially structured near the boundaries $t \approx 0$ and $t \approx 1$.

Those boundary regimes are also qualitatively distinct. Near $t = 0$, very few tokens are masked, so the loss is dominated by rare-mask events combined with the ELBO reweighting. Near $t = 1$, almost every token is masked, and the denoising problem approaches a fully corrupted reconstruction task. Recent timestep weighting and adaptive sampling papers report that the effective importance of diffusion steps is strongly non-uniform and often concentrated near boundary regimes, which is exactly the pattern that a two-exponent surrogate is designed to express (Hang et al., 2023; Lee et al., 2025; Kim et al., 2025). A model of the form

$$V_\theta(t) \approx C \cdot t^a (1-t)^b\,, \qquad C > 0,\ a, b > -2,$$

is therefore a natural low-parameter surrogate: the exponent a captures effective low-noise behavior, the exponent b captures effective high-noise behavior, and the condition $a, b > -2$ guarantees that the induced optimal proposal remains normalizable.

B.2 Beta Proposal

By Theorem 4.3, the variance-optimal sampling density satisfies

$$p^\star_\theta(t) \propto \sqrt{V_\theta(t)}.$$

Under the boundary-aware power-law surrogate above, this becomes

$$p^\star_\theta(t) \propto t^{a/2}(1-t)^{b/2}\,,$$

which is exactly a Beta density after normalization. Thus the Beta family is not an arbitrary convenience class; it is the normalized form of the optimal proposal under the structural approximation used in Section 4.

The same conclusion appears in log coordinates. A Beta density satisfies

$$\log p(t) = c + (\mu - 1)\log t + (\nu - 1)\log(1-t)\,,$$

which is precisely the linear basis used in the fitting rule in equation 22. This basis is expressive enough to represent uniform sampling, early-time concentration, late-time concentration, and interior concentration, while remaining simple enough to estimate online from a small number of statistics.

B.3 Fitted $\sqrt{\hat{V}_j}$

Given our approach having a heuristic argument, we overlays the empirical binned $\hat{V}(t)$ against the fitted $t^a(1-t)^b$ to check the assumption. We do this in Figure 3, showing that the general behavior of the bin-wise variances can be well captured through fitting a Beta distribution, making it a natural fit for the method.

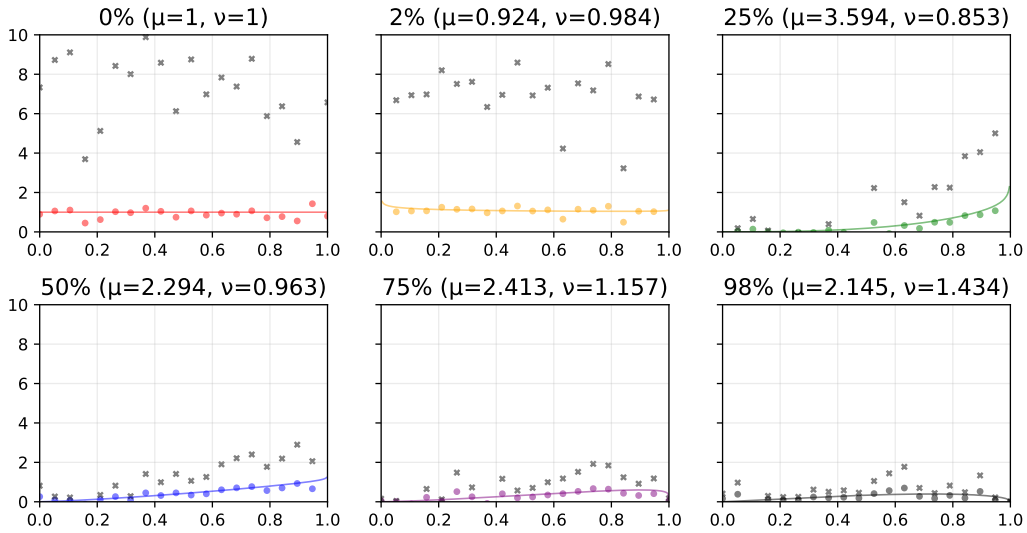

Figure 3: Overlay of fitted $p(t)$. Colored lines designate the fitted $p^*(t)$ functions against the bin-wise variances (black x markers) and the normalized bin-wise variances (colored dots). The Beta distribution captures the general behavior in the variance, highlighting the suitability of the model. Checkpoints are taken from the same points during training as in Figure 2.

