TUBE: Tangent Upper Bound on Evidence for Discrete Diffusion Language Models

Anonymous Authors[1]

Abstract

Log-likelihood is a standard metric for evaluating generative models. Unfortunately, in contrast to autoregressive models (ARMs), discrete diffusion models generally do not admit exact computation of this quantity. Existing evaluations, therefore, rely on the evidence lower bound (ELBO), leaving unclear how much higher the true value may be. We address this by introducing the **Tangent Upper Bound on Evidence** (**TUBE**), a variational upper bound on log-likelihood that admits an unbiased Monte Carlo estimator. Our TUBE extends across latent-variable models, including masked diffusion models (MDMs), any-order ARMs (AO-ARMs), and block variants of both. Applied to block MDMs and block AO-ARMs, TUBE reveals our key empirical finding that these models lie strictly below the exact ARM baseline, showing that ARMs still dominate in likelihood.

1 Introduction

Autoregressive models (ARMs) remain a central paradigm in language modeling, supported by strong empirical scaling behavior in large-scale settings (Kaplan et al., 2020; Hoffmann et al., 2022). However, this efficiency arises from imposing a fixed autoregressive decomposition, which makes generation inherently order-dependent. At the same time, the best ordering typically depends on the domain and task, and learned or alternative orderings can outperform standard fixed choices in practice (Li et al., 2021; Wang et al., 2025b). This has motivated a growing line of work on *any-order autoregressive models* (AO-ARMs) (Uria et al., 2014; Shih et al., 2022) and related *masked diffusion models* (MDMs) (Austin et al., 2021; Shi et al., 2024; Sahoo et al., 2024), which both replace a single fixed decomposition with a probabilistic family of generation orderings.

Despite their different theoretical constructions, AO-ARMs and MDMs both generalize beyond left-to-right order by repeatedly selecting token positions at random and filling in the corresponding values. MDMs further extend this procedure by allowing multiple tokens to be generated in a single step, enabling faster inference. Beyond these fully random ordering schemes, blockwise generation provides a mixed alternative (Arriola et al., 2025), using a fixed autoregressive ordering over blocks and a random ordering within each block. This retains block-level autoregressive structure, enabling techniques such as KV caching. Together, these features make AO-ARMs and MDMs viable alternatives to strictly autoregressive generation. More broadly, both have been applied beyond standard left-to-right language modeling to other discrete domains, including images (Pang et al., 2025; Austin et al., 2021), graphs (Kelvinius & Lindsten, 2024; Seo et al., 2025), molecular sequences (Lee et al., 2025), and vector-quantized image representations (Gu et al., 2022). In language modeling, recent large-scale MDMs have been shown to be competitive in the generation quality to ARMs (Nie et al., 2025; Bie et al., 2025; Ye et al., 2025; Karimi Monsefi et al., 2026).

At the same time, principled **evaluation** of such models calls for **log-likelihood**, which remains the canonical measure of distributional fit. While ARM enables exact log-likelihood evaluation, for AO-ARM and MDM, on the other hand, this quantity is typically intractable. It is instead replaced by the *evidence lower bound* (ELBO), or related approximations (Sahoo et al., 2024; Haxholli et al., 2025; Jeon et al., 2025). However, the ELBO does not indicate how much higher the true log-likelihood may be. As a result, ELBO-based evaluation alone cannot reliably assess the likelihood of AO-ARMs and MDMs, nor support rigorous comparison with ARM baselines.

Recent work has made this issue increasingly explicit both for MDMs and for broader classes of latent-variable models. For MDMs, exact log-likelihood evaluation is available only in special cases, such as deterministic unmasking (Turok et al., 2026), while recent work has also proposed upper-bound estimator (Wang et al., 2026). In parallel, the variational-inference literature has developed upper-bound

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the FoGen Workshop at ICML 2026. Do not distribute.

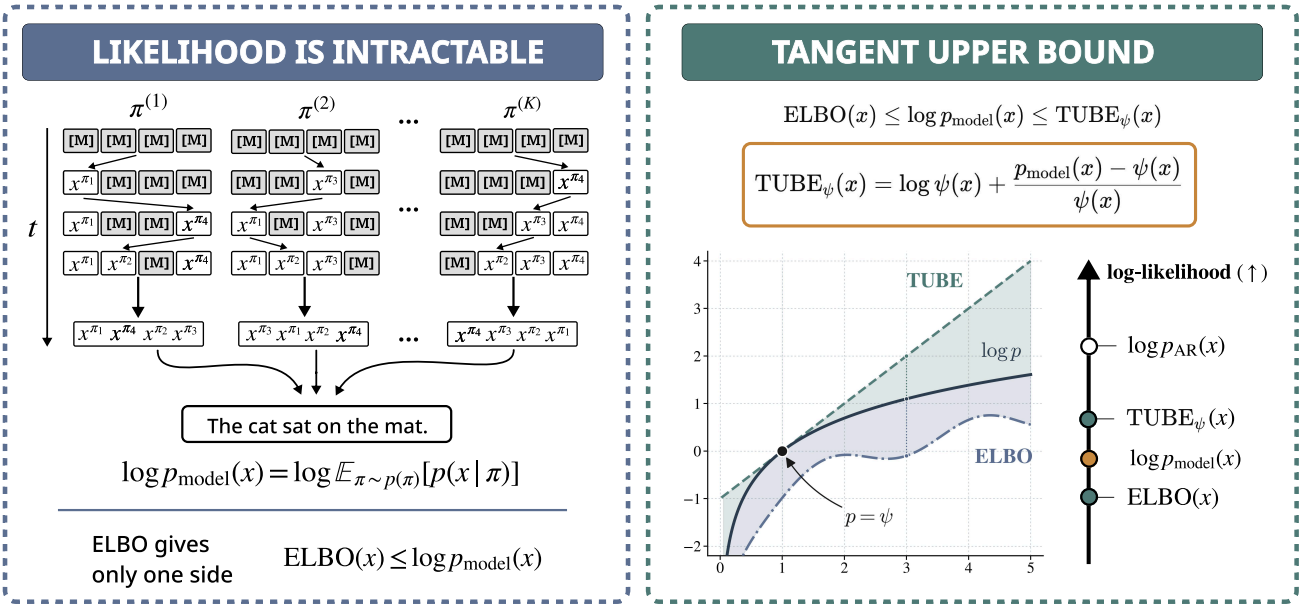

Figure 1. **A tight tangent upper bound on** $\log p_{\text{model}}(x)$**.** Our TUBE bounds the intractable marginal from above using a tractable surrogate ψ, with equality at $\psi = p_{\text{model}}$.

approaches for estimating log-likelihood in general latent-variable models (Dieng et al., 2017; Struski et al., 2023). In practice, however, exact methods apply only in narrow fixed-order settings, and upper-bound estimators are hard to use reliably because they become biased when a nonlinear function is applied after Monte Carlo approximation. This leaves open the problem of reliable finite-sample log-likelihood estimation for AO-ARMs and MDMs, which we address in this paper. Our **contributions** are as follows:

- **Method.** We propose the *Tangent Upper Bound on Evidence* (TUBE), a variational pointwise upper bound on the log-likelihood with a tractable surrogate and an unbiased Monte Carlo estimator (§3). Together with the ELBO, TUBE yields a two-sided localization of the log-likelihood.
- **Analysis.** We evaluate pretrained block AO-ARMs and MDMs and identify a clear empirical likelihood gap relative to standard ARM baselines (§5).

Notation. We write $\mathcal{V} = \{1, \dots, V\}^L$ for the space of length-L token sequences and $x = (x^1, \dots, x^L) \in \mathcal{V}$ for a data sample. We use p to denote distributions, with subscripts indicating the corresponding distribution when needed, e.g., p_{model}. The generation ordering is denoted by π. By $\mathcal{S}$ we denote single-token orderings, where each step reveals one position, and $\mathcal{G}$ denotes grouped orderings, where a step may reveal multiple positions. For an index set π_t at step $t \in \{1, \dots, T\}$, x^{π_t} denotes the tokens of x at positions in π_t, and $\pi_{<t}$ denotes the positions revealed before that step.

2 Background

In this section, we first recall ARMs, including their definition and training procedure (§2.1). We then generalize this view to AO-ARMs (§2.2) and relate them to MDMs (§2.3). Finally, we introduce block variants of the latter two (§2.4) and discuss the challenges of likelihood evaluation (§2.5).

2.1 Autoregressive models (ARM)

We consider the problem of generating a sequence $x = (x^1, \dots, x^L) \in \mathcal{V}$ of length L, where each x^l denotes a token. ARMs $p_{\text{ARM}}(x)$ model this sequence by generating one token at each step $t \in \{1, \dots, L\}$ according to a fixed generation order $\pi \in \mathcal{S}$:

$$\begin{aligned} p_{\text{ARM}}(x) &= p(x^{\pi_1}, \dots, x^{\pi_L}) \\ &= p(x^{\pi_1}) \dots p(x^{\pi_L} | x^{\pi_{L-1}}, \dots, x^{\pi_1}) \\ &= \prod_{t=1}^{L} p(x^{\pi_t} | x^{\pi_{<t}}), \end{aligned} \tag{1}$$

where π_t denotes the single position generated at step t, so the full sequence x is generated in L steps.

For ARMs, log-likelihood can be evaluated directly due to the fixed decomposition in (1):

$$\log p_{\text{ARM}}(x) = \sum_{t=1}^{L} \log p(x^{\pi_t} | x^{\pi_{<t}}). \tag{2}$$

Common choices of π include left-to-right orderings for text (Radford & Narasimhan, 2018) and raster-scan orderings for

images (Chen et al., 2020). Fixed orderings make training and log-likelihood evaluation tractable. However, they force the model to generate variables in one prescribed order, even when other orders may be equally natural or more suitable for the data.

