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

Arseny Ivanov[1,2], **Sergei Kholkin**[2], **Vladislav Gromadskii**[2],
Grigoriy Ksenofontov[2,3], **Ivan Oseledets**[1,2], **Alexander Korotin**[2,1]

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. Code available at: https://github.com/arsenyivanov/tube.

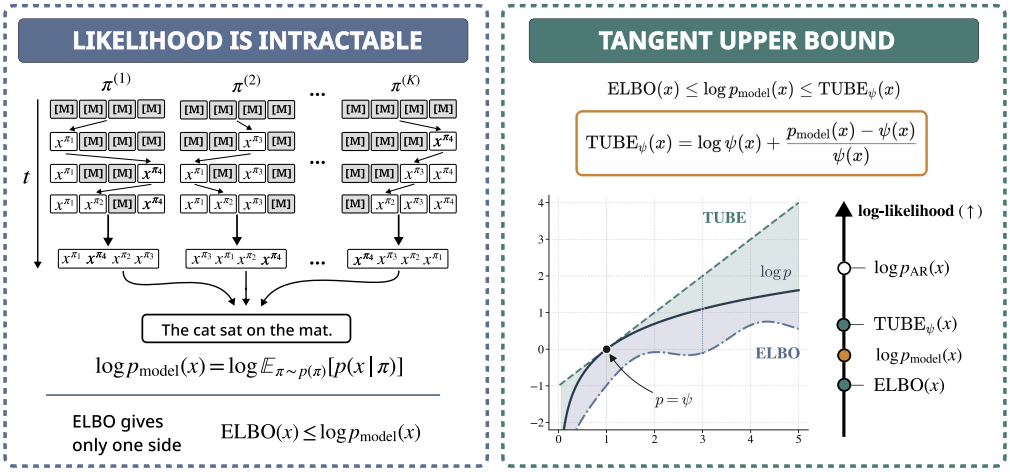


Figure 1: **A tight tangent upper bound on** $\log p_{\text{model}}(x)$**.** Our TUBE bounds the intractable marginal from above using a tractable surrogate ψ, with equality at $\psi = p_{\text{model}}$.

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

[1] AXXX, Russia; [2] Applied AI Institute, Moscow, Russia; [3] MIRAI, Russia.
* Correspondence to: Arseny Ivanov <a5r5s5e5n5y@gmail.com>, Alexander Korotin <iamalexkorotin@gmail.com>

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

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 and 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 likelihoods of AO-ARMs and MDMs, nor can it support rigorous comparisons 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 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 reliable upper bound on the log-likelihood with a tractable surrogate and an unbiased Monte Carlo estimator (§3). Together with the ELBO, TUBE yields a reliable two-sided localization of the log-likelihood.
- **Analysis.** We evaluate TUBE on MDMs and BMs (including LLaDA2.0-mini [Bie et al., 2025]) across OWT and LM1B, obtaining tight bounds for ARM comparison. We find that BMs underperform ARMs under comparable training conditions, while uniform sampling excels at low NFE and stochastic confidence-based sampling at high NFE, with greedy sampling performing worse (§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 introduce a latent-variable view of sequence generation (§2.1), which provides a common framework for the model classes considered throughout the paper. We then instantiate this

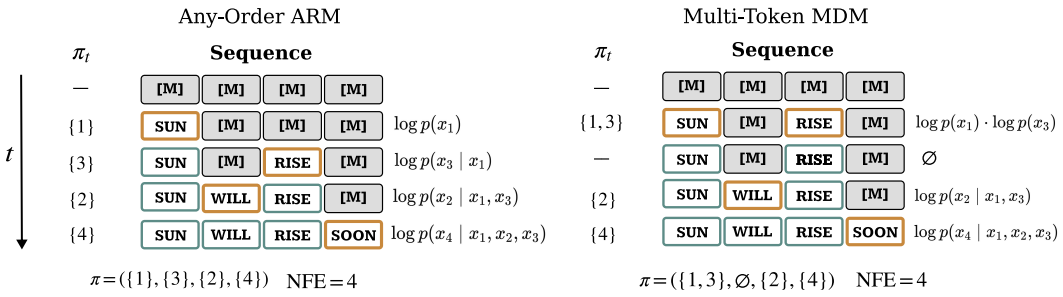


Figure 2: **AO-ARM and MDM.** Both define $p_{\text{model}}(x)$ via a latent generation ordering π. AO-ARM reveals one token per step ($\pi \in \mathcal{S}$). MDM allows zero, one, or many tokens per step ($\pi \in \mathcal{G}$).

framework for autoregressive models (§2.2), any-order autoregressive models (§2.3), and masked diffusion models (§2.4). Next, we show how different sampling strategies, including confidence-based sampling, can be represented within the same framework (§2.5). Finally, we discuss the challenges of likelihood evaluation for latent variable models (§2.6).

2.1 Latent Variable Models (LVMs)

We consider the problem of generating a sequence $x = (x^1, \ldots, x^L) \in \mathcal{V}$ of length L, where each x^l denotes a token. A general way to model the distribution over such sequences is through an LVM. An LVM introduces a latent variable z that controls the generation process and defines the marginal distribution over x as

$$p_{\text{LVM}}(x) = \mathbb{E}_{p(z)}\left[p(x|z)\right]. \tag{1}$$

In practice, modern generative models can often be viewed as specific realizations of this formulation, induced by different choices of the *latent variable* z. While different choices of z give rise to a broad range of LVMs, we focus on the case where $z = \pi$ represents an ordering, with $\pi \in \mathcal{S}$ or $\pi \in \mathcal{G}$ (see §2.2 and §2.4 for their respective definitions). Importantly, this specialization covers any-order autoregressive models and masked diffusion models considered in this paper. Their latent ordering structure makes them particularly amenable to analyzing our method as admits simple and low-variance likelihood estimators (see discussion in Appendix C.3), enabling robust likelihood comparison. Beyond this specialization, we consider a broader setting of latent-variable formulations through the example of uniform diffusion models in Appendix C.3.

2.2 Autoregressive models (ARMs)

Within the ordering-based formulation, the simplest construction corresponds to a deterministic generation order. We write this order as $z = \pi \in \mathcal{S}$, where $\mathcal{S}$ denotes the set of *single-token orderings*. An ARM $p_{\text{ARM}}(x)$ fixes π, i.e., $p(\pi)$ is a degenerate distribution, and factorizes the sequence probability according to it:

$$p_{\text{ARM}}(x) = p(x^{\pi_1}, \ldots, x^{\pi_L}) = p(x^{\pi_1}) \ldots p(x^{\pi_L}|x^{\pi_{L-1}}, \ldots, x^{\pi_1}) = \prod_{t=1}^{L} p(x^{\pi_t}|x^{\pi<t}), \tag{2}$$

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

For ARMs, the fixed decomposition in (2) enables direct log-likelihood estimation as follows:

$$\log p_{\text{ARM}}(x) = \sum_{t=1}^{L} \log p(x^{\pi_t}|x^{\pi<t}). \tag{3}$$

Common choices of π include left-to-right orderings for text [Radford and 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.