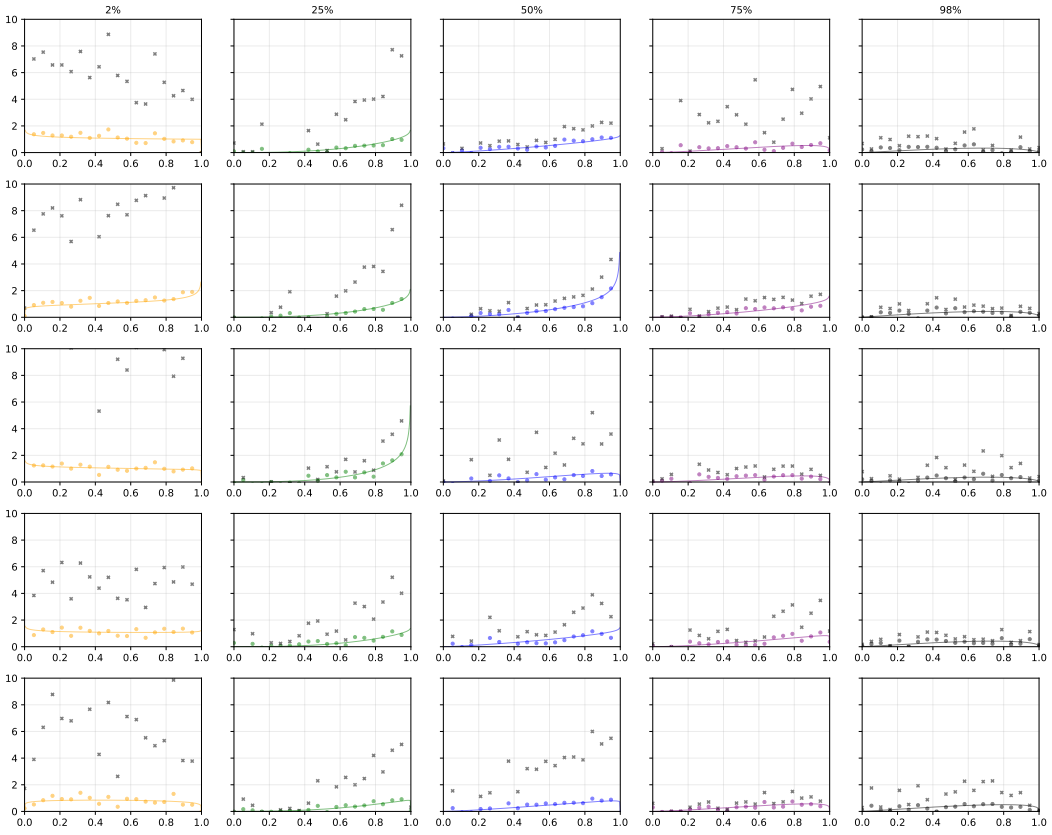

Figure 4: Overlay of fitted $p(t)$. Colored lines designate the fitted $p^*(t)$ functions against the bin-wise variances (black x markers) and the normalized bin-wise variances (colored dots). The Beta distribution captures the general behavior in the variance, highlighting the suitability of the model. Checkpoints are taken from the same points during training as in Figure 2. Plotted for multiple seeds (rows) for the same model (500M `LLaDA` model variant).

C Additional Experimental Details and Results

C.1 Technical Details For Reproducibility

All experiments were conducted on a single machine with 4 NVIDIA H100 80GB GPUs. Experiments were run in an environment using CUDA version 12.8 and PyTorch 2.7.1. Our experiments were adapted from the `dllm` repository[2] at commit `0dec970`.