2.2 Any-order autoregressive models (AO-ARM)

To reduce dependence on a fixed order, AO-ARMs (Uria et al., 2014; Shih et al., 2022) model the data distribution by averaging over autoregressive decompositions induced by different orderings:

$$\begin{aligned} p_{\text{AO-ARM}}(x) &= \mathbb{E}_{\pi\sim p(\pi)}\left[p(x|\pi)\right] \\ &= \mathbb{E}_{\pi\sim p(\pi)}\left[\prod_{t=1}^{L} p(x^{\pi_t}|x^{\pi_{<t}})\right], \end{aligned} \tag{3}$$

where $p(\pi)$ is typically a uniform distribution over orderings $\pi \in \mathcal{S}$, and generation takes L steps.

From this perspective, AO-ARMs can be viewed as latent-variable models in which the ordering π is latent, thereby increasing modeling flexibility. Unfortunately, exact log-likelihood computation is generally infeasible due to the large size of $\mathcal{S}$. One may therefore approximate the expectation in (3) using Monte Carlo, but placing this approximation inside the logarithm yields a biased estimator of the log-likelihood. Thus, in previous works (Uria et al., 2014; Hoogeboom et al., 2022), log-likelihood estimation is typically based on ELBO:

$$\begin{aligned} \log p_{\text{AO-ARM}}(x) &= \log \mathbb{E}_{\pi\sim p(\pi)}\left[\prod_{t=1}^{L} p(x^{\pi_t}|x^{\pi_{<t}})\right] \\ &\geq \mathbb{E}_{\pi\sim p(\pi)}\left[\sum_{t=1}^{L} \log p(x^{\pi_t}|x^{\pi_{<t}})\right]. \end{aligned} \tag{4}$$

This latent-order perspective also provides a natural bridge to discrete diffusion models.

2.3 Masked diffusion models (MDM)

In MDMs (Austin et al., 2021; Shi et al., 2024; Sahoo et al., 2024), the latent variable can be interpreted as an unmasking trajectory, or equivalently as a grouped ordering; see Figure 2. We write an unmasking trajectory as $\pi = (\pi_1, \ldots, \pi_T) \in \mathcal{G}$, where $\mathcal{G}$ denotes the set of *grouped orderings*. At each generation step $t \in \{1, \ldots, T\}$, the set $\pi_t \subseteq \{1, \ldots, L\}$ contains the positions revealed at that step. The sets π_t are pairwise disjoint, may be empty, and cover all positions, i.e., $\bigsqcup_{t=1}^{T} \pi_t = \{1, \ldots, L\}$. Thus, a step may reveal zero, one, or multiple positions, enabling parallel token generation. To make such parallel updates tractable, the joint distribution over the revealed tokens is typically factorized over positions, $p(x^{\pi_t}|x^{\pi_{<t}}, \pi) = \prod_{d\in\pi_t} p(x^d|x^{\pi_{<t}}, \pi)$. With this parameterization, MDMs mirror AO-ARMs by averaging over grouped generation orderings:

$$\begin{aligned} p_{\text{MDM}}(x) &= \mathbb{E}_{\pi\sim p(\pi)}\left[p(x|\pi)\right] \\ &= \mathbb{E}_{\pi\sim p(\pi)}\left[\prod_{t=1}^{T}\prod_{d\in\pi_t} p(x^d|x^{\pi_{<t}}, \pi)\right], \end{aligned} \tag{5}$$

where $p(\pi)$ is the distribution over unmasking trajectories induced by the chosen masking process. Generation proceeds in T steps, which may be smaller or larger than L.

Notably, the grouped-ordering view also clarifies the relation to AO-ARMs in continuous time. In models such as (Sahoo et al., 2024), each position has a continuous generation time $t \in [0, T]$, so different positions are revealed at the same time with probability zero. After sorting the reveal events and removing empty intervals, the trajectory reduces to single-position groups. As a result, generation proceeds in an AO-ARM-like manner, revealing one token at a time according to a random ordering $\pi \in \mathcal{S}$ induced by the masking process.

Regarding log-likelihood estimation, in continuous time one can use the standard MDM ELBO, e.g., (Ou et al., 2025, Eq. 2.6), which is notably equivalent to the AO-ARM ELBO in (4) (Ou et al., 2025, Thm. 2). For discretized-time MDMs, however, the relation to AO-ARMs yields a similar ELBO, but over grouped trajectories $\pi \in \mathcal{G}$:

$$\begin{aligned} \log p_{\text{MDM}}(x) &= \log \mathbb{E}_{\pi\sim p(\pi)}\left[\prod_{t=1}^{T}\prod_{d\in\pi_t} p(x^d|x^{\pi_{<t}}, \pi)\right] \\ &\geq \mathbb{E}_{\pi\sim p(\pi)}\left[\sum_{t=1}^{T}\sum_{d\in\pi_t} \log p(x^d|x^{\pi_{<t}}, \pi)\right]. \end{aligned} \tag{6}$$

2.4 Block models (BM)

BMs (Arriola et al., 2025) provide another way to combine the tractability of ARMs with the flexibility of AO-ARMs and MDMs. Rather than defining a single ordering over all positions, they split the sequence into B blocks of size L', denoted by disjoint position sets $\mathcal{B}_1, \ldots, \mathcal{B}_B$ with $\bigsqcup_{b=1}^{B} \mathcal{B}_b = \{1, \ldots, L\}$. This gives the following block-level decomposition:

$$p_{\text{BM}}(x) = \prod_{b=1}^{B} p(x^{\mathcal{B}_b}|x^{\mathcal{B}_{<b}}), \tag{7}$$

where $p(x^{\mathcal{B}_b}|x^{\mathcal{B}_{<b}})$ is parameterized as in (3) or (5). As a result, the block factorization preserves the autoregressive structure needed for techniques such as KV caching, while the within-block conditionals can still use random-order or parallel generation.

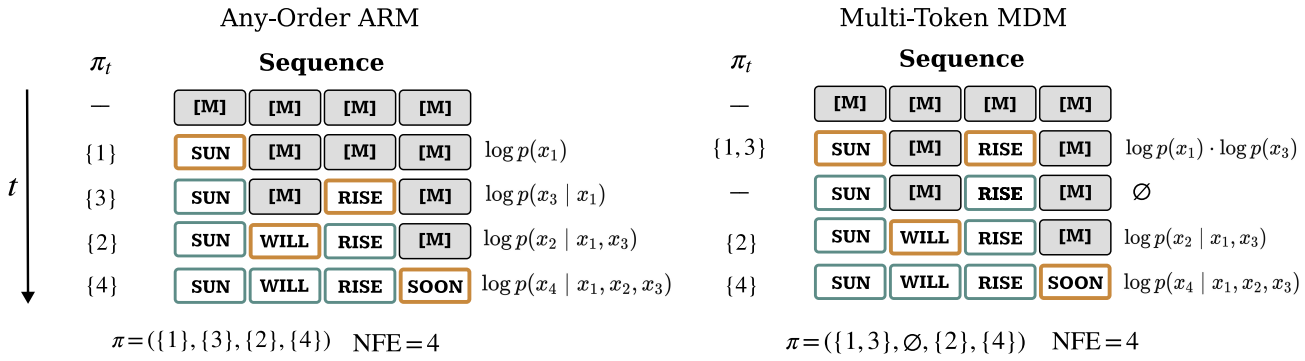

Figure 2. **AO-ARM and MDM.** Both define $p_{\text{model}}(x)$ via a latent generation ordering π. AO-ARM reveals one token per step ($\pi \in \mathcal{S}$). MDM allows zero, one, or many tokens per step ($\pi \in \mathcal{G}$).

In both parameterizations, the decomposition in (7) is restricted to positions inside $\mathcal{B}^b$ and conditioned on the previously generated blocks $x^{\mathcal{B}_{<b}}$. Thus, log-likelihood evaluation reduces to a sum of block conditional log-likelihoods. However, the AO-ARM formulation in (3) and the MDM formulation in (5) still require averaging within each block over local orderings $\pi \in \mathcal{S}$ or grouped orderings $\pi \in \mathcal{G}$, making exact evaluation intractable and requiring the ELBOs from (4) and (6), respectively.

2.5 Likelihood evaluation challenges

Likelihood-based comparison of ARMs, AO-ARMs, and MDMs involves two separate choices: the trained model p_{model} and the estimator used to evaluate its likelihood. For the model, notably, a single network trained with either an AO-ARM or MDM ELBO can be reused under different generation orderings. This is possible because these formulations parameterize the same family of conditional distributions, $p_{\text{model}}(x^d \mid x^{\pi_{<t}}, \pi_{<t})$ for $d \in \pi_t$, with ARMs recovered as the fixed-order special case. Moreover, their training objectives have equivalent minimizers under the connection established by (Ou et al., 2025). For estimator, on the other hand, the ELBO in (4) and (6) do not localize the true log-likelihood of AO-ARMs and MDMs, and therefore cannot support rigorous comparison with ARMs on their own. We therefore introduce a complementary upper bound that, together with the ELBO, localizes the log-likelihood of AO-ARMs and MDMs.

3 Method

We construct our TUBE, a variational upper bound on $\log p(x)$ that, combined with ELBO, localizes the likelihood from both sides. We first state the bound itself (§3.1). Then take a look at its practical implementation for AO-ARM and MDM models in (§3.2).

3.1 Tangent Upper Bound on Evidence (TUBE)

Our method of constructing the upper bound on $\log p(x)$ rests on the concavity of $\log a$ function in a and the tangent majorization of $\log a$ by the taylor decomposition at point b:

$$\log a \leq \log b + \frac{a-b}{b}, \quad \forall\, a, b > 0. \tag{8}$$

The upper bound has b as the variational variable and allows us to construct a linear bound for $\log p(x)$ treating the upper bound at each $a = p(x)$ separately.