2.3 Any-order autoregressive models (AO-ARMs)

To reduce dependence on a fixed order, AO-ARMs [Uria et al., 2014, Shih et al., 2022] instead treat the latent variable as a random generation order $z = \pi \sim p(\pi)$ and average over the corresponding autoregressive decompositions:

$$p_{\text{AO-ARM}}(x) = \mathbb{E}_{\pi\sim p(\pi)}\left[p(x|\pi)\right] = \mathbb{E}_{\pi\sim p(\pi)}\left[\prod_{t=1}^{L} p(x^{\pi_t}|x^{\pi_{<t}})\right], \tag{4}$$

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 since $|\mathcal{S}| = L!$. One may therefore approximate the expectation in (4) using Monte Carlo (MC), 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 the ELBO, which follows from Jensen's inequality:

$$\log p_{\text{AO-ARM}}(x) = \log \mathbb{E}_{\pi\sim p(\pi)}\left[\prod_{t=1}^{L} p(x^{\pi_t}|x^{\pi_{<t}})\right] \geq \mathbb{E}_{\pi\sim p(\pi)}\left[\sum_{t=1}^{L} \log p(x^{\pi_t}|x^{\pi_{<t}})\right]. \tag{5}$$

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

2.4 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 $z = \pi = (\pi_1, \ldots, \pi_T) \in \mathcal{G}$, 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:

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

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 of MDMs also clarifies its relation to AO-ARMs when viewed 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, Equation 2.6], which is equivalent to the AO-ARM ELBO in (5), as shown in [Ou et al., 2025, Theorem 2]. For discretized-time MDMs, however, the relation to AO-ARMs yields a similar ELBO, but over grouped orderings $\pi \in \mathcal{G}$:

$$\log p_{\text{MDM}}(x) = \log \mathbb{E}_{\pi\sim p(\pi)}\left[\prod_{t=1}^{T}\prod_{d\in\pi_t} p(x^d|x^{\pi_{<t}}, \pi)\right] \geq \mathbb{E}_{\pi\sim p(\pi)}\left[\sum_{t=1}^{T}\sum_{d\in\pi_t} \log p(x^d|x^{\pi_{<t}}, \pi)\right]. \tag{7}$$

Notably, both AO-ARMs and MDMs admit block model (BM) variants, in which blocks are generated autoregressively, one after another, and tokens within a block by AO-ARM or MDM; see Appendix C.2.1 for details.

Taken together, the common ordering-based formulation of these models is particularly useful in practice, as it enables comparisons of ARM, AO-ARM, MDM, and BM generation strategies under the same model parameterization, as demonstrated in our experiments (§5).

2.5 Confidence samplers

The LVM view also captures different sampling strategies within the same model class by changing the distribution over the latent ordering or trajectory. More generally, the next block can be selected based on the currently revealed context $\pi_t \sim q(\cdot \mid x^{\pi_{<t}}, \pi_{<t})$, which induces

$$p_{\text{model},q}(x, \pi) = \prod_{t=1}^{T} q(\pi_t \mid x^{\pi_{<t}}, \pi_{<t}) p_{\text{model}}(x^{\pi_t} \mid x^{\pi_{<t}}, \pi_{<t}, \pi_t). \tag{8}$$

Here, p_{model} denotes the model of interest (ARM, AO-ARM, or MDM), while π represents either an AO-ARM ordering, $\pi \in \mathcal{S}$, or an MDM grouped ordering, $\pi \in \mathcal{G}$, recovering either model class.

One possible choice is a confidence-based policy for selecting the next reveal set:

$$q_\gamma(\pi_t = i \mid x^{\pi_{<t}}, \pi_{<t}) = \frac{c_t(i)^\gamma}{\sum_{j \notin \pi_{<t}} c_t(j)^\gamma}, \qquad c_t(i) = \max_v p_{\text{model}}(x^i = v \mid x^{\pi_{<t}}, \pi_{<t}, i), \tag{9}$$

where γ controls the strength of the confidence-based preference. For $\gamma = 0$, the policy is uniform, recovering the random ordering of AO-ARMs and MDMs. Increasing γ to 1 makes positions sampled in proportion to the model's confidence, while in the limit $\gamma \to \infty$, the policy becomes deterministic and greedy, always revealing the most confident position.

2.6 Likelihood evaluation challenges

Likelihood evaluation is generally challenging for LVMs, as marginalizing over the latent variable is often intractable. The models considered above provide concrete examples of this issue. For ARMs, the log-likelihood can be evaluated exactly under a fixed generation order. In contrast, for AO-ARMs and MDMs, the ELBOs in (5) and (7) provide only lower bounds and therefore do not determine the true log-likelihood. BMs in (23) provide an intermediate case, simplifying likelihood evaluation while remaining intractable in general. Consequently, these bounds alone are insufficient for a rigorous likelihood comparison across model classes. This motivates complementing the ELBO with an upper bound on the log-likelihood. If both bounds can be estimated sufficiently tightly, they localize the true log-likelihood and make likelihood-based comparison possible.

3 Method

In this section, 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 discussed 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 the function $\log a$ in a and the tangent majorization of $\log a$ by the Taylor decomposition at the point b:

$$\log a \le \log b + \frac{a - b}{b}, \quad \forall\, a, b > 0. \tag{10}$$

The upper bound has b as the variational variable and allows us to construct a linear bound for $\log p(x)$ by 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:*

$$\log p(x) \;\le\; \text{TUBE}_\psi(x) \;:=\; \log \psi(x) \;+\; \frac{p(x) - \psi(x)}{\psi(x)}, \tag{11}$$

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

This bound, being based on linearization of the logarithm, allows the $p(x)$ term to be linearized. In the context of LVMs of the form (1), this enables unbiased Monte Carlo estimation of the bound; see

§3.2 below. The introduced function $\psi(x)$, which we call the *surrogate*, can be any positive function, and the tightness of TUBE depends on how close $\psi(x)$ is 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 Burda et al. [2016], yields a population two-sided localization of $\log p(x)$ for any ψ:

Two-sided localization of log-likelihood

$$\text{ELBO}(x) \;\le\; \log p(x) \;\le\; \text{TUBE}_\psi(x). \tag{12}$$

3.2 TUBE estimation for AO-ARM and MDM

Monte Carlo estimation. To estimate TUBE_ψ in practice for a model of interest p_{model}, we first consider the LVM formulation p_{LVM} defined in (1) and approximate its marginal likelihood using Monte Carlo sampling. Substituting this approximation into TUBE_ψ yields an **unbiased estimator**:

$$\widehat{\text{TUBE}}_{\psi,K}(x) := \log \psi(x) + \frac{\widehat{p}_{\text{LVM},K}(x) - \psi(x)}{\psi(x)}, \tag{13}$$

$$\widehat{p}_{\text{LVM},K}(x) = \frac{1}{K}\sum_{k=1}^{K}\left[p_{\text{LVM}}(x|z^{(k)})\right], \quad z^{(1)},\dots,z^{(K)} \overset{\text{i.i.d.}}{\sim} p(z)$$