C.2 Results on Pre-Trained Checkpoints

For further confirmation of our method, we do additional tuning on pre-existing checkpoints. We run these experiments using the pre-trained base checkpoints and conduct an SFT phase using the `Tulu3Mixture` dataset provided by Lambert et al. (2024) and follow their hyper-parameter setup.

C.2.1 SFT on `LLaDA-8B`

Table 7: SFT Results on a `LLaDA-8B` model. We report SFT results from training on the provided checkpoint, comparing the use of our approach against the standard uniform sampling approach. The top two rows represent results directly from Nie et al. (2025b) unless denoted with *, which indicates the result was evaluated by us directly.

	MMLU	MMLU-Pro	ARC-c	Hella.	GSM8K	Math	GPQA	HumanEval	MBPP
`LLaDA-8B-Base`	65.9	36.5*	45.9	70.5	70.3	31.4	25.2	35.4	40.0
`LLaDA-8B-SFT`	65.5	37.0	88.5	74.6	69.4	31.9	33.3	49.4	41.0
Uniform	65.6	36.5	54.1	69.2	70.7	31.6	28.9	38.9	40.4
`NoNoMoAware`	65.6	36.6	58.3	72.2	70.5	31.7	30.4	39.4	40.3

C.2.2 SFT on `ModernBERT-Large`

Table 8: SFT Results on a `ModernBERT-large` model. We report SFT results from training on the provided checkpoint, comparing the use of our approach against the standard uniform sampling approach. The top two rows represent results obtained from directly evaluating `ModernBERT-large` and `ModernBERT-large-chat-v0.1` models.

	GSM8K	BBH	MATH	MMLU	Hella.	LMB.	Wino.	CEval	CMMLU
`ModernBERT-large`	0.1	20.4	0.1	24.6	25.6	38.6	51.3	25.5	24.1
`ModernBERT-large-chat`	9.3	25.6	3.6	29.6	40.9	46.3	53.1	26.5	26.5
Uniform	5.6	23.4	2.9	26.3	35.3	44.2	52.5	25.6	25.8
`NoNoMoAware`	6.1	23.3	2.8	28.3	37.2	44.6	52.4	25.7	26.0

C.3 Ablation on Number of Masked Tokens

Learning on masked tokens introduces an implicit bias, that being that the number of losses computed is no longer fixed per token but dependent on the masking rate. This introduces an additional question regarding training: does performance scale with the number of masked tokens seen during training, rather than the number of samples.

To verify this, we compare two regimes that observe the same number of masking tokens (in expectation), but with different schedules. Each uses fixed masked rates of 30% and 60% on masking on a dataset of 30B and 15B. In expectation, each will observe roughly 9B masked tokens during training. We keep all training hyper-parameters the same except for the total number of training steps; all other relevant hyper-parameters are set relative to the total number of training steps. We observe in Table 9 that while the more total masked tokens seen does appear to prove training, using larger masking rates directly on the same dataset does not directly lead to major improvements (through comparison of training at 60% masking and 30% masking on the same 15B dataset).

[2] https://github.com/ZHZisZZ/dllm

Table 9: Controlling for the number of masked tokens.

Strategy Rate (Tokens)	MMLU acc	BBH acc	Hella. acc	ARC-c acc	TQA acc	Wino. acc	PIQA acc	GSM8K acc	Math acc	GPQA acc
30% (30B)	45.42	35.51	32.53	52.54	35.54	54.54	52.54	55.35	31.53	25.33
60% (15B)	45.34	35.44	32.65	52.75	35.53	54.46	52.65	55.45	31.58	25.34
30% (15B×2)	45.35	35.47	32.36	52.44	35.82	54.74	52.98	55.27	31.42	25.22
30% (30B)	45.22	35.35	32.34	52.65	33.35	54.67	52.93	54.95	30.83	25.02

C.4 Ablation on Form of Distribution

To further validate the usage of a Beta distribution, we compare alternative forms of time-step sampling. We compare against a learnable gaussian approach, as well as candidate fixed Gaussian distributions, namely $\mathcal{N}(0.6, 0.25)$, $\mathcal{N}(0.75, 0.25)$ and $\mathcal{N}(0.75, 0.1)$ distributions. For the learnable gaussian, we simply use the time-step as the mean of the normal distribution μ and fix the standard deviation σ. In this case, we clip the sampled time-steps to a range $[\epsilon, 1-\epsilon]$ to ensure stability.