Definition 3.1 (Tangent Upper Bound on Evidence)**.** *Consider positive functions $p(x)$ and $\psi(x)$, where $p(x), \psi(x) > 0$ for every $x \in \mathcal{X}$. The following variational bound holds:*

$$\log p(x) \;\leq\; \text{TUBE}_{\psi}(x) \;:=\; \log \psi(x) + \frac{p(x) - \psi(x)}{\psi(x)}, \tag{9}$$

where the function $\psi(x)$ is a auxiliary function, and equality holds iff $\psi(x) = p(x)$.

This bound being based on the logarithm linearization allows for linearization of $p(x)$ term, which in context of latent variable models of type $p(x) = \mathbb{E}_{\pi}[p(x|\pi)]$, see §2, allows for unbiased Monte Carlo estimation of the bound, see §3.2 below. The introduced function $\psi(x)$, which we call **surrogate**, can be any positive function and the tightness of TUBE depends on the closeness of $\psi(x)$ to $p(x)$.

Since TUBE is an upper bound, combining it with any lower bound, such as ELBO or its multi-sample extension ELBO_K, yields a population two-sided localization of $\log p(x)$ for any ψ:

Two-sided localization of log-likelihood

$$\mathrm{ELBO}(x) \;\le\; \log p(x) \;\le\; \mathrm{TUBE}_{\psi}(x). \qquad (10)$$

3.2 TUBE estimation for AO-ARM and MDM

Monte Carlo estimation. While TUBE can applied to any latent variable model $p(x) = \mathbb{E}_z[p(x|z)]$, such as VAE (Kingma & Welling, 2013) or DDPM (Ho et al., 2020), our work focuses on the application of TUBE for AO-ARM (Hoogeboom et al., 2022) and MDM (Sahoo et al., 2024) models. Let us consider the model $p_{\text{model}} = \mathbb{E}_{\pi}[p_{\text{model}}(x|\pi)]$ and its Monte Carlo approximated likelihood, then we can finally construct an **unbiased estimator** for the TUBE_{ψ}:

$$\widehat{\mathrm{TUBE}}_{\psi,K}(x) := \log \psi(x) + \frac{\widehat{p}_{\text{model},K}(x) - \psi(x)}{\psi(x)},$$
$$\widehat{p}_{\text{model},K}(x) := \frac{1}{K}\sum_{k=1}^{K} p_{\text{model}}(x|\pi^{(k)}), \qquad (11)$$

where $\pi^{(1)}, \dots, \pi^{(K)} \overset{\text{i.i.d.}}{\sim} p(\pi)$. This straightforward possibility to construct the unbiased estimator is one of the core properties of our bound. This is a contrast with other upper bounds like CUBO (Dieng et al., 2017) or TVO (Masrani et al., 2019) where the estimation is biased, see Table 1 and discussion in §4. In detail, for AO-ARM case the Monte Carlo $\widehat{p}_{\text{model},K}(x)$ takes the form:

$$\widehat{p}_{\text{model},K}(x) = \frac{1}{K}\sum_{k=1}^{K}\left[\prod_{t=1}^{L} p_{\text{model}}(x^{\pi_t^{(k)}}|x^{\pi_{<t}})\right],$$
$$\pi^{(1)}, \dots, \pi^{(K)} \overset{\text{i.i.d.}}{\sim} p(\pi), \quad \pi^{(k)} \in \mathcal{S} \qquad (12)$$

and in the case of MDM with T steps:

$$\widehat{p}_{\text{model},K}(x) = \frac{1}{K}\sum_{k=1}^{K}\left[\prod_{t=1}^{T}\prod_{d\in\pi_t} p_{\text{model}}(x^d|x^{\pi_{<t}}, \pi^{(k)})\right],$$
$$\pi^{(1)}, \dots, \pi^{(K)} \overset{\text{i.i.d.}}{\sim} p(\pi), \quad \pi^{(k)} \in \mathcal{G}. \qquad (13)$$

The Block Models extensions simply follows from (7).

Choice of surrogate ψ. Since the tightness of TUBE depends on the closeness of surrogate $\psi(x)$ to the $p(x)$, the choice of it is important. Here we propose two versions of its construction.

(1) Exact likelihood surrogate model. One option is to take the model which has the similar density to $p_{\text{model}}(x)$, but allows for fast and exact likelihood computation, e.g., ARM. Since p_{model} is trained on data p_{data}, than it is natural to take ARM trained on the same data ($p_{\text{ARM}} \approx p_{\text{data}} \approx p_{\text{model}}$):

$$\psi_{ARM}(x) = p_{\text{ARM}}(x) = \prod_{t=1}^{L} p_{\psi}(x^{\pi_t}|x^{\pi_{<t}}). \qquad (14)$$

Furthermore, to bring the surrogate ψ_{ARM} model closer to the p_{model} one may finetune the ψ_{ARM} using the synthetic data generated by p_{model}, which can be done be regular likelihood maximization procedure (Radford & Narasimhan, 2018) or more complex procedures (Ouyang et al., 2022).

(2) Self-surrogate. Another option, is to suppose that conditional on arbitrary order π likelihood $p_{\text{model}}(x|\pi)$ is close to full model likelihood, i.e., $p_{\text{model}}(x|\pi) \approx p_{\text{model}}(x)$, which is in fact true at the model optimum (Ou et al., 2025). Furthermore, we can take several orders $\pi^{(m)}$ and average $p_{\text{model}}(x|\pi^{(m)})$ for them, which on practice should be even closer to the $p_{\text{model}}(x)$. We consider:

$$\psi_M(x) = \frac{1}{M}\sum_{m=1}^{M} p_{\text{model}}(x|\hat{\pi}^{(m)}) \qquad (15)$$

where $\hat{\pi}^{(1)}, \dots, \hat{\pi}^{(M)}$ are some chosen orders. Note that both $\widehat{p}_{\text{model},K}(x)$ and $\psi_M(x)$ utilize orders $\pi^{(k)}$ and $\hat{\pi}^{(m)}$. If one samples the orders for ψ_M at random, e.g., from $p(\pi)$, they should be *independent* from orders for $\widehat{p}_{\text{model},K}(x)$ to avoid correlation bias in $\mathrm{TUBE}_{\psi,K}(x)$.

The computation of TUBE. The practical procedure for two-sided likelihood localization consists of sampling latent variables $\pi^{(k)}$, evaluating the target model $p_{\text{model}}(x \mid \pi^{(k)})$, computing the surrogate $\psi(x)$, and then evaluating (11). One can see the detailed algorithm in Appendix B. The computational complexity is then determined by two components: *(1)* evaluation of the likelihood model p_{model} and *(2)* evaluation of the surrogate ψ. Estimating complexity of $\widehat{p}_{\text{model},K}$ scales linearly with the number of Monte Carlo samples K, i.e., $\mathcal{O}(K)$, since it requires K evaluations of full $p_{\text{model}}(x \mid \pi^{(k)})$. The cost of ψ depends on its form: for ψ_{ARM}, it is a single forward pass through the AR model, i.e., $\mathcal{O}(1)$, while for $\psi_M(x)$ the cost is similar to $\widehat{p}_{\text{model},K}$ and linear in M, i.e., $\mathcal{O}(M)$.

4 Related work

External-evaluator metrics. A parallel line of work bypasses $\log p_{\text{model}}(x)$ and scores samples under a separate pretrained reference model. *Generative perplexity (gen-PPL)* (Lou et al., 2024; Sahoo et al., 2024) reports the perplexity of samples under GPT-2 Large. However, it can be lowered simply by reducing sample diversity (Zheng et al., 2025), motivating the de facto practice of reporting it alongside sample entropy which complicated comparisons. *MAUVE* (Pillutla et al., 2021) replaces gen-PPL's

Table 1. Estimators for various bounds on log-likelihood for latent-variable models. We say that the empirical estimator is **bound-preserving** if its *expectation* provably preserves the bound direction. Bound-breaking terms are highlighted in red; our TUBE's estimator is bound-preserving.

Bound name	Bound type	Empirical estimator	Bound-preserving?
ELBO	Lower	$\log p_{\text{model}}(x\|\pi^{(1)})$	✓
ELBO_K	Lower	$\log\left(\frac{1}{K}\sum_{k=1}^{K} p_{\text{model}}(x\|\pi^{(k)})\right)$	✓
$\text{CUBO}_{\beta\geq 1}$	Upper	$\frac{1}{\beta}\log\left(\frac{1}{K}\sum_{k=1}^{K} p_{\text{model}}(x\|\pi^{(k)})^{\beta}\right)$	✗
TVO_{Λ}^{U}	Upper	$\frac{1}{\Lambda}\sum_{\lambda=1}^{\Lambda}\sum_{k=1}^{K} \bar{w}_k^{(\beta_\lambda)} \log p_{\text{model}}(x\|\pi^{(k)})$	✗
IS-VG-B	Upper	$\frac{1}{n_p}\sum_{j=1}^{n_p}\log\left(\frac{1}{s}\sum_{i=1}^{s} p_{\text{model}}(x\|\pi^{(j,i)})\right) + \log\left(\frac{1}{n_p}\sum_{j=1}^{n_p}\frac{\sum_{i=1}^{s} p_{\text{model}}(x\|\widetilde{\pi}^{(j,i)})}{\sum_{i=1}^{s} p_{\text{model}}(x\|\pi^{(j,i)})}\right)$	✗
TUBE (Ours)	Upper	$\log\psi(x) + \frac{\frac{1}{K}\sum_{k=1}^{K} p_{\text{model}}(x\|\pi^{(k)}) - \psi(x)}{\psi(x)}$	✓

token-level scoring with an embedding-space divergence and inherits the encoder's coverage biases. (Pynadath et al., 2026) formalizes the perplexity-entropy pairing via a KL decomposition and proposes a frontier-based comparison, while (Salimans et al., 2026) introduces a gradient-based alternative to gen-PPL that vanishes when samples match the training distribution. All of these score the model's samples through a chosen reference rather than evaluating $\log p_{\text{model}}(x)$ on real data, so they cannot be compared to ARM's exact log-likelihood on the same axis.