Substituting the specific formulations of AO-ARMs and MDMs into TUBE_ψ, we obtain the corresponding estimators. For AO-ARMs, the Monte Carlo likelihood estimator takes the form:

$$\widehat{p}_{\text{AO-ARM},K}(x) = \frac{1}{K}\sum_{k=1}^{K}\left[\prod_{t=1}^{L} p_{\text{AO-ARM}}(x^{\pi_t^{(k)}}|x^{\pi_{<t}^{(k)}})\right], \quad \pi^{(1)},\dots,\pi^{(K)} \overset{\text{i.i.d.}}{\sim} p(\pi), \quad \pi^{(k)} \in \mathcal{S} \tag{14}$$

and for MDMs with T sampling steps:

$$\widehat{p}_{\text{MDM},K}(x) = \frac{1}{K}\sum_{k=1}^{K}\left[\prod_{t=1}^{T}\prod_{d\in\pi_t^{(k)}} p_{\text{MDM}}(x^d|x^{\pi_{<t}^{(k)}},\pi^{(k)})\right], \pi^{(1)},\dots,\pi^{(K)} \overset{\text{i.i.d.}}{\sim} p(\pi), \pi^{(k)} \in \mathcal{G}. \tag{15}$$

BM extensions simply follow from (23).

In addition, the tightness of TUBE depends on the closeness of the surrogate $\psi(x)$ to $p_{\text{model}}(x)$; thus, its choice is important. Here, we propose to use $p_{\text{model}}(x \mid z)$ itself as a surrogate, while in Appendix C.2.3, we explore alternative surrogate choices and analyze the effect of the self-surrogate parameters. Specifically, suppose that, conditional on an arbitrary latent z, the likelihood $p_{\text{model}}(x \mid z)$ is close to the full model likelihood, i.e., $p_{\text{model}}(x \mid z) \approx p_{\text{model}}(x)$, which is in fact true at the model optimum Ou et al. [2025]. Furthermore, we can take several latent variables $z^{(m)}$ and average the corresponding $p_{\text{model}}(x \mid z^{(m)})$, which in practice should be even closer to $p_{\text{model}}(x)$. We consider:

$$\psi_M(x) = \frac{1}{M}\sum_{m=1}^{M} p_{\text{model}}(x|\hat{z}^{(m)}) \tag{16}$$

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

The computation of TUBE. The practical procedure for two-sided likelihood localization consists of sampling latent variables $z^{(k)}$, evaluating the target model $p_{\text{model}}(x \mid z^{(k)})$, computing the surrogate $\psi(x)$, and then evaluating (13). One can see the detailed algorithm in Appendix C.1. 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 $p_{\text{model}}(x \mid z^{(k)})$. The cost of $\psi_M(x)$ is similar to $\widehat{p}_{\text{model},K}$ and linear in M, i.e., $\mathcal{O}(M)$.

3.3 Theoretical guarantees

In this section, we establish theoretical guarantees on the bias and variance of the Monte Carlo approximation of TUBE, i.e., $\widehat{\mathrm{TUBE}}_{\psi,K}(x)$.

Proposition 3.1 (Bias and Variance of TUBE)**.** *Consider a data point x, a fixed surrogate $\psi(x) > 0$ and the Monte Carlo estimator $\widehat{\mathrm{TUBE}}_{\psi,K}(x)$, see (13), with number of Monte Carlo samples K. Then:*

$$\begin{aligned}\mathrm{Bias}\left[\widehat{\mathrm{TUBE}}_{\psi,K}(x)\right] &:= \mathbb{E}_{\pi^{1:K}}\left[\widehat{\mathrm{TUBE}}_{\psi,K}(x)\right] - \log p_{\text{model}}(x) \\ &= \log \frac{\psi(x)}{p_{\text{model}}(x)} + \frac{p_{\text{model}}(x)}{\psi(x)} - 1 \geq 0 \qquad (17)\end{aligned}$$

$$\mathrm{Var}_{z^{1:K}}\left[\widehat{\mathrm{TUBE}}_{\psi,K}(x)\right] = \frac{\mathrm{Var}_{\pi}[p_{\text{model}}(x \mid z)]}{K\,\psi(x)^2} \qquad (18)$$

One can see that K does not contribute to the bias term. Furthermore, the bias depends only on the variational parameter $\psi(x)$ and is always non-negative. Thus, the upper-bound property is statistically preserved, in contrast to other upper bounds; see, for example, the bias and variance analysis of CUBO in Proposition D.1. Additional theoretical analysis of the bias and variance for the surrogate in (16) is presented in Proposition B.1.

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 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 $\mathbf{ELBO}_K$, 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 and 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 apply to MDMs and are most related to our work: $\mathbf{CUBO}_\beta$, $\mathbf{TVO}_U$, **IS-VG-B**; see Table 1 and Appendix D for summary. Importantly, we emphasize the sign of the bias introduced by Monte Carlo approximation. We call a bound *reliable* if this bias does not act against the direction of the bound, so that the approximation preserves the upper-bound guarantee. The Monte Carlo biases of $\mathbf{CUBO}_\beta$, $\mathbf{TVO}_U$, and **IS-VG-B**, however, have indefinite sign. Consequently, the expectations of their Monte Carlo approximations are not reliable upper bounds and may lie on either side of $\log p_{\text{model}}(x)$.

- **$\mathbf{CUBO}_\beta$ / Rényi** [Li and Turner, 2016, Dieng et al., 2017, Wang et al., 2026] take the form $\frac{1}{\beta}\log \mathbb{E}_\pi[p_{\text{model}}(x|z)^\beta]$ 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 and makes the CUBO unreliable upper bound, see Prop D.1.
- **TVO** [Masrani et al., 2019] is a right Riemann sum of the thermodynamic identity along the geometric path $q_\beta(\pi|x) \propto p_{\text{model}}(x|z)^\beta$, 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 D.2). TVO empirical estimator also has bias of non definite sign Masrani et al. [2019], which makes it unreliable upper bound.

- **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 upper bound-preserving, i.e., unreliable, since both log terms apply Jensen-on-log to MC averages (see Table 1 and Appendix D.3).

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

Table 1: Estimators for various bounds on log-likelihood for latent-variable models. We say that the empirical estimator is *reliable* if its empirical Monte Carlo estimator provably preserves the bound direction, i.e., keeps the Monte Carlo bias non-negative. Bound-breaking terms are highlighted in red; our TUBE's estimator is bound-preserving.