Table 10: Comparing Beta with other approaches.

Distribution	MMLU	Hella.	ARC-c	GPQA
Unif	48.42	57.70	62.44	25.10
Ours	49.38	58.34	62.64	25.93
Learnable Gaussian ($\sigma = 0.10$)	47.85	56.84	62.58	25.09
Learnable Gaussian ($\sigma = 0.25$)	47.58	56.65	61.86	25.14
$\mathcal{N}(0.60, 0.25)$	48.44	57.84	62.43	25.65
$\mathcal{N}(0.75, 0.10)$	47.92	57.62	62.05	25.03
$\mathcal{N}(0.75, 0.25)$	47.94	57.33	61.96	24.96

Table 10 demonstrates that the Beta fit that we propose is indeed more performative than other possible alternatives such as normal distributions centered around a fixed mean. Additionally, there is further evidence that solely using larger noising time-steps does not necessarily always lead to greater performance, as we observe how centering around a more moderate noising rate (e.x. 0.6) can be more effective than more severe corruption rates. Furthermore, we can observe that the learnable strategy described above does reach the same performance as our method, likely due to the lack of direct ability to fit the variation.

C.5 Comparison to Sampling with Respect to Probabilities

Due to the discrete nature at which we measure $\sqrt{\hat{V}_j}$, we run a result on the following.

First, we run SFT run repeatedly on the `allenai/tulu-3-sft-mixture` dataset on a `answerdotai/ModernBERT-base` checkpoint. We tune for 10 epochs on this dataset. We repeat this for 5 seeds and report mean $\pm$ standard deviation.

Our method works as follows: we first measure the variances using the histogram bin center points and compute a normalized probability distribution based on it. Depending on the sampled bin, we then randomly select a t value inside the range associated with the specific bin.

	GSM8K	BBH	MATH	MMLU	HellaSwag	LAMBADA	Winogrande	CEval	CMMLU
Uniform	$6.76_{\pm 0.14}$	$25.67_{\pm 0.65}$	$6.57_{\pm 0.09}$	$30.66_{\pm 0.66}$	$37.34_{\pm 0.78}$	$51.54_{\pm 0.53}$	$51.64_{\pm 0.35}$	$28.54_{\pm 1.13}$	$28.54_{\pm 0.94}$
Ours	$7.53_{\pm 0.27}$	$26.88_{\pm 0.59}$	$6.64_{\pm 0.11}$	$32.54_{\pm 0.75}$	$39.45_{\pm 0.54}$	$52.38_{\pm 0.47}$	$52.83_{\pm 0.54}$	$28.45_{\pm 0.84}$	$28.67_{\pm 1.04}$
$\sqrt{\hat{V}_j}$	$7.22_{\pm 0.35}$	$25.76_{\pm 0.58}$	$6.61_{\pm 0.19}$	$31.48_{\pm 0.54}$	$38.15_{\pm 0.45}$	$51.86_{\pm 0.56}$	$52.68_{\pm 0.47}$	$28.65_{\pm 1.04}$	$28.45_{\pm 0.88}$

Overall, we can see that using some sampling from $\sqrt{\hat{V}_j}$ can improve upon the uniform sampling approach, while using our additional weighting scheme we can further improve on this on most tasks.

C.6 Training Dynamics

We further provide training plots to demonstrate the gradient-estimator variance claims for our approach, as well as general training loss curves to demonstrate the general training improvements brought about in our approach. For the former, we compute the individual variance in losses within each bin and then compute the total variance across these individual bins.