Log-likelihood approximation for MDMs. Evaluating $\log p_{\text{model}}(x)$ for a pretrained MDM is intractable, so most works report single-sample ELBO (Sahoo et al., 2024; Haxholli et al., 2025) or its multi-sample tightening $\textbf{ELBO}_K$, also known as I-WAE (Burda et al., 2016), see Table 1. Two recent works go further: DUEL (Turok et al., 2026) computes $\log p_{\text{model}}(x)$ exactly, but only under deterministic unmasking, restricting the sampler family, while SPG (Wang et al., 2026) introduces a sandwich estimate of $\log p_{\text{model}}(x)$ between ELBO and Rényi upper bound (Li & Turner, 2016).

Upper bound estimators on log-likelihood. Here we highlight three established families of generic upper-bound estimators on $\log p_{\text{model}}(x)$ for latent variable models that are applicable to MDMs and most related to our work: $\textbf{CUBO}_\beta$, $\textbf{TVO}_L$, **IS-VG-B**, see Table 1 and Appendix C for summary.

- **CUBO_β / Rényi** (Li & Turner, 2016; Dieng et al., 2017; Wang et al., 2026) take the form $\frac{1}{\beta}\log\mathbb{E}_\pi[p_{\text{model}}(x|\pi)^\beta]$ for $\beta\geq 1$, an upper bound on $\log p_{\text{model}}(x)$; exact at $\beta=1$. Its empirical estimator (see Table 1) applies an outer log to a Monte Carlo mean, so Jensen acts *against* the upper bound at finite K, adding bias which may remove the upper bound guarantee for expectation.
- **TVO** (Masrani et al., 2019) is a right Riemann sum of the thermodynamic identity along the geometric path $q_\beta(\pi|x) \propto p_{\text{model}}(x|\pi)^\beta$, discretized at Λ grid points $\beta_\lambda = \lambda/\Lambda$, with self-normalized weights biased at finite K and an $\mathcal{O}(1/\Lambda)$ discretization residual (Table 1, Appendix C.2).
- **IS-VG-B** (Struski et al., 2023) adds an independent log-ratio correction to ELBO_s governed by two hyperparameters ($n_p \geq 2$ pairs of s-sample MC estimators, total budget $K = 2\,s\,n_p$). The bound is rigorous in the population, but the empirical estimator is biased and not bound-preserving, since both log terms apply Jensen-on-log to MC averages (see Table 1 and Appendix C.3).

In comparison to all these bounds, our TUBE admits unbiased bound-preserving estimator.

5 Experiments

We demonstrate our TUBE through two categories of experiments. First, we show that TUBE is tight enough and helps to compare AO-ARM/MDM with ARMs (§5.1). Second, we ablate the tightness of TUBE with respect to the choice of surrogate ψ (§5.2). The code is available in the supplementary.

All evaluations are done with BMs (Arriola et al., 2025) at block sizes $L' \in 4, 8, 16$ on OWT and LM1B (Gokaslan & Cohen, 2019; Chelba et al., 2014). Following standard practice for MDMs (§2.3), we report results in terms of perplexity $\text{PPL} = \exp(-\log p_{\text{model}})$ rather than log-likelihood directly. *In this form, upper bounds for log-likelihood result in lower bounds for PPL, and vice-versa, lower log-likelihood bounds produce upper bounds for PPL.* For OWT, we use the available pretrained checkpoints. For LM1B, we train the BM from scratch using the original code, since checkpoints are unavailable. The conclusions are similar on both datasets, so we report only OWT results in the main text and defer LM1B results to Appendix A.

Table 2. **Test perplexity (↓) on OWT.** We report $\mathrm{PPL} = \exp(-\log p_{\mathrm{model}}(x))$. The $\pm$ values denote standard deviations over 10 random subsets of orderings, with all estimators in a row computed using the same total number of sampled orderings. The reference column reports ELBO_K, which is exact for $L' = 4$ (marked with *) and Monte Carlo-estimated for $L' \in \{8, 16\}$. **Color coding:** coral background marks biased estimators, and red text marks cells where the value exceeds ELBO_K, violating the expected bound relation.

		Biased upper estimators for log-likelihood			Ours	References		Gap
L'	Regime	$\mathrm{CUBO}_{\beta=2}$	TVO_U	IS-VG-B	TUBE	ELBO_K	ELBO	|TUBE−ARM|
Exact ARM baseline: **17.54** PPL								
4	NFE= 1	97.78	97.78	97.78	97.78	97.78	–	80.24
	NFE= 2	24.34 ± 0.01	27.68 ± 0.01	29.11 ± 0.03	$\geq 21.55 \pm 0.03$	≤ 27.40	–	4.01
	NFE= 4	19.78 ± 0.01	21.89 ± 0.01	22.52 ± 0.01	$\geq 19.16 \pm 0.02$	≤ 21.63	–	1.62
	AO-ARM	17.50 ± 0.00	18.55 ± 0.00	18.76 ± 0.01	$\geq 17.74 \pm 0.01$	$= 18.46^\star$	$\leq 21.36 \pm 0.21$	0.20
8	NFE= 1	221.54	221.54	221.54	221.54	221.54	–	204.00
	NFE= 2	29.04 ± 0.01	33.41 ± 0.02	34.95 ± 0.03	$\geq 24.32 \pm 0.06$	≤ 32.63	–	6.78
	NFE= 4	20.90 ± 0.01	23.54 ± 0.01	24.15 ± 0.01	$\geq 20.23 \pm 0.03$	≤ 23.25	–	2.69
	NFE= 8	18.85 ± 0.01	20.83 ± 0.01	21.17 ± 0.01	$\geq 19.15 \pm 0.02$	≤ 20.67	–	1.61
	AO-ARM	17.95 ± 0.00	19.31 ± 0.00	19.46 ± 0.00	$\geq 18.67 \pm 0.01$	≤ 19.24	$\leq 22.18 \pm 0.58$	1.13
16	NFE= 1	398.63	398.63	398.63	398.63	398.63	–	381.09
	NFE= 2	35.80 ± 0.01	39.80 ± 0.01	41.93 ± 0.06	$\geq 22.50 \pm 0.15$	≤ 38.43	–	4.96
	NFE= 4	22.89 ± 0.02	25.21 ± 0.01	25.96 ± 0.02	$\geq 19.03 \pm 0.08$	≤ 24.75	–	1.49
	NFE= 8	19.59 ± 0.01	21.38 ± 0.01	21.77 ± 0.01	$\geq 18.44 \pm 0.04$	≤ 21.14	–	0.90
	NFE= 16	18.44 ± 0.01	19.98 ± 0.00	20.26 ± 0.01	$\geq 18.22 \pm 0.03$	≤ 19.83	–	0.68
	AO-ARM	18.08 ± 0.00	19.39 ± 0.00	19.56 ± 0.01	$\geq 18.47 \pm 0.01$	≤ 19.30	$\leq 22.81 \pm 0.61$	0.93

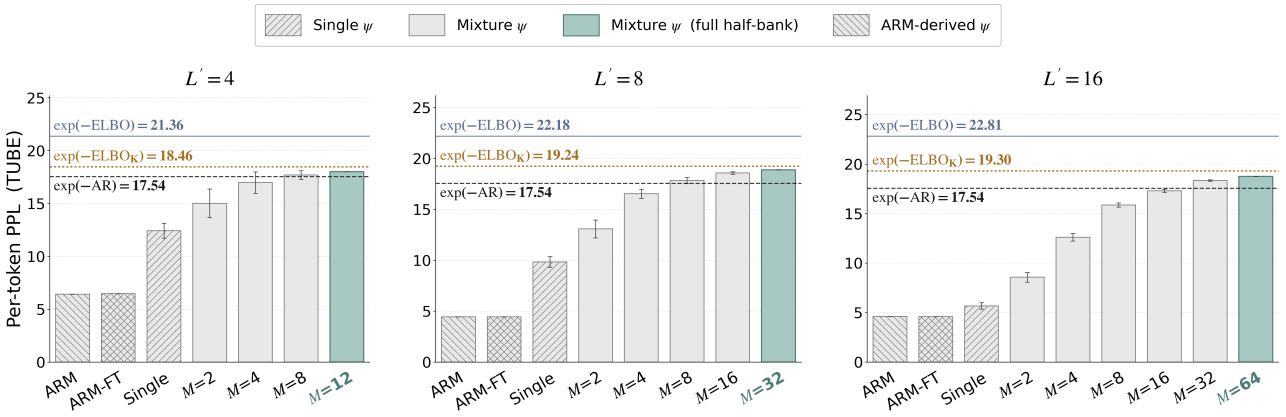

Figure 3. **Test perplexity (↓) with different ψ choices on OWT.** We report $\mathrm{PPL} = \exp(-\log p_{\mathrm{model}}(x))$. The bar plot reports standard deviations over 10 repeated ordering samples with different random seeds. The dotted gold reference line shows ELBO_K.

The experimental details provided in Appendix D.

5.1 Likelihood comparison

Here we ask two questions: is the true log-likelihood of AO-ARM/MDM competitive with an ARM baseline, and are existing lower-side estimators on log-likelihood tight enough to settle the comparison? We address both by pairing ELBO_K with TUBE and benchmarking against prior estimators on a shared set of Monte Carlo samples.