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

5 Experiments

We demonstrate our TUBE through two categories of experiments. In §5.1 we study BMs: we first validate all estimators against the exact likelihood, which is tractable for small blocks, and then scale TUBE to larger blocks and models. In §5.2 we apply TUBE to full-sequence MDMs. The code is available in https://github.com/arsenyivanov/tube

We use the same network weights for ARM, AO-ARM, MDM, and BM evaluations, changing only the Monte Carlo sampling scheme. This is possible because these formulations parameterize the same family of conditionals $p_{\mathrm{model}}(x^d \mid x^{\pi_{<t}}, \pi_{<t})$, $d \in \pi_t$, and the AO-ARM and MDM training objectives have equivalent minimizers [Ou et al., 2025]. Following standard practice, we report perplexity $\mathrm{PPL} = \exp(-\frac{1}{N}\log p_{\mathrm{model}}(x))$ for a sequence of N tokens, where upper bounds on log-likelihood yield lower bounds on PPL, and vice versa.

5.1 Likelihood Estimation in Block Diffusion Models

In these experiments, we ask three questions. Can BMs match ARMs in likelihood? Can existing estimators reliably establish this comparison? How do different generation regimes and sampling strategies affect the likelihood? To address these questions, we evaluate each BM as a block AO-ARM and a block MDM, under uniform or confidence-based sampling [Nie et al., 2025]. We use BD3-LMs [Arriola et al., 2025] with $L' \in \{4, 8, 16\}$ on OWT and LM1B [Gokaslan and Cohen, 2019, Chelba et al., 2014], with the released OWT checkpoints and an LM1B model trained from scratch with the original code. At scale, we use the released LLaDA2.0-mini [Bie et al., 2025] and the ARM Ling-mini-2.0 [Team et al., 2025]. The LM1B results and experimental details are provided in Appendices C.2.2 and E, respectively.

5.1.1 Estimator Comparison and Exact-Likelihood Validation

We first compare TUBE with existing estimators on small-block models with $L' \in \{4, 8, 16\}$, where exact likelihood is tractable for $L' \in \{4, 8\}$. This allows us to validate the estimators against ground truth where available, using shared Monte Carlo samples for TUBE, ELBO_K, and prior estimators.

Table 2: **Test perplexity (↓) of BD3-LMs on OWT** under the uniform and confidence-based trajectory policies q_γ of §2.5. The values are means over 10 seeds. The reference column reports ELBO_K, which is exact for $L' \in \{4, 8\}$ (marked as $=$) and Monte Carlo-estimated 256-trajectory bound for $L' = 16$ (marked $\leq$). At $\gamma = 0$, uniform sampling recovers the native BD3-LM ELBO in expectation. At $\gamma = \infty$, deterministic AO-ARM yields a unique trajectory, making TUBE equal to the exact DUEL likelihood [Turok et al., 2026]. **Color coding:** red background marks biased estimators, and red text marks cells where the value exceeds ELBO_K, violating the expected bound relation.

		Uniform ($\gamma = 0$)						**Confidence** ($\gamma = 1$)			**Confidence** ($\gamma = \infty$, DUEL)		
		Biased upper estimators			Ours	References		Ours	References		Ours	References	
L'	**Regime**	**CUBO**$_{\beta=2}$	**TVO**$_U$	**IS-VG-B**	**≥ TUBE**	**ELBO**$_K$	**≤ ELBO**	**≥ TUBE**	**ELBO**$_K$	**≤ ELBO**	**≥ TUBE**	**ELBO**$_K$	**≤ ELBO**
4	NFE= 1	97.43	97.43	97.43	**97.43**	97.43	97.43	**97.43**	97.43	97.43	**97.43**	97.43	97.43
	NFE= 2	23.39	26.92	26.58	**24.12**	= 26.48	43.23	**24.40**	= 26.71	43.26	**30.12**	= 31.12	45.37
	NFE= 4	19.27	21.51	21.35	**20.35**	= 21.28	29.72	**20.30**	= 21.20	29.47	**22.23**	= 22.77	30.27
	AO-ARM	17.67	18.74	18.67	**18.42**	= 18.66	20.88	**18.28**	= 18.50	20.40	**19.81**	19.81	19.81
8	NFE= 1	218.95	218.95	218.95	**218.95**	218.95	218.95	**218.95**	218.95	218.95	**218.95**	218.95	218.95
	NFE= 2	27.73	32.45	32.37	**27.47**	= 31.63	62.87	**27.92**	= 32.36	65.05	**52.51**	= 54.00	83.20
	NFE= 4	20.27	23.13	23.07	**21.50**	= 22.77	36.20	**21.51**	= 22.84	36.79	**26.60**	= 28.07	44.69
	NFE= 8	18.44	20.58	20.53	**19.80**	= 20.39	27.95	**19.67**	= 20.26	27.89	**22.46**	= 22.81	30.97
	AO-ARM	17.41	18.82	18.79	**18.55**	= 18.75	21.84	**18.32**	= 18.50	21.28	**20.43**	20.43	20.43
16	NFE= 1	392.81	392.81	392.81	**392.81**	392.81	392.81	**392.81**	392.81	392.81	**392.81**	392.81	392.81
	NFE= 2	34.10	38.52	38.87	**25.44**	≤ 37.58	81.53	**25.84**	≤ 38.79	85.74	**99.59**	≤ 106.60	160.19
	NFE= 4	22.28	24.87	24.93	**20.88**	≤ 24.57	41.36	**20.68**	≤ 24.80	43.22	**32.50**	≤ 39.72	72.97
	NFE= 8	19.25	21.23	21.24	**19.55**	≤ 21.10	30.20	**19.35**	≤ 21.01	30.69	**23.78**	≤ 26.71	43.07
	NFE= 16	18.20	19.90	19.90	**18.91**	≤ 19.82	25.97	**18.67**	≤ 19.58	25.88	**21.94**	≤ 22.66	30.96
	AO-ARM	17.44	18.85	18.83	**18.37**	≤ 18.80	22.43	**18.09**	≤ 18.46	21.84	**20.84**	20.84	20.84

Exact ARM baseline: **17.82** PPL

Setup. Specifically, 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 $\in \{1, 2, \ldots, L'\}$), each with the uniform ($\gamma = 0$) and confidence-based ($\gamma \in \{1, \infty\}$) policies of §2.5. Since BD3-LM is not time-conditioned, its prediction for a block depends only on which positions are revealed. For $L' \in \{4, 8\}$ we compute the exact likelihood under every regime and policy and report it as exact in place of ELBO_K. For $L' = 16$ this is intractable, so we sample 256 orderings per block and report ELBO_K over all of them as the reference. All estimators use the same number $(24, 64, 128)$ of orderings per block for $L' = 4, 8, 16$. For TUBE, these are split equally into two independent sets used for $\widehat{p}_{\mathrm{model},K}$ and ψ_M.

Results. At the number of Monte Carlo sampled orderings of Table 2, TUBE never exceeds the reference PPL and is the **tightest valid lower bound** in most configurations, across all NFEs ($= T$ in §2.4) and sampling strategies. In a few low-NFE configurations, $\mathrm{CUBO}_{\beta=2}$ or IS-VG-B come closer to the reference, but lack finite-sample guarantees.