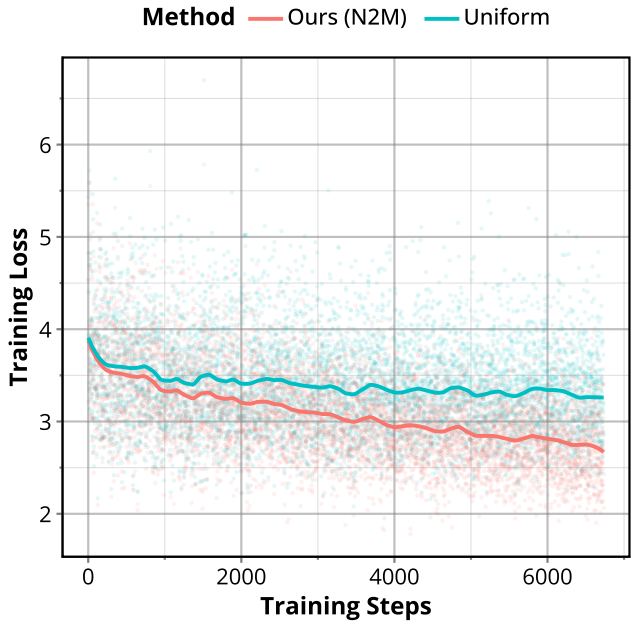

Figure 5: Pre-training loss of a 500M parameter LLaDA model.

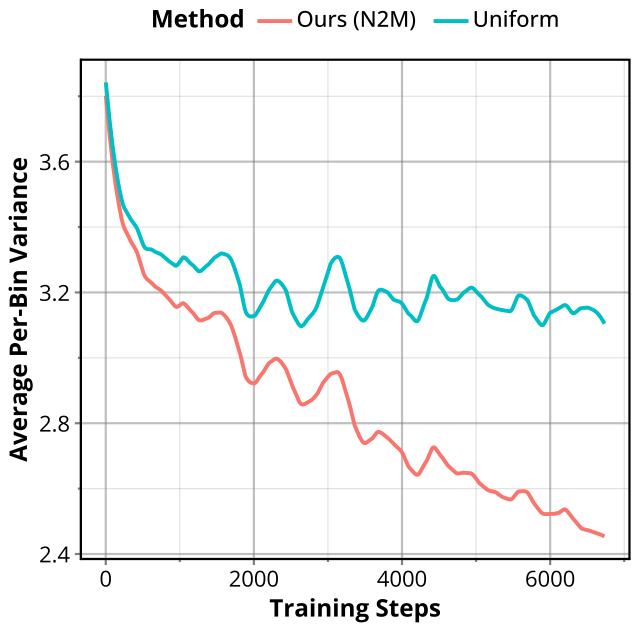

Figure 6: Average-per bin variance in loss of a 500M parameter LLaDA model during pre-training.

D Extended Background

D.1 Diffusion Language Modeling

Diffusion models are trained to undo a forward corruption process q that takes clean data and defines a latent variable $\mathbf{z}_t$ for $t \in [0,1]$ that represents progressively noisy versions of a data point $\boldsymbol{x}$ (Ho et al., 2020; Sohl-Dickstein et al., 2015; Song et al., 2021b; Wang et al., 2023; Sahoo et al., 2024b). The standard forward process for a continuous $\boldsymbol{x} \in \mathbb{R}^d$ is defined by

$$\mathbf{z}_t = \sqrt{\alpha(t)}\boldsymbol{x} + \sqrt{1-\alpha(t)}\boldsymbol{\epsilon}, \tag{87}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_d, \boldsymbol{I}_{d\times d})$ and $\alpha\left(t\right)$ is a noise schedule monotonically decreasing in $t \in [0,1]$. The parameterized reverse diffusion model p_θ over $\boldsymbol{x}$ and $\mathbf{z}_t$ is trained to maximize a variational lower bound on log-likelihood (ELBO) (Kingma and Welling, 2014). Given a number of discretization steps T, defining $s(i) = \left(i-1\right)/T$ and $t(i) = i/T$, the Negative ELBO (NELBO) equals