Setup. We compare TUBE against $\mathrm{CUBO}_{\beta=2}$, TVO_U, IS-VG-B, and also report ELBO and ELBO_K. The same pretrained BM is evaluated under two regimes: AO-ARM ($\pi \in \mathcal{S}$) and MDM ($\pi \in \mathcal{G}$, NFE $\in \{1, 2, 4, \dots\}$). For $L' = 4$ in AO-ARM we enumerate all $4! = 24$ orderings, which gives the **exact** per-block log-likelihood and we report it instead of ELBO_K. For $L' \in \{8, 16\}$ this is intractable, so we sample 64 and 128 orderings respectively. Then, split equally into two independent sets used for $\widehat{p}_{\mathrm{model},K}$ and ψ_M, with ELBO_K reported as a Monte Carlo estimate.

Results. For MDMs, Table 2 shows that TUBE is generally the tightest lower bound on PPL across all NFEs ($= T$ in §2.3), compared with the competing estimators, with $\mathrm{CUBO}_{\beta=2}$ being the only exception. However, we emphasize that $\mathrm{CUBO}_{\beta=2}$ is not reliable, as discussed in the next paragraph. When comparing MDMs with the ARM baseline, the gap in the rightmost column, decreases from about 5 to 1 PPL as NFE increases. For AO-ARMs, TUBE is the tightest estimator across the reported settings. The conclusions are the clearest for $L' = 4$, where ELBO_K is exact, so the likelihood gap between AO-ARMs/MDMs and ARMs is localized without Monte Carlo ambiguity.

> **Takeaway.** In every configuration the AO-ARM/MDM PPL lies strictly above the ARM, so ARM is dominant in likelihood across all block sizes and generation regimes we consider.

The other estimators behave differently. TVO_U and IS-VG-B sit *above* ELBO_K in every regime with $\mathrm{NFE} \geq 2$, **so they fail to give a reliable bound**. This is structural: each applies a nonlinear function after Monte Carlo averaging, so the population bound does not transfer to its empirical estimator. TUBE avoids this by being affine in $\widehat{p}_{\mathrm{model},K}$, so the bound from (9) transfers to the Monte Carlo estimator at every $K \geq 1$. $\mathrm{CUBO}_{\beta=2}$ behaves differently from TVO_U and IS-VG-B: in our data it sits below ELBO_K in every row. Nevertheless, this does not make $\mathrm{CUBO}_{\beta=2}$ a valid upper bound: as discussed in §4, its empirical estimator applies log to a Monte Carlo mean and is therefore biased downward. Furthermore, the choice of β can substantially change the estimator bias and even lead to values exceeding $\overline{\mathrm{ELBO}}_K$, see Appendix C.1. This makes CUBO **unreliable** as well.

5.2 Surrogate ψ choice and finite-sample behavior

By (9), the tightness of TUBE depends on two factors: the number of orderings used to estimate $\widehat{p}_{\mathrm{model},K}$, and the surrogate ψ used to construct the bound. While the Monte Carlo error in estimating p_{model} decreases as $1/\sqrt{K}$, the role of ψ is less straightforward. So we ask which construction from §3.2 closes gap between TUBE_ψ and $\log p_{\mathrm{model}}(x)$ most effectively.

Setup. With $\widehat{p}_{\mathrm{model},K}$ fixed, we compare four self-surrogates: a single-order surrogate ψ_π, the self-surrogate ψ_M, an external ARM surrogate ψ_{ARM}, and a variant of ARM fine-tuned on samples generated by the BM (Appendix D). We vary the number of sampled orderings M on a $\log_2$ grid up to the full ordering budget used in §5.1.

Results. Figure 3 shows that the self-surrogate ψ_M gets closest to ELBO_K at every block size with increase of M. The external ARM surrogate ψ_{ARM} leaves a much wider gap, and fine-tuning it on BM samples shifts the bound only marginally. A single fixed order ψ_π is looser still. This matches the discussion of §3.2: ψ_M averages $p_{\mathrm{model}}(x|\pi)$ over independent orderings drawn from the same model and therefore tracks $p_{\mathrm{model}}(x)$ pointwise better than an external density.

6 Potential impact and limitations

Limitations. Our empirical study is limited to BM (Arriola et al., 2025) which is the dominant inference-time paradigm of large-scale MDMs such as LLaDA 2.0 (Bie et al., 2025) and Mercury (Khanna et al., 2025). However, applying TUBE to non-block MDM may be challenging. TUBE's tightness requires the surrogate $\psi(x) \approx p_{\mathrm{model}}(x)$. For non-block MDM at typical sequence lengths $L \sim 10^2$ to 10^3, $p_{\mathrm{model}}(x)$ is approximately product of L per-token probabilities, vanishingly small in absolute scale. Thus, ψ becomes hard to estimate reliably in practice. In turn, BM decomposition (7) for moderate block size keeps computing log-likelihood numerically feasible.

Potential impact. The reliable two-sided estimation of $\log p_{\mathrm{model}}(x)$ enables several downstream uses. It tightens perplexity bounds for diffusion language models (Sahoo et al., 2024; Arriola et al., 2025), supplies finite-sample policy ratios for RL post-training (Christiano et al., 2017; Ouyang et al., 2022; Shao et al., 2024; Rafailov et al., 2023), and supports likelihood-based evaluation in scientific generative modeling such as protein sequence design (Alamdari et al., 2023; Wang et al., 2025a). Studying these directions using our TUBE is a promising avenue for the future work.

References

Alamdari, S., Thakkar, N., van den Berg, R., Lu, A. X., Fusi, N., Amini, A. P., and Yang, K. K. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, 2023. doi: 10.1101/2023.09.11.556673.

Arriola, M., Gokaslan, A., Chiu, J. T., Yang, Z., Qi, Z., Han, J., Sahoo, S. S., and Kuleshov, V. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Bie, T., Cao, M., Chen, K., Du, L., Gong, M., Gong, Z., Gu, Y., Hu, J., Huang, Z., Lan, Z., Li, C., Li, C., Li, J., Li, Z., Liu, H., Liu, L., Lu, G., Lu, X., Ma, Y., Tan, J.,

Wei, L., Wen, J.-R., Xing, Y., Zhang, X., Zhao, J., Zheng, D., Zhou, J., Zhou, J., Zhou, Z., Zhu, L., and Zhuang, Y. LLaDA2.0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*, 2025.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. In *International Conference on Learning Representations (ICLR)*, 2016.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., and Robinson, T. One billion word benchmark for measuring progress in statistical language modeling, 2014. URL https://arxiv.org/abs/1312.3005.

Chen, M., Radford, A., Wu, J., Jun, H., Dhariwal, P., Luan, D., and Sutskever, I. Generative pretraining from pixels. In *International Conference on Machine Learning*, 2020. URL https://api.semanticscholar.org/CorpusID:219781060.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Dieng, A. B., Tran, D., Ranganath, R., Paisley, J., and Blei, D. M. Variational inference via χ-upper bound minimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Gokaslan, A. and Cohen, V. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., and Guo, B. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10696–10706, 2022.

Haxholli, E., Gurbuz, Y. Z., Can, O., and Waxman, E. Efficient perplexity bound and ratio matching in discrete diffusion language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 30016–30030, 2022.

Hoogeboom, E., Gritsenko, A. A., Bastings, J., Poole, B., van den Berg, R., and Salimans, T. Autoregressive diffusion models. In *The Tenth International Conference on Learning Representations (ICLR)*, 2022.

Jeon, M., Shin, S., Jeon, D., and No, A. Information-theoretic discrete diffusion. *arXiv preprint arXiv:2510.24088*, 2025.

Ji, C. and Shen, H. Stochastic variational inference via upper bound. In *NeurIPS Workshop on Bayesian Deep Learning*, 2019. arXiv:1912.00650 – KL-based EUBO; included for completeness, distinct from SPG's Rényi-based bound.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Karimi Monsefi, A., Bhendawade, N., Ciosici, M. R., Culver, D., Zhang, Y., and Belousova, I. FS-DFM: Fast and accurate long text generation with few-step diffusion language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Kelvinius, F. E. and Lindsten, F. Discriminator guidance for autoregressive diffusion models. In *International Conference on Artificial Intelligence and Statistics*, pp. 3403–3411. PMLR, 2024.

Khanna, S., Kharbanda, S., Li, S., Varma, H., Wang, E., Birnbaum, S., Luo, Z., Miraoui, Y., Palrecha, A., Ermon, S., Grover, A., and Kuleshov, V. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Lee, S., Kreis, K., Veccham, S. P., Liu, M., Reidenbach, D., Peng, Y., Paliwal, S. G., Nie, W., and Vahdat, A. Genmol: A drug discovery generalist with discrete diffusion. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=KM7pXWG1xj.

Li, X., Trabucco, B., Park, D. H., Luo, M., Shen, S., Darrell, T., and Gao, Y. Discovering non-monotonic autoregressive orderings with variational inference. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jP1vTH3inC.

Li, Y. and Turner, R. E. Rényi divergence variational inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Lou, A., Meng, C., and Ermon, S. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Masrani, V., Le, T. A., and Wood, F. The thermodynamic variational objective. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.-R., and Li, C. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Ou, J., Nie, S., Xue, K., Zhu, F., Sun, J., Li, Z., and Li, C. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Owen, A. B. *Monte Carlo theory, methods and examples*. Stanford University, 2013.

Pang, Z., Zhang, T., Luan, F., Man, Y., Tan, H., Zhang, K., Freeman, W. T., and Wang, Y.-X. Randar: Decoder-only autoregressive visual generation in random orders. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 45–55, 2025.

Pillutla, K., Swayamdipta, S., Zellers, R., Thickstun, J., Welleck, S., Choi, Y., and Harchaoui, Z. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems*, 2021.

Pynadath, P., Shi, J., and Zhang, R. Generative frontiers: Why evaluation matters for diffusion language models. *arXiv preprint arXiv:2604.02718*, 2026.