Table 3: **PPL gap at AO-ARM on OWT**, measured as $\mathrm{TUBE} - \mathrm{ARM}$, the minimum across all regimes.

	OWT (ARM 17.82)		
L'	γ=0	γ=1	γ=∞
4	0.60	0.46	1.99
8	0.72	0.49	2.61
16	0.55	**0.27**	3.01

In particular, TVO_U and IS-VG-B sit above ELBO_K in every regime with NFE ≥ 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 (11) 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. Yet, 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. Also, the choice of β can substantially change the estimator bias and even lead to values exceeding ELBO_K, see Appendix D.1. This makes CUBO **unreliable** as well.

Turning to model comparison, Table 3 compares TUBE with the exact ARM baseline in the AO-ARM regime, which yields the smallest gap across generation regimes. Since TUBE provides a lower bound on the true BD3-LM PPL, the strictly positive differences $\mathrm{TUBE} - \mathrm{ARM}$ establish that the ARM achieves lower PPL in every configuration.

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.

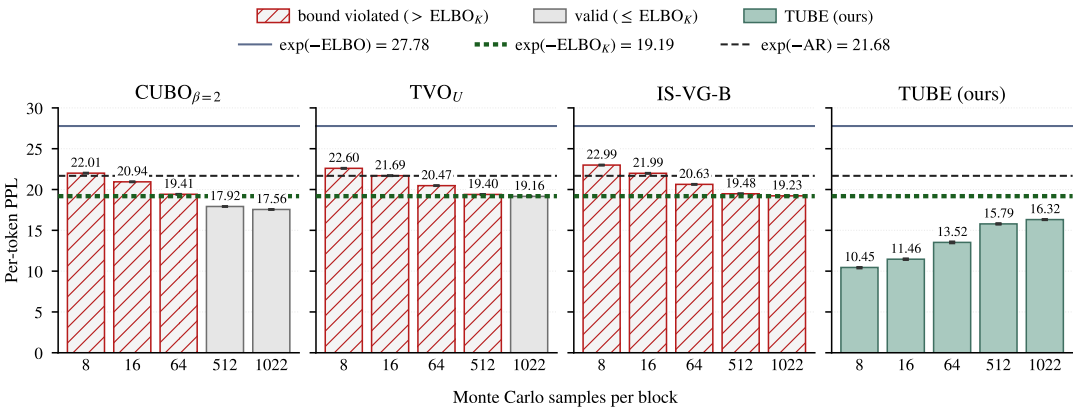


Figure 3: **Validity of PPL estimators for LLaDA2.0-mini in the AO-ARM regime on OWT.** Hatched bars exceed ELBO_K and therefore violate the bound on PPL.

Regarding generation strategies, Table 2 shows that TUBE captures the expected PPL improvement with increasing NFE, as revealing fewer tokens at a time reduces uncertain predictions. Confidence-based sampling ($\gamma = 1$) often outperforms uniform sampling ($\gamma = 0$) by revealing more confident positions first, potentially improving the context for subsequent predictions. However, greedy sampling ($\gamma = \infty$) yields higher PPL despite always selecting the most confident position.

Takeaway. Uniform sampling performs better at low NFE, while stochastic confidence-based sampling generally excels at higher NFE. Greedy sampling consistently underperforms both.

5.1.2 Scaling to Larger Blocks and Models

We next evaluate TUBE on larger blocks and models, where exact likelihood computation becomes intractable. Specifically, we consider LLaDA2.0-mini [Bie et al., 2025], a 16B-parameter MoE BM with 1.4B active parameters and $L' = 32$, twice the largest block considered in §5.1.1.

Table 4: **Test perplexity (↓) of LLaDA2.0-mini on OWT** with $L' = 32$ and 512 orderings per block.

Regime	$\geq$ **TUBE (Ours)**	$\leq$ **ELBO**$_K$	$\leq$ **ELBO**	**Left-to-right**
Ling-mini-2.0 exact ARM likelihood: **21.68** PPL				
NFE= 4	$\geq$ **17.42**$_{\pm 0.07}$	$\leq$ 29.40	$\leq$ 60.40	–
NFE= 16	$\geq$ **15.53**$_{\pm 0.05}$	$\leq$ 20.07	$\leq$ 32.14	–
NFE= 32	$\geq$ **15.88**$_{\pm 0.02}$	$\leq$ 18.31	$\leq$ 26.46	–
AO-ARM	$\geq$ **15.79**$_{\pm 0.03}$	$\leq$ 19.19	$\leq$ 27.78	= 20.24

Setup & Results. AO-ARM samples uniform orderings ($\gamma = 0$), while MDM uses the native confidence-threshold policy of LLaDA2.0 with NFE $\in \{4, 16, 32\}$. Since exact likelihood is intractable at $L' = 32$, we use ELBO_K over all collected orderings as a reference. TUBE yields **tight, non-trivial likelihood intervals** across all 32 evaluated block positions, remaining correctly ordered with ELBO_K rather than collapsing to the trivial interval $[0, \mathrm{ELBO}_K]$. These intervals also enable comparison with the Ling-mini-2.0 ARM reference: at NFE=16, the entire LLaDA2.0-mini interval lies below its exact PPL of 21.68, allowing TUBE to establish an advantage that the native ELBO alone cannot. This is consistent with LLaDA2.0-mini [Team et al., 2025] receiving additional continued pre-training, block-diffusion SFT, and DPO after conversion. Finally, biased estimators exceed ELBO_K with few samples, whereas TUBE retains its bound guarantee in expectation for any sample size (Figure 3).

5.2 Full-Sequence MDMs

To assess scalability with sequence length, we evaluate TUBE on an MDM [Sahoo et al., 2024] with $L = 1024$, where orderings span the entire sequence rather than a generation block. Using 1024 Monte Carlo orderings, TUBE yields non-trivial intervals across all generation regimes. Despite reduced tightness, the intervals remain correctly ordered with ELBO_K, demonstrating the practical feasibility of TUBE at full-sequence scale.

Table 5: **Test PPL (↓) of MDM on OWT** ($L' = 1024$) and 1024 orderings.

Regime	$\geq$ **TUBE (Ours)**	$\leq$ **ELBO**$_K$	$\leq$ **ELBO**
Exact ARM likelihood: **17.82** PPL			
NFE= 128	**8.46**	24.37	27.20
NFE= 512	**8.46**	23.91	26.69
AO-ARM	**7.78**	23.82	26.52

Acknowledgments and Disclosure of Funding

The work was supported by the grant for research centers in the field of AI provided by the Ministry of Economic Development of the Russian Federation in accordance with the agreement 000000C313925P4F0002 and the agreement №139-10-2025-033.

References

Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex X. Lu, Nicolo Fusi, Ava P. Amini, and Kevin K. Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, 2023. doi: 10.1101/2023.09.11.556673.

Marianne Arriola and Volodymyr Kuleshov. Set diffusion: Interpolating token orderings between autoregression and diffusion for fast and flexible decoding, 2026. URL https://arxiv.org/abs/2607.01775.