$$\mathbb{E}_q[\underbrace{-\log p_\theta\big(\boldsymbol{x}|\mathbf{z}_{t(0)}\big)}_{\mathcal{L}_{\text{recon}}} + \underbrace{\sum_{i=1}^{T} D_{\text{KL}}\big(q\big(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)},\boldsymbol{x}\big)\big\|p_\theta\big(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)}\big)\big)}_{\mathcal{L}_{\text{diff}}}] + \underbrace{D_{\text{KL}}\big(q\big(\mathbf{z}_{t(t)}|\boldsymbol{x}\big)\big\|p_\theta\big(\mathbf{z}_{t(t)}\big)\big)}_{\mathcal{L}_{\text{prior}}} \tag{88}$$

where D_{KL} is the Kullback-Leibler divergence. For simplicity, i can be dropped from $s(i)$ and $t(i)$; in general, s denotes the time step before t. However, this process acts in a continuous space, i.e. the samples $\boldsymbol{x}$ are sampled from a continuous multi-dimensional distribution, meaning there remains a gap to resolve when applying it to tasks where inputs are inherently discrete, such as language.

Discrete diffusion models can be broken down into two categories. The first are settings where discrete structures are embedded in continuous space and then use the above Gaussian diffusion processes on these continuous representations (Chen et al., 2023; Dieleman et al., 2022; Gulrajani and Hashimoto, 2023; Han et al., 2023; Li et al., 2022; Lovelace et al., 2023; Strudel et al., 2022). The other is settings where the diffusion process is directly defined on discrete structures (Lou et al., 2024; Ou et al., 2025; Sahoo et al., 2024a). Austin et al. (2021) first introduce a framework with a Markov forward process $q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \text{Cat}(\mathbf{z}_t; Q_t\mathbf{z}_{t-1})$ defined by the multiplication of matrices Q_t over T discrete time steps. This induces marginals

$$q(\mathbf{z}_t|\boldsymbol{x}) = \text{Cat}\big(\mathbf{z}_t; \bar{Q}_t\boldsymbol{x}\big) = \text{Cat}(\mathbf{z}_t; Q_tQ_{t-1}\cdots Q_1\boldsymbol{x}) \tag{89}$$

that represent the discrete-state form of Equation 87. This can be extended to continuous time through continuous time Markov chain theory (Campbell et al., 2022), which leads to generalizations in the score matching perspective on diffusion modeling (Song et al., 2021b) to discrete data (Lou et al., 2024; Sun et al., 2023). Lou et al. (2024) in particular connect score-based approaches with ELBO maximization to enable performant likelihood-based training score-based models. In these settings, *masked diffusion* (Nie et al., 2025a; Shi et al., 2024; He et al., 2023) has consistently achieved greater performance.

To understand masked diffusion, we re-use our notation from auto-regressive language modeling. Further denote $\text{Cat}(\cdot;\boldsymbol{\pi})$ as the categorical distribution over the K classes with probabilities given by $\boldsymbol{\pi} \in \Delta^K$ (the K-simplex). We assume the K-th category to correspond to a special `[MASK]` token and $\boldsymbol{m} \in \mathcal{V}$ be its one-hot vector. From before, the marginal of $\mathbf{z}_t$ conditioned on $\boldsymbol{x}$ at time t is

$$q(\mathbf{z}_t|\boldsymbol{x}) = \text{Cat}(\mathbf{z}; \alpha_t\boldsymbol{x} + (1-\alpha_t)\,\boldsymbol{\pi})\,,$$

where $\alpha_t \equiv \alpha\left(t\right) \in [0,1]$ is a strictly decreasing function with $\alpha_0 \approx 1$ and $\alpha_1 \approx 0$. This implies forward transition probabilities

$$q(\mathbf{z}_t|\mathbf{z}_s) = \text{Cat}\big(\mathbf{z}; \alpha_{t|s}\mathbf{z}_s + (1-\alpha_{t|s})\boldsymbol{\pi}\big)\,,\ \alpha_{t|s} = \alpha_t/\alpha_s. \tag{90}$$