Radford, A. and Narasimhan, K. Improving language understanding by generative pre-training. 2018. URL https://api.semanticscholar.org/CorpusID:49313245.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Sahoo, S. S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J. T., Rush, A. M., and Kuleshov, V. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Sahoo, S. S., Yang, Z., Akhauri, Y., Liu, J., Singh, D., Cheng, Z., Liu, Z., Xing, E., Thickstun, J., and Vahdat, A. Esoteric language models. *arXiv preprint arXiv:2506.01928*, 2025.

Salimans, T., Mensink, T., Heek, J., and Hoogeboom, E. Beyond single tokens: Distilling discrete diffusion models via discrete MMD. *arXiv preprint arXiv:2603.20155*, 2026.

Seo, H., Kim, T., Yu, S., and Ahn, S. Learning flexible forward trajectories for masked molecular diffusion. *arXiv preprint arXiv:2505.16790*, 2025.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Shi, J., Han, K., Wang, Z., Doucet, A., and Titsias, M. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167, 2024.

Shih, A., Sadigh, D., and Ermon, S. Training and inference on any-order autoregressive models the right way. *Advances in Neural Information Processing Systems*, 35: 2762–2775, 2022.

Struski, Ł., Mazur, M., Batorski, P., Spurek, P., and Tabor, J. Bounding evidence and estimating log-likelihood in VAE. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 675–693, 2023.

Turok, G., De Sa, C., and Kuleshov, V. DUEL: Exact likelihood for masked diffusion via deterministic unmasking. *arXiv preprint arXiv:2603.01367*, 2026.

Uria, B., Murray, I., and Larochelle, H. A deep and tractable density estimator. In *International Conference on Machine Learning*, pp. 467–475. PMLR, 2014.

Wang, C., Rashidinejad, P., Su, D., Jiang, S., Wang, S., Zhao, S., Zhou, C., Shen, S. Z., Chen, F., Jaakkola, T., Tian, Y., and Liu, B. SPG: Sandwiched policy gradient for masked diffusion language models. In *The Fourteenth International Conference on Learning Representations*, 2026. Poster.

Wang, X., Zheng, Z., Ye, F., Xue, D., Huang, S., and Gu, Q. DPLM-2: A multimodal diffusion protein language

model. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025a.

Wang, Z., Shi, J., Heess, N., Gretton, A., and Titsias, M. Learning-order autoregressive models with application to molecular graph generation. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=EY6pXIDi3G.

Ye, J., Xie, Z., Zheng, L., Gao, J., Wu, Z., Jiang, X., Li, Z., and Kong, L. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Zheng, K., Chen, Y., Mao, H., Liu, M.-Y., Zhu, J., and Zhang, Q. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

A Experiments on LM1B

This section reports the results of our TUBE evaluation on LM1B dataset (Table 3 and Figure 4). The conclusions are analogous to those from the main paper obtained on OWT dataset.

B TUBE Estimation Algorithm

Here we present the Algorithm 1 for the two-sided likelihood localization via our TUBE_ψ (9) and ELBO_K estimate (Sahoo et al., 2025) which originates from I-WAE model (Burda et al., 2016).

Algorithm 1 TUBE_ψ and ELBO_K: two-sided likelihood interval

Require: sample x, surrogate type $c \in \{\mathrm{ARM}, \mathrm{self}\}$, sample budgets $K, M \geq 1$, model $p_{\mathrm{model}}(\cdot|\pi)$, latent prior $p(\pi)$, optional ARM surrogate p_{ARM}

1: Sample $\pi^{(1)}, \ldots, \pi^{(K)} \overset{\text{i.i.d.}}{\sim} p(\pi)$

2: $\widehat{p}_{\mathrm{model},K}(x) \leftarrow \frac{1}{K}\sum_{k=1}^{K} p_{\mathrm{model}}(x|\pi^{(k)})$

3: $\widehat{\mathrm{TUBE}}_{\psi,K}(x) \leftarrow \log\psi(x) + \frac{\widehat{p}_{\mathrm{model},K}(x) - \psi(x)}{\psi(x)}$

4: $\widehat{\mathrm{ELBO}}_K(x) \leftarrow \log\widehat{p}_{\mathrm{model},K}(x)$

5: **return** LL interval $[\,\widehat{\mathrm{ELBO}}_K(x),\ \widehat{\mathrm{TUBE}}_{\psi,K}(x)\,]$

C Comparison of likelihood estimators

In this appendix we give the precise definitions of the upper-bound estimators outlined in §4 (CUBO_β, TVO_Λ, IS-VG-B), derive their finite-sample Monte Carlo forms, and identify the structural source of bias in each. Throughout, we use the notation of §3: π is the latent generation structure (an order $\pi \in \mathcal{S}$ for AO-ARM, a grouped ordering $\pi \in \mathcal{G}$ for MDM), $p(\pi)$ is a distribution on orders defined by the model, and the standard K-sample Monte Carlo estimator $\widehat{p}_{\mathrm{model},K}(x)$ is the one defined in (11). Each estimator below uses this single shared sample bank.

C.1 CUBO and the Rényi variational bound

The χ^β-divergence upper bound (CUBO) of (Dieng et al., 2017) and the Rényi variational bound of (Li & Turner, 2016) are two equivalent formulations of the same family of upper bounds on $\log p_{\mathrm{model}}(x)$. In our latent-mixture form,

$$\begin{aligned}\mathrm{CUBO}_\beta(x) &= \frac{1}{\beta}\log\mathbb{E}_{\pi\sim p(\pi)}\left[p_{\mathrm{model}}(x|\pi)^\beta\right] \\ &\geq \log p_{\mathrm{model}}(x), \quad \beta \geq 1.\end{aligned} \tag{16}$$

with equality at $\beta = 1$ and monotone non-decreasing in β (Dieng et al., 2017).

Empirical Monte Carlo estimator. It can be practically estimated (Table 1) as

$$\widehat{\mathrm{CUBO}}_{\beta,K}(x) = \tfrac{1}{\beta}\log\Big(\tfrac{1}{K}\textstyle\sum_{k=1}^{K} p_{\mathrm{model}}(x|\pi^{(k)})^\beta\Big) \tag{17}$$

Source of bias. The estimator (17) is biased at finite K: by Jensen on the outer log, $\mathbb{E}[\widehat{\mathrm{CUBO}}_{\beta,K}] \leq \mathrm{CUBO}_\beta$. The population guarantee $\log p_{\mathrm{model}}(x) \leq \mathrm{CUBO}_\beta$ therefore does not transfer to the empirical estimator: the chain $\log p_{\mathrm{model}}(x) \leq \mathbb{E}[\widehat{\mathrm{CUBO}}_{\beta,K}]$ is no longer ensured, so the empirical expectation may fall below $\log p_{\mathrm{model}}(x)$ and void the upper bound. Figure 5 shows configurations of $(|\mathcal{B}|, \beta)$ where this occurs in practice. SPG (Wang et al., 2026) observes this and notes that the linearization $\log x \leq x - 1$ would restore the bound at the cost of looseness, motivating their tighter-but-biased choice.

Connection to SPG. SPG's "evidence upper bound" (EUBO) (Wang et al., 2026) is CUBO_β specialized to the MDLM absorbing forward process (Sahoo et al., 2024): SPG's Lemma 1 is (16) under $n \leftrightarrow \beta$, and Theorem 1 specializes it to the per-token MDLM kernel. This is distinct from the KL-based EUBO of (Ji & Shen, 2019), which requires posterior samples.

C.2 TVO: Thermodynamic-integration Upper Bound

The TVO of (Masrani et al., 2019) starts from the thermodynamic identity

$$\begin{aligned}\log p_{\mathrm{model}}(x) &= \int_0^1 \mathbb{E}_{\pi\sim q_\beta(\cdot|x)}[\log p_{\mathrm{model}}(x|\pi)]\ d\beta, \\ q_\beta(\pi|x) &\propto p_{\mathrm{model}}(x|\pi)^\beta,\end{aligned} \tag{18}$$

where q_β interpolates between the prior $q_0 = p(\pi)$ and the posterior $q_1 = p_{\mathrm{model}}(\pi|x)$. The integrand is monotone in β (Masrani et al., 2019), so the right Riemann sum on the equispaced grid $\beta_\lambda = \lambda/\Lambda$ upper-bounds the integral:

$$\begin{aligned}\log p_{\mathrm{model}}(x) &\leq \frac{1}{\Lambda}\sum_{\lambda=1}^{\Lambda}\mathbb{E}_{\pi\sim q_{\beta_\lambda}(\cdot|x)}[\log p_{\mathrm{model}}(x|\pi)] \\ &=: \mathrm{TVO}_\Lambda^U(x).\end{aligned} \tag{19}$$

Empirical Monte Carlo estimator. Using the same K sampled orderings as in (11) with self-normalized weights $\bar{w}_k^{(\beta)} = p_{\mathrm{model}}(x|\pi^{(k)})^\beta \,/\, \sum_j p_{\mathrm{model}}(x|\pi^{(j)})^\beta$, the empirical MC estimator (Table 1) is

$$\widehat{\mathrm{TVO}}^U_{\Lambda,K}(x) = \frac{1}{\Lambda}\sum_{\lambda=1}^{\Lambda}\sum_{k=1}^{K}\bar{w}_k^{(\beta_\lambda)}\ \log p_{\mathrm{model}}(x|\pi^{(k)}). \tag{20}$$

Table 3. **Test perplexity (↓) on LM1B.** We report $\mathrm{PPL} = \exp(-\log p_{\mathrm{model}}(x))$. The ± values denote standard deviations over 10 random subsets of orderings, with all estimators in a row computed using the same total number of sampled orderings. The reference column reports ELBO_K, which is exact for $L' = 4$ (marked with *) and Monte Carlo-estimated for $L' \in \{8, 16\}$. **Color coding:** coral background marks biased estimators, and red text marks cells where the value exceeds ELBO_K, violating the expected bound relation.