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, Chengxi Li, Chongxuan Li, Jianguo Li, Zehuan Li, Huabin Liu, Ling Liu, Guoshan Lu, Xiaocheng Lu, Yuxin Ma, Jianfeng Tan, Lanning Wei, Ji-Rong Wen, Yipeng Xing, Xiaolu Zhang, Junbo Zhao, Da Zheng, Jun Zhou, Junlin Zhou, Zhanchao Zhou, Liwang Zhu, and Yihong Zhuang. LLaDA2.0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*, 2025.

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

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling, 2014. URL https://arxiv.org/abs/1312.3005.

Mark Chen, Alec Radford, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, 2020. URL https://api.semanticscholar.org/CorpusID:219781060.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Adji Bousso Dieng, Dustin Tran, Rajesh Ranganath, John Paisley, and David M. Blei. Variational inference via χ-upper bound minimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

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

Samson Gourevitch, Yazid Janati, Dario Shariatian, Umut Simsekli, Eric Moulines, Eric P Xing, and Alain Durmus. Uniform diffusion models revisited: Leave-one-out denoiser and absorbing state reformulation. *arXiv preprint arXiv:2605.22765*, 2026.

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

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

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 30016–30030, 2022.

Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. In *The Tenth International Conference on Learning Representations (ICLR)*, 2022.

Moongyu Jeon, Sangwoo Shin, Dongjae Jeon, and Albert No. Information-theoretic discrete diffusion. *arXiv preprint arXiv:2510.24088*, 2025.

Chunlin Ji and Haige Shen. 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.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Amin Karimi Monsefi, Nikhil Bhendawade, Manuel Rafael Ciosici, Dominic Culver, Yizhe Zhang, and Irina Belousova. FS-DFM: Fast and accurate long text generation with few-step diffusion language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Filip Ekström Kelvinius and Fredrik Lindsten. Discriminator guidance for autoregressive diffusion models. In *International Conference on Artificial Intelligence and Statistics*, pages 3403–3411. PMLR, 2024.

Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, Aditya Grover, and Volodymyr Kuleshov. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.

Seul Lee, Karsten Kreis, Srimukh Prasad Veccham, Meng Liu, Danny Reidenbach, Yuxing Peng, Saee Gopal Paliwal, Weili Nie, and Arash Vahdat. Genmol: A drug discovery generalist with discrete diffusion. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=KM7pXWG1xj.

Xuanlin Li, Brandon Trabucco, Dong Huk Park, Michael Luo, Sheng Shen, Trevor Darrell, and Yang Gao. Discovering non-monotonic autoregressive orderings with variational inference. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jP1vTH3inC.

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

Aaron Lou, Chenlin Meng, and Stefano Ermon. 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.

Vaden Masrani, Tuan Anh Le, and Frank Wood. The thermodynamic variational objective. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *The Thirteenth International Conference on Learning Representations*, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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

Ziqi Pang, Tianyuan Zhang, Fujun Luan, Yunze Man, Hao Tan, Kai Zhang, William T Freeman, and Yu-Xiong Wang. Randar: Decoder-only autoregressive visual generation in random orders. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 45–55, 2025.

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

Patrick Pynadath, Jiaxin Shi, and Ruqi Zhang. Generative frontiers: Why evaluation matters for diffusion language models. *arXiv preprint arXiv:2604.02718*, 2026.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander M. Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Subham Sekhar Sahoo, Zhihan Yang, Yash Akhauri, Johnna Liu, Deepansha Singh, Zhoujun Cheng, Zhengzhong Liu, Eric Xing, John Thickstun, and Arash Vahdat. Esoteric language models. *arXiv preprint arXiv:2506.01928*, 2025.

Tim Salimans, Thomas Mensink, Jonathan Heek, and Emiel Hoogeboom. Beyond single tokens: Distilling discrete diffusion models via discrete MMD. *arXiv preprint arXiv:2603.20155*, 2026.

Hyunjin Seo, Taewon Kim, Sihyun Yu, and SungSoo Ahn. Learning flexible forward trajectories for masked molecular diffusion. *arXiv preprint arXiv:2505.16790*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

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

Łukasz Struski, Marcin Mazur, Paweł Batorski, Przemysław Spurek, and Jacek Tabor. 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*, pages 675–693, 2023.

Ling Team, Ang Li, Ben Liu, Binbin Hu, Bing Li, Bingwei Zeng, Borui Ye, Caizhi Tang, Changxin Tian, Chao Huang, Chao Zhang, Chen Qian, Chenchen Ju, Chenchen Li, Chengfu Tang, Chilin Fu, Chunshao Ren, Chunwei Wu, Cong Zhang, Cunyin Peng, Dafeng Xu, Daixin Wang, Dalong Zhang, Dingnan Jin, Dingyuan Zhu, Dongke Hu, Fangzheng Zhao, Feifan Wu, Feng Zhu, Gangshan Wang, Haitao Zhang, Hailin Zhao, Hanxiao Zhang, Hanzi Wang, Hao Qian, Haoyi Yu, Heng Zhang, Hongliang Zhang, Hongzhi Luan, Huirong Dong, Huizhong Li, Jia Li, Jia Liu, Jialong Zhu, Jian Sha, Jianping Wei, Jiaolong Yang, Jieyue Ma, Jiewei Wu, Jinjing Huang, Jingyun Tian, Jingyuan Zhang, Jinquan Sun, Juanhui Tu, Jun Liu, Jun Xu, Jun Zhou, Junjie Ou, Junpeng Fang, Kaihong

Zhang, Kaiqin Hu, Ke Shi, Kun Tang, Kunlong Chen, Lanyin Mei, Lei Liang, Lei Xu, Libo Zhang, Lin Ju, Lin Yuan, Ling Zhong, Lintao Ma, Lu Liu, Lu Yu, Lun Cai, Meiqi Zhu, Mengying Li, Min Chen, Minghao Xue, Minghong Cai, Mingming Yin, Peijie Jiang, Peilong Zhao, Pingping Liu, Qian Zhao, Qing Cui, Qingxiang Huang, Qingyuan Yang, Quankun Yu, Shaowei Wei, Shijie Lian, Shoujian Zheng, Shun Song, Shungen Zhang, Shuo Zhang, Siyuan Li, Song Liu, Ting Guo, Tong Zhao, Wanli Gu, Weichang Wu, Weiguang Han, Wenjing Fang, Wubin Wang, Xiang Shu, Xiao Shi, Xiaoshun Lan, Xiaolu Zhang, Xiaqing Sun, Xin Zhao, Xingyu Lu, Xiong Xu, Xudong Wang, Xudong Wang, Xuemin Yang, Yajie Yang, Yang Xiang, Yanzhe Li, Yi Zhang, Yilong Wang, Yingxue Li, Yongzhen Guo, Yuzhuo Fu, Yuanyuan Wang, Yue Yang, Yue Yu, Yufeng Deng, Yun Zhang, Yunfei Yu, Yuqi Zhang, Yuxiao He, Zengke Gui, Zhaoxin Huan, Zhaoyang Wang, Zhibo Zhu, Zhihao Wang, Zhiqiang Zhang, Zhoufei Wang, Zihang Zeng, Ziqi Liu, Zitao Xuan, and Zuoli Tang. Every activation boosted: Scaling general reasoner to 1 trillion open language foundation, 2025. URL https://arxiv.org/abs/2510.22115.