Thus for each diffusion step $s \to t$, a fraction $(1-\alpha_{t|s})$ of the probability mass is transferred to the prior distribution $\boldsymbol{\pi}$. The reverse posterior is then

$$q(\mathbf{z}_s|\mathbf{z}_t,\boldsymbol{x}) = \text{Cat}\left(\mathbf{z}_s; \frac{\left(\alpha_{t|s}\mathbf{z}_t + (1-\alpha_{t|s})\mathbf{1}\boldsymbol{\pi}^\top\mathbf{z}_t\right) \odot \left(\alpha_s\boldsymbol{x} + (1-\alpha_s)\boldsymbol{\pi}\right)}{\alpha_t\mathbf{z}_t^\top\boldsymbol{x} + (1-\alpha_t)\,\mathbf{z}_t^\top\boldsymbol{\pi}}\right). \tag{91}$$

These yield a simplified variational lower bound objective with lower variance during training. One can set $\boldsymbol{\pi} = \boldsymbol{m}$. At each noising step t, the input $\boldsymbol{x}$ transitions to a masked state $\boldsymbol{m}$ with some probability. When $\boldsymbol{x}$ transitions to $\boldsymbol{m}$ at time t', for any time $t > t'$ it remains in that state, i.e. $q(\mathbf{z}_t|\mathbf{z}_{t'} = \boldsymbol{m}) = \mathrm{Cat}(\mathbf{z}_t; \boldsymbol{m})$. By convention, at time T all inputs are masked with certainty. Using properties of the masking process, Equation 91 simplifies to

$$q(\mathbf{z}_s|\mathbf{z}_t, \boldsymbol{x}) = \begin{cases} \mathrm{Cat}(\mathbf{z}_s; \mathbf{z}_t) & \mathbf{z}_t \neq \boldsymbol{m} \\ \mathrm{Cat}\left(\mathbf{z}_s; \frac{(1-\alpha_s)\boldsymbol{m} + (\alpha_s - \alpha_t)\boldsymbol{x}}{1-\alpha_t}\right) & \mathbf{z}_t = \boldsymbol{m} \end{cases}. \tag{92}$$

For the reverse process, a finite number of steps T is again considered with a continuous time model corresponding to $T \to \infty$. In the discrete-time case, the generative model is expressed as

$$p_\theta(\boldsymbol{x}) = \int_{\mathbf{z}} p_\theta(\mathbf{z}_1)\, p_\theta(\boldsymbol{x}|\mathbf{z}_0) \prod_{i=1}^{T} p_\theta(\mathbf{z}_s|\mathbf{z}_t)\, \mathrm{d}\mathbf{z}_{0:1}.$$

The optimal form of $p_\theta(\mathbf{z}_s|\mathbf{z}_t)$ matches the true posterior in Equation 92 by Equation 88, which is a sum of terms of the form $D_{\mathrm{KL}}(q(\mathbf{z}_s \mid \mathbf{z}_t, \boldsymbol{x}) \| p_\theta(\mathbf{z}_s \mid \mathbf{z}_t))$. However, Equation 92 is conditioned on $\boldsymbol{x}$, which is unknown and therefore a model $\boldsymbol{x}_\theta(\mathbf{z}_t, t) : \mathcal{V} \times [0, 1] \to \Delta^K$ is introduced to approximate $\boldsymbol{x}$ with a neural network with two restrictions: (1) $\langle \boldsymbol{x}_\theta(\mathbf{z}_t, t)\,, \boldsymbol{m} \rangle = 0$ by definition and (2) $\boldsymbol{x}_\theta(\mathbf{z}_t, t) = \mathbf{z}_t$ for any unmasked $\mathbf{z}_t$. Under this parameterization