		Biased upper estimators			Ours	References		Gap
L'	Regime	$\mathrm{CUBO}_{\beta=2}$	TVO_U	IS-VG-B	TUBE	ELBO_K	ELBO	TUBE−ARM
Exact ARM baseline: 22.46 PPL								
4	NFE= 1	137.09	137.09	137.09	137.09	137.09	–	114.62
	NFE= 2	34.67 ± 0.01	39.45 ± 0.01	41.40 ± 0.04	$\geq 30.87 \pm 0.03$	≤ 39.08	–	8.41
	NFE= 4	28.05 ± 0.01	31.01 ± 0.01	31.85 ± 0.02	$\geq 27.34 \pm 0.03$	≤ 30.67	–	4.88
	AO-ARM	25.25 ± 0.00	26.54 ± 0.00	26.78 ± 0.00	$\geq 25.63 \pm 0.01$	$= 26.45^\star$	$\leq 28.95 \pm 0.38$	3.17
8	NFE= 1	299.88	299.88	299.88	299.88	299.88	–	277.42
	NFE= 2	41.45 ± 0.02	47.68 ± 0.02	49.73 ± 0.03	$\geq 35.24 \pm 0.05$	≤ 46.66	–	12.77
	NFE= 4	29.78 ± 0.01	33.53 ± 0.01	34.34 ± 0.02	$\geq 29.11 \pm 0.04$	≤ 33.14	–	6.65
	NFE= 8	26.88 ± 0.01	29.65 ± 0.01	30.09 ± 0.01	$\geq 27.51 \pm 0.02$	≤ 29.44	–	5.05
	AO-ARM	25.14 ± 0.00	26.90 ± 0.00	27.09 ± 0.01	$\geq 26.15 \pm 0.01$	≤ 26.82	$\leq 30.32 \pm 0.38$	3.69
16	NFE= 1	546.32	546.32	546.32	546.32	546.32	–	523.86
	NFE= 2	51.26 ± 0.03	57.02 ± 0.03	59.97 ± 0.03	$\geq 31.99 \pm 0.13$	≤ 55.13	–	9.53
	NFE= 4	32.81 ± 0.02	36.15 ± 0.02	37.18 ± 0.02	$\geq 27.39 \pm 0.06$	≤ 35.52	–	4.93
	NFE= 8	28.00 ± 0.01	30.54 ± 0.01	31.07 ± 0.01	$\geq 26.60 \pm 0.04$	≤ 30.23	–	4.14
	NFE= 16	26.34 ± 0.01	28.50 ± 0.01	28.86 ± 0.01	$\geq 26.24 \pm 0.02$	≤ 28.30	–	3.78
	AO-ARM	25.14 ± 0.00	26.88 ± 0.00	27.09 ± 0.01	$\geq 25.82 \pm 0.02$	≤ 26.78	$\leq 31.03 \pm 0.49$	3.36

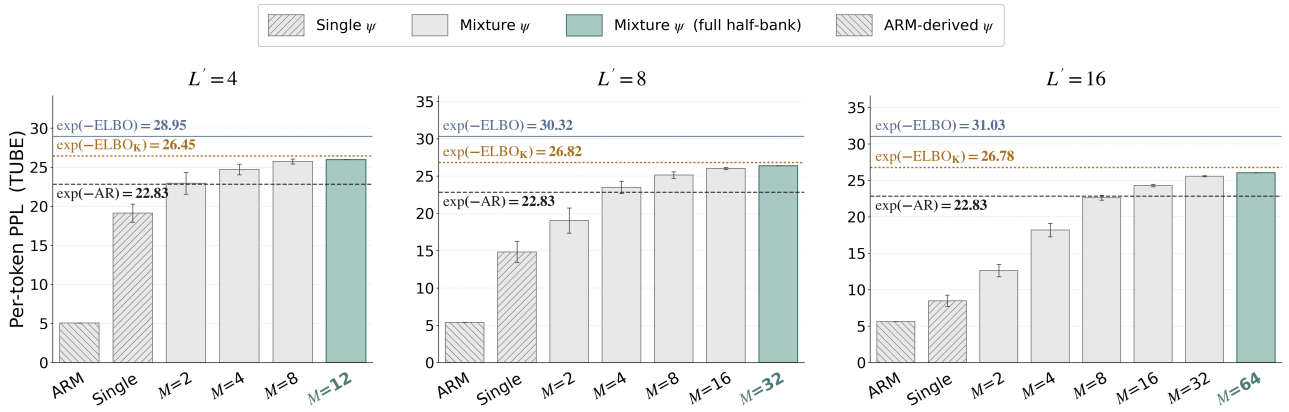

Figure 4. **Test perplexity (↓) with different ψ choices on LM1B.** We report $\mathrm{PPL} = \exp(-\log p_{\mathrm{model}}(x))$. The bar plot reports standard deviations over 10 repeated ordering samples with different random seeds. The dotted gold reference line shows ELBO_K.

Sources of bias. Two structural sources. *(i)* Self-normalized importance weights $\bar{w}_k^{(\beta)}$ have $\mathcal{O}(1/K)$ bias (Owen, 2013): as $\beta \to 1$ the target q_β concentrates on a vanishing subset of orderings while the proposal stays uniform, so weights at large β collapse onto a few outlier samples. As a result, $\mathbb{E}[\widehat{\mathrm{TVO}}^{U}_{\Lambda,K}] \neq \mathrm{TVO}^{U}_{\Lambda}$ at finite K, and the population upper-bound guarantee no longer ensures $\mathbb{E}[\widehat{\mathrm{TVO}}^{U}_{\Lambda,K}] \geq \log p_{\mathrm{model}}(x)$. *(ii)* The Riemann discretization adds an $\mathcal{O}(1/\Lambda)$ residual that does not vanish at fixed Λ.

C.3 IS-VG-B: Importance Sampling Variational Gap Bound

IS-VG-B (Struski et al., 2023) constructs an upper bound on $\log p_{\mathrm{model}}(x)$ by adding a correction term to the (lower) ELBO_s bound. The strategy is to pair a two-point inequality with a tunable parameter C, optimize C in closed form, and estimate the optimum using a second independent MC estimator drawn from the same distribution.

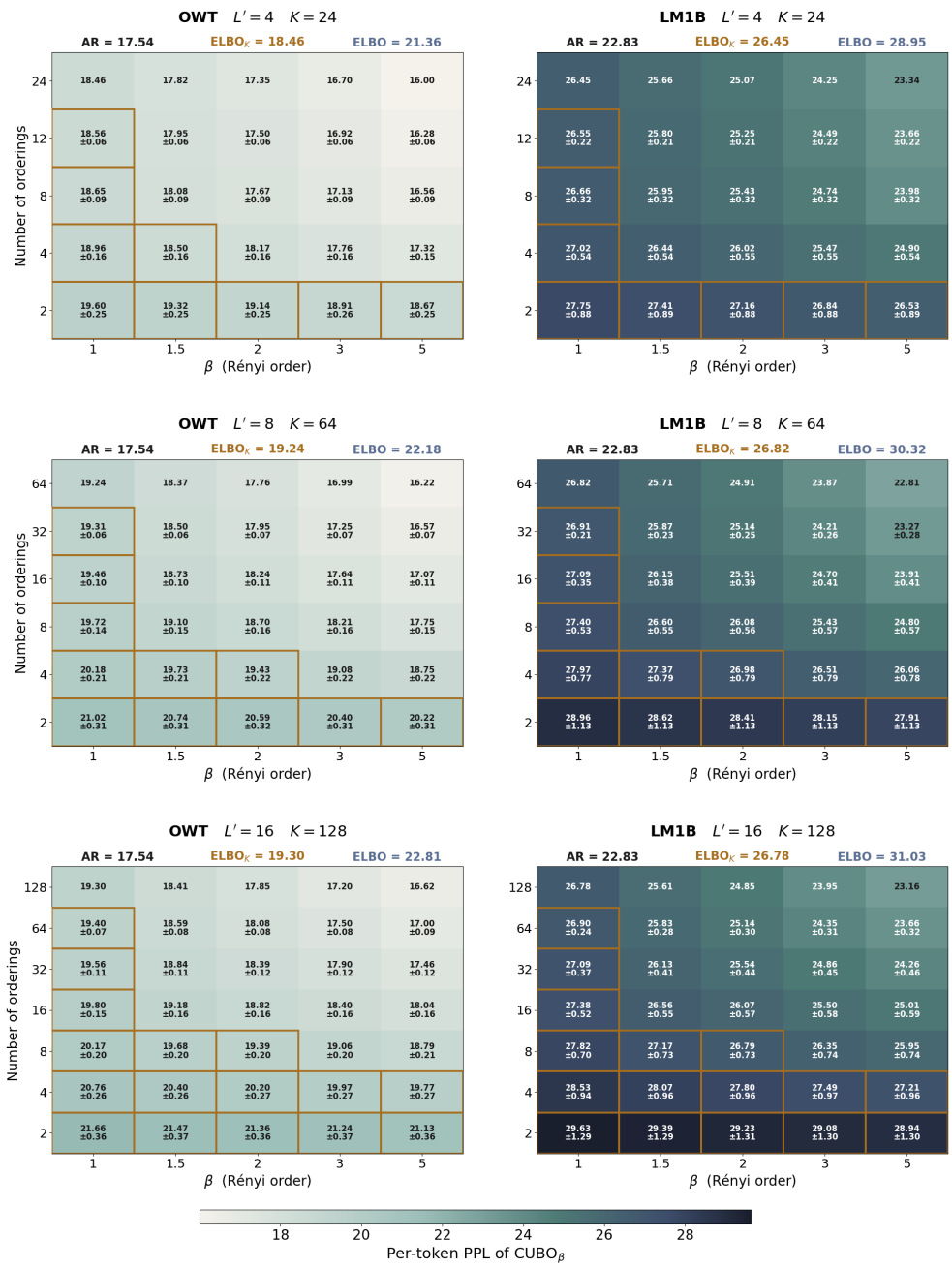

Figure 5. **CUBO$_\beta$ diagnostic on $L' \in \{4, 8, 16\}$.** PPL of CUBO$_\beta$ as a function of the number of orderings and the Rényi exponent β, on OWT (left) and LM1B (right). Gold borders mark cells exceeding $\mathrm{ELBO}_K^{\mathrm{full}}$, where CUBO violates its nominal upper-bound guarantee. The only cell in the plane that is both valid and tight is (number of orderings = 24, $\beta = 1$), which coincides with $\mathrm{ELBO}_K^{\mathrm{full}}$ by construction; reducing number of orderings breaks the bound upward, increasing β pushes it past the truth downward.