Gilad Turok, Chris De Sa, and Volodymyr Kuleshov. DUEL: Exact likelihood for masked diffusion via deterministic unmasking. *arXiv preprint arXiv:2603.01367*, 2026.

Benigno Uria, Iain Murray, and Hugo Larochelle. A deep and tractable density estimator. In *International Conference on Machine Learning*, pages 467–475. PMLR, 2014.

Chenyu Wang, Paria Rashidinejad, DiJia Su, Song Jiang, Sid Wang, Siyan Zhao, Cai Zhou, Shannon Zejiang Shen, Feiyu Chen, Tommi Jaakkola, Yuandong Tian, and Bo Liu. SPG: Sandwiched policy gradient for masked diffusion language models. In *The Fourteenth International Conference on Learning Representations*, 2026. Poster.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. DPLM-2: A multimodal diffusion protein language model. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025a.

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

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. 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 Potential impact and limitations

Limitations. Our empirical study primarily focuses on BMs [Arriola et al., 2025], a common inference-time paradigm of large-scale MDMs such as LLaDA 2.0 [Bie et al., 2025] and Mercury [Khanna et al., 2025]. While we also demonstrate TUBE on full-sequence MDMs (§5.2) and UDMs (Appendix C.3), achieving tight bounds remains challenging. In particular, TUBE's tightness requires a surrogate $\psi(x) \approx p_{\text{model}}(x)$, which becomes difficult to estimate reliably for long sequences ($L \sim 10^2$ to 10^3), where likelihoods can be extremely small. In contrast, BM decomposition (23) with moderate block sizes makes likelihood estimation more tractable and improves localization.

Potential impact. The reliable two-sided estimation of $\log p_{\text{model}}(x)$ 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.

B Theoretical results and proofs

Proof of Proposition 3.1. Since $z^{(1)}, \dots, z^{(K)}$ are i.i.d.,

$$\mathbb{E}_{z^{1:K}}\left[\widehat{p}_{\text{model},K}(x)\right] = \mathbb{E}_{z\sim p(z)}\left[p_{\text{model}}(x \mid z)\right] = p_{\text{model}}(x).$$

Therefore,

$$\begin{aligned}\mathbb{E}_{z^{1:K}}\left[\widehat{\text{TUBE}}_{\psi,K}(x)\right] &= \log\psi(x) + \frac{\mathbb{E}_{z^{1:K}}\left[\widehat{p}_{\text{model},K}(x)\right] - \psi(x)}{\psi(x)} \\ &= \log\psi(x) + \frac{p_{\text{model}}(x)}{\psi(x)} - 1.\end{aligned}$$

Hence,

$$\begin{aligned}\text{Bias}\left[\widehat{\text{TUBE}}_{\psi,K}(x)\right] &= \mathbb{E}_{z^{1:K}}\left[\widehat{\text{TUBE}}_{\psi,K}(x)\right] - \log p_{\text{model}}(x) \\ &= \log\frac{\psi(x)}{p_{\text{model}}(x)} + \frac{p_{\text{model}}(x)}{\psi(x)} - 1 \geq 0,\end{aligned}$$

where the inequality follows from $\log t \leq t - 1$ applied to $t = p_{\text{model}}(x)/\psi(x) > 0$.

For the variance, the only random term in $\widehat{\text{TUBE}}_{\psi,K}(x)$ is $\widehat{p}_{\text{model},K}(x)$, since $\psi(x)$ is fixed. Thus,

$$\text{Var}_{z^{1:K}}\left[\widehat{\text{TUBE}}_{\psi,K}(x)\right] = \frac{1}{\psi(x)^2}\,\text{Var}_{z^{1:K}}\left[\widehat{p}_{\text{model},K}(x)\right].$$

By independence of the K samples,

$$\begin{aligned}\text{Var}_{z^{1:K}}\left[\widehat{p}_{\text{model},K}(x)\right] &= \text{Var}_{z^{1:K}}\left[\frac{1}{K}\sum_{k=1}^{K} p_{\text{model}}(x \mid z^{(k)})\right] \\ &= \frac{1}{K}\,\text{Var}_{z}\left[p_{\text{model}}(x \mid z)\right].\end{aligned}$$

Consequently,

$$\text{Var}_{z^{1:K}}\left[\widehat{\text{TUBE}}_{\psi,K}(x)\right] = \frac{\text{Var}_{z}\left[p_{\text{model}}(x \mid z)\right]}{K\psi(x)^2}.$$

□

Proposition B.1 (Bias Variance Decomposition of TUBE with self-surrogate ψ)**.** *Consider data point x, a self-surrogate $\psi_M(x) = \frac{1}{M}\sum_{m=1}^{M} p_{\text{model}}(x|z)$, see (16), the corresponding Monte Carlo estimator $\widehat{\text{TUBE}}_{\psi_M,K}(x)$, see (13), with number of Monte Carlo samples K and denote by the variance of $\log p(x|z)$ as $V = \text{Var}_{\pi}[p_{model}(x \mid z)]$. Then:*

$$\mathrm{Bias}\big[\widehat{\mathrm{TUBE}}_{\psi_M,K}(x)\big] = \frac{V}{2M\,p_{\mathit{model}}(x)^2} + O(M^{-3/2}) \geq 0 \tag{19}$$

$$\mathrm{Var}_{\pi^{1:K}}\big[\widehat{\mathrm{TUBE}}_{\psi_M,K}(x)\big] = \frac{V}{K\,p_{\mathit{model}}(x)^2} + \frac{3V^2}{MK\,p_{\mathit{model}}(x)^4} + \frac{V^2}{2M^2\,p_{\mathit{model}}(x)^4} + O\Big(M^{-3},\, M^{-3/2}K^{-1}\Big) \tag{20}$$

As a result, the MSE of our TUBE estimation with self-surrogate w.r.t. $\log p$ *can be decomposed into bias and variance in the following way:*

$$\mathbb{E}_{\pi^{1:K}}\Big[\big(\widehat{\mathrm{TUBE}}_{\psi_M,K}(x) - \log p_{\mathit{model}}(x)\big)^2\Big] \tag{21}$$

$$= \underbrace{\frac{V^2}{4M^2\,p_{\mathit{model}}(x)^4}}_{\mathit{Bias}^2} + \underbrace{\frac{V}{K\,p_{\mathit{model}}(x)^2} + \frac{1}{2}\cdot\frac{V^2}{M^2\,p_{\mathit{model}}(x)^4} + \frac{3V^2}{MK\,p_{\mathit{model}}(x)^4}}_{\mathrm{Variance}} + O\Big(M^{-3},\, M^{-3/2}K^{-1}\Big) \tag{22}$$