$$p_\theta(\mathbf{z}_s|\mathbf{z}_t) = q(\mathbf{z}_s|\mathbf{z}_t, \boldsymbol{x} = \boldsymbol{x}_\theta(\mathbf{z}_t, t)) = \begin{cases} \mathrm{Cat}(\mathbf{z}_s; \mathbf{z}_t) & \mathbf{z}_t \neq \boldsymbol{m} \\ \mathrm{Cat}\left(\mathbf{z}_s; \frac{(1-\alpha_s)\boldsymbol{m} + (\alpha_s - \alpha_t)\boldsymbol{x}_\theta(\mathbf{z}_t, t)}{1-\alpha_t}\right) & \mathbf{z}_t = \boldsymbol{m} \end{cases}. \tag{93}$$

This simplifies the discrete-time diffusion loss for finite T to

$$\mathcal{L}_{\text{diff}} = \sum_{i=1}^{T} \mathbb{E}_q \left[D_{\mathrm{KL}}\left(q\left(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)}, \boldsymbol{x}\right) \middle\| p_\theta\left(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)}\right)\right)\right] \tag{94}$$

$$= \sum_{i=1}^{T} \mathbb{E}_q \left[\frac{\alpha_{t(i)} - \alpha_{s(i)}}{1 - \alpha_{t(i)}} \log \left\langle \boldsymbol{x}_\theta\left(\mathbf{z}_{t(i)}, t\right), \boldsymbol{x} \right\rangle \right]. \tag{95}$$

Forming a continuous time extension of Equation 94 by taking $T \to \infty$

$$\mathcal{L}^{\infty}_{\text{NELBO}} = \mathbb{E}_q \left[\int_0^1 \frac{\alpha'_t}{1 - \alpha_t} \log \left\langle \boldsymbol{x}_\theta(\mathbf{z}_t, t)\,, \boldsymbol{x} \right\rangle \mathrm{d}t \right]. \tag{96}$$

When the above strategy is applied to language modeling over sequences $\boldsymbol{X}$ from a vocabulary $\mathcal{V}$, a common implicit assumption is that the forward noising process is applied independently across the sequence and the denoising process factorizes independently across tokens conditioned on the latents $\mathbf{Z}_t = [\mathrm{z}_t^1, \ldots, \mathrm{z}_t^L]$ of each token. Thus a single model can be used to compute each $\boldsymbol{x}_\theta^\ell(\mathbf{Z}_t, t)$ for each ℓ from a masked sequence $\mathbf{Z}$, optimizing

$$\mathcal{L}^{\infty}_{\text{NELBO}} = \mathbb{E}_q \left[\int_0^1 \frac{\alpha'_t}{1 - \alpha_t} \sum_{\ell=1}^{L} \log \left\langle \boldsymbol{x}_\theta^\ell(\mathbf{Z}_t, t)\,, \boldsymbol{x}^\ell \right\rangle \mathrm{d}t \right]. \tag{97}$$

This process, called *masked diffusion language modeling* (MDLM), has a simple form in that it is a weighted average over fixed-rate MLM-style losses (Devlin et al., 2019; Warner et al., 2025). In particular, one can note that unmasked tokens do not contribute to the loss due to being copied over by the denoising network, reducing the $\log \left\langle \boldsymbol{x}_\theta^\ell(\mathbf{Z}_t, t)\,, \boldsymbol{x}^\ell \right\rangle$ term to zero. For computational simplicity, t is often sampled uniformly within the range $[0, 1]$, leading to an ELBO formulation of

$$\mathcal{L}^{\infty}_{\text{MDLM}} = \mathbb{E}_q \left[\int_0^1 \frac{1}{t} \sum_{\ell=1}^{L} \log \left\langle \boldsymbol{x}_\theta^\ell(\mathbf{Z}_t, t)\,, \boldsymbol{x}^\ell \right\rangle \mathrm{d}t \right]. \tag{98}$$