The starting two-point inequality is

$$\log \mathbb{E}X \;\leq\; \mathbb{E}\log X - 1 + C + e^{-C}\,\mathbb{E}[Y/X],$$

which holds for any $C \in \mathbb{R}$ which holds for any $C \in \mathbb{R}$ when X and Y are i.i.d. positive random variables, with optimum at $C^\star = \log \mathbb{E}[Y/X]$ (Struski et al., 2023, Theorem 4 and Corollary 4). Substituting $C^\star$ collapses the right-hand side to $\mathbb{E}\log X + \log \mathbb{E}[Y/X]$. Replacing X by the s-sample MC average $\bar{X}_s = \frac{1}{s}\sum_i X_i$ preserves the bound at each s since $\mathbb{E}\bar{X}_s = \mathbb{E}X$, and $\log \mathbb{E}[\bar{Y}_s/\bar{X}_s] \to 0$ as $s \to \infty$ on bounded support (Struski et al., 2023, Theorem 5), so the bound becomes asymptotically tight. Specializing to our

setting with $X \equiv p_{\text{model}}(x|\pi)$ and uniform $p(\pi)$ gives

$$\begin{aligned}\text{IS-VG-B}_s(x) &:= \underbrace{\mathbb{E}[\log \widehat{p}_{\text{model},s}(x)]}_{=\,\text{ELBO}_s} \\ &\quad + \underbrace{\log \mathbb{E}\left[\frac{\widetilde{p}_{\text{model},s}(x)}{\widehat{p}_{\text{model},s}(x)}\right]}_{\text{correction term}} \\ &\geq \log p_{\text{model}}(x),\end{aligned} \tag{21}$$

where $\widehat{p}_{\text{model},s}$ and $\widetilde{p}_{\text{model},s}$ are independent s-sample MC averages of $p_{\text{model}}(x|\pi)$ as in (11). The first term is the ELBO_s *lower* bound at sample size s, the correction is what flips the inequality into an *upper* bound.

Empirical Monte Carlo estimator. The correction's outer expectation is itself estimated by Monte Carlo over n_p independent pairs of s-sample MC estimators (total budget $K = 2\,s\,n_p$). Expanding the per-pair averages $\widehat{p}^{(j)}_{\text{model},s}(x) = \frac{1}{s}\sum_{i=1}^{s} p_{\text{model}}(x|\pi^{(j,i)})$ and $\widetilde{p}^{(j)}_{\text{model},s}(x) = \frac{1}{s}\sum_{i=1}^{s} p_{\text{model}}(x|\widetilde{\pi}^{(j,i)})$ explicitly recovers the form shown in Table 1:

$$\begin{aligned}\widehat{\text{IS-VG-B}}_{s,n_p}(x) &= \frac{1}{n_p}\sum_{j=1}^{n_p} \log\left(\frac{1}{s}\sum_{i=1}^{s} p_{\text{model}}(x|\pi^{(j,i)})\right) \\ &\quad + \log\left(\frac{1}{n_p}\sum_{j=1}^{n_p} \frac{\sum_{i=1}^{s} p_{\text{model}}(x|\widetilde{\pi}^{(j,i)})}{\sum_{i=1}^{s} p_{\text{model}}(x|\pi^{(j,i)})}\right),\end{aligned} \tag{22}$$

where $\pi^{(j,i)}$ and $\widetilde{\pi}^{(j,i)}$ are the i-th i.i.d. samples of the j-th X-side and Y-side MC estimators, respectively, both drawn from $p(\pi)$. The first term averages n_p ELBO_s samples, the second averages the ratios *inside* the log to estimate the population correction.

Source of bias. The first term in (22) is an MC estimate of ELBO_s, which by Jensen lies below $\log p_{\text{model}}(x)$ at any finite s, this gap is part of the construction, since the correction is designed precisely to lift the bound back above $\log p_{\text{model}}(x)$. The second term is meant to do that lifting, but is itself a log of a Monte Carlo mean: letting $r^{(j)} = \sum_{i=1}^{s} p_{\text{model}}(x|\widetilde{\pi}^{(j,i)}) \,/\, \sum_{i=1}^{s} p_{\text{model}}(x|\pi^{(j,i)})$ denote the per-pair ratio, Jensen gives $\mathbb{E}\left[\log \frac{1}{n_p}\sum_j r^{(j)}\right] \leq \log \mathbb{E}[r]$, so the empirical correction underestimates the population value and the empirical IS-VG-B can fall below $\log p_{\text{model}}(x)$ at finite K. (Struski et al., 2023) acknowledge this in their Limitations: the empirical estimator "results in nonrigorous bounds."

D Experimental details

This appendix provides the experimental details for the results reported in the main paper: models and datasets (§D.1), construction of the orders (§D.2), estimator hyperparameters used in the main tables (§D.3), and compute resources (§D.4).

D.1 Models and datasets

We evaluate the public BMs from (Arriola et al., 2025) (`https://github.com/kuleshov-group/bd3lms`). For OWT, we use publicly available checkpoints.

LM1B training details. For LM1B, we follow the original BD3-LM training procedure. We first train an MDLM with the BD3-LM repository's default hyperparameters (global batch size 512, learning rate 3×10^{-4}, 150,000 steps, sentence-wrap data preprocessing), and then fine-tune it separately for each block size $L' \in \{4, 8, 16\}$ to obtain the BD3-LM checkpoints. The ARM baseline is trained independently from scratch with the same architecture, tokenizer, and sequence length. The synthetic-finetuned ARM $\psi_{\text{ARM-FT}}$ used in Figure 3 is a one-off OWT experiment and is not available for LM1B, so the corresponding bar is missing from the LM1B panel of Figure 4.

D.2 Sampling orderings

For each block size L', we precompute a fixed set of M latent orderings per block, with each ordering specifying the unmasking sequence over the L' block positions. The bank is constructed once per $(L', \text{dataset})$ pair and shared across all estimators in the table. Number of latent orderings and construction protocol are summarized in Table 4.

Table 4. **Latent orderings construction per block size.**

L'	Construction	Number of orderings
4	full enumeration of 4! orederings	24
8	random sampling	64
16	random sampling	128

To cover the MDM regime, the same set of orderings is evaluated under the multi-step parallel-generation schedule with $\text{NFE} \in \{1, 2, 4, \ldots, L'\}$, where NFE is the number of forward passes through the MDM per block. Smaller NFE corresponds to coarser parallel denoising. The validation-set size used per evaluation is 100 batches with the eval batch size of the BD3-LM repo, giving roughly 4×10^5 per-block scores per ordering at $L' = 4$.

D.3 Estimator hyperparameters

All estimators in Tables 2–3 share a common per-row number of orderings depending on L'. This number is split between the two halves used by TUBE, and the corresponding hyperparameters of CUBO, TVO_U, and IS-VG-B are chosen so that all estimators draw orderings per row. Table 5 summarizes the configuration.

Table 5. **Estimator hyperparameters used in the main tables.** The per-row number of orderings is fixed per block size so that estimators are compute-comparable across rows. ELBO_K uses all available orderings deterministically.

Estimator	$L' = 4$	$L' = 8$	$L' = 16$
TUBE (split $B/2 + B/2$)	$12 + 12$	$32 + 32$	$64 + 64$
$\mathrm{CUBO}_{\beta=2}$ (single MC set of size B)	24	64	128
TVO_U (Riemann partitions Λ, $K = B$)	$\Lambda = 200$	$\Lambda = 200$	$\Lambda = 200$
IS-VG-B ($n_{\mathrm{p}} = 2$, $k = B/4$)	2×6	2×16	2×32
ELBO_K (full bank, deterministic)	$M = 24$	$M = 64$	$M = 128$
ELBO (15 random-seed runs)	15 seeds	15 seeds	15 seeds
MC re-seeds for $\pm$ std	10	10	10

Table 6. **Wall-clock per evaluation on a single H200 140 GB** (OWT/LM1B, 100 batches, default eval batch size).

Evaluation	$L' = 4$	$L' = 8$	$L' = 16$
ARM baseline (per-block AR)	12 m	12 m	12 m
ARM-FT (synth-finetuned, OWT only)	14 m	14 m	14 m
ELBO (15 seeds)	28 m	32 m	38 m
Ordering bank ($M \in \{24, 64, 128\}$ orderings)	35 m	1 h 25 m	2 h 50 m
Multi-step bank (full NFE schedule)	42 m	1 h 40 m	3 h 30 m

The heatmap in Appendix C sweeps wider hyperparameter ranges $\beta \in \{1.0, 1.5, 2.0, 3.0, 5.0\}$ for CUBO. The $\pm$ standard deviations in the tables are computed across 10 MC re-seeds, each drawing a fresh per-row subset of size B from the bank.

D.4 Compute

All evaluations run on a single NVIDIA H200 140 GB GPU under PyTorch 2.5.1 / CUDA 12.1. Wall-clock times for the OWT validation split with 100 batches are summarized in Table 6. LM1B times are within 30% of OWT. Total evaluation across both datasets and all three block sizes is roughly 50–60 GPU-hours.