Proof of Proposition B.1. Let

$$\mu := p_{\mathrm{model}}(x), \qquad Y(z) := p_{\mathrm{model}}(x \mid z), \qquad V := \mathrm{Var}_z[Y(z)],$$

and write

$$\psi_M(x) = \mu + \Delta_M, \qquad \Delta_M := \frac{1}{M}\sum_{m=1}^{M} Y(z^{(m)}) - \mu.$$

Then

$$\mathbb{E}[\Delta_M] = 0, \qquad \mathbb{E}[\Delta_M^2] = \frac{V}{M}.$$

Under the usual finite-moment assumptions,

$$\mathbb{E}[|\Delta_M|^3] = O(M^{-3/2}), \qquad \mathbb{E}[\Delta_M^4] = \frac{3V^2}{M^2} + O(M^{-3}).$$

We assume that the samples used to construct $\psi_M(x)$ are independent of the K samples used in $\widehat{p}_{\mathrm{model},K}(x)$.

Conditionally on $\psi_M(x)$, the result for a fixed surrogate gives

$$\mathbb{E}\Big[\widehat{\mathrm{TUBE}}_{\psi_M,K}(x) \mid \psi_M(x)\Big] = \log\psi_M(x) + \frac{\mu}{\psi_M(x)} - 1.$$

Consequently,

$$\mathrm{Bias}\left[\widehat{\mathrm{TUBE}}_{\psi_M,K}(x)\right] = \mathbb{E}\left[\log\frac{\psi_M(x)}{\mu} + \frac{\mu}{\psi_M(x)} - 1\right].$$

Since $\log t \leq t - 1$ for every $t > 0$, the quantity inside the expectation is nonnegative. Hence the bias is nonnegative.

To obtain its asymptotic expression, define

$$h(s) := \log s + \frac{\mu}{s} - 1.$$

Since

$$h(\mu) = \log\mu, \qquad h'(\mu) = 0, \qquad h''(\mu) = \frac{1}{\mu^2},$$

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

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

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

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

a Taylor expansion around μ yields

$$h(\mu + \Delta_M) = \log\mu + \frac{\Delta_M^2}{2\mu^2} + O(|\Delta_M|^3).$$

Taking expectations gives

$$\mathrm{Bias}\left[\widehat{\mathrm{TUBE}}_{\psi_M,K}(x)\right] = \frac{V}{2M\mu^2} + O(M^{-3/2}) = \frac{V}{2Mp_{\mathrm{model}}(x)^2} + O(M^{-3/2}).$$

For the variance, the law of total variance gives

$$\begin{aligned}\mathrm{Var}\left[\widehat{\mathrm{TUBE}}_{\psi_M,K}(x)\right] &= \mathbb{E}\left[\mathrm{Var}\left(\widehat{\mathrm{TUBE}}_{\psi_M,K}(x) \mid \psi_M(x)\right)\right] \\ &\quad + \mathrm{Var}\left(\mathbb{E}\left[\widehat{\mathrm{TUBE}}_{\psi_M,K}(x) \mid \psi_M(x)\right]\right) \\ &= \frac{V}{K}\mathbb{E}\left[\frac{1}{\psi_M(x)^2}\right] + \mathrm{Var}\left[h(\psi_M(x))\right].\end{aligned}$$

A Taylor expansion of $(\mu + \Delta_M)^{-2}$ gives

$$\frac{1}{(\mu+\Delta_M)^2} = \frac{1}{\mu^2} - \frac{2\Delta_M}{\mu^3} + \frac{3\Delta_M^2}{\mu^4} + O(|\Delta_M|^3),$$

and therefore

$$\mathbb{E}\left[\frac{1}{\psi_M(x)^2}\right] = \frac{1}{\mu^2} + \frac{3V}{M\mu^4} + O(M^{-3/2}).$$

Moreover,

$$h(\mu+\Delta_M) - \log\mu = \frac{\Delta_M^2}{2\mu^2} + O(\Delta_M^3),$$

so

$$\begin{aligned}\mathrm{Var}\left[h(\psi_M(x))\right] &= \frac{1}{4\mu^4}\left(\mathbb{E}[\Delta_M^4] - \mathbb{E}[\Delta_M^2]^2\right) + O(M^{-3}) \\ &= \frac{V^2}{2M^2\mu^4} + O(M^{-3}).\end{aligned}$$

Combining these expansions yields

$$\mathrm{Var}\left[\widehat{\mathrm{TUBE}}_{\psi_M,K}(x)\right] = \frac{V}{K\mu^2} + \frac{3V^2}{MK\mu^4} + \frac{V^2}{2M^2\mu^4} + O\left(M^{-3} + M^{-3/2}K^{-1}\right).$$

Finally, by the standard bias–variance decomposition,

$$\begin{aligned}\mathbb{E}\left[\left(\widehat{\mathrm{TUBE}}_{\psi_M,K}(x) - \log\mu\right)^2\right] &= \mathrm{Bias}\left[\widehat{\mathrm{TUBE}}_{\psi_M,K}(x)\right]^2 + \mathrm{Var}\left[\widehat{\mathrm{TUBE}}_{\psi_M,K}(x)\right] \\ &= \underbrace{\frac{V^2}{4M^2\mu^4}}_{\mathrm{Bias}^2} + \underbrace{\frac{V}{K\mu^2} + \frac{3V^2}{MK\mu^4} + \frac{V^2}{2M^2\mu^4}}_{\mathrm{Variance}} \\ &\quad + O\left(M^{-3} + M^{-3/2}K^{-1}\right).\end{aligned}$$

Substituting $\mu = p_{\mathrm{model}}(x)$ proves the result. □

C Additional Methodology and Experiments

C.1 TUBE Estimation Algorithm

Here we present the Algorithm 1 for the two-sided likelihood localization via our TUBE_ψ (11) and ELBO_K estimate Sahoo et al. [2025] which originates from I-WAE model Burda et al. [2016].

C.2 Block Models (BMs) and Additional Analysis of §5.1.1

In this section, we first formally define the BMs used in our experiments (Appendix C.2.1). We then provide an additional estimator comparison on the LM1B dataset (Appendix C.2.2) and discuss the choice of surrogate function ψ (Appendix C.2.3) following the setup of §5.1.1.

C.2.1 Block models (BMs)

BMs [Arriola et al., 2025, Arriola and Kuleshov, 2026] 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 $\mathcal{B}_1, \dots, \mathcal{B}_B$ with $\bigsqcup_{b=1}^{B} \mathcal{B}_b = \{1, \dots, L\}$. This gives the following block-level decomposition:

$$p_{\text{BM}}(x) = \prod_{b=1}^{B} p(x^{\mathcal{B}_b} | x^{\mathcal{B}_{<b}}), \qquad \text{where } p(x^{\mathcal{B}_b} | x^{\mathcal{B}_{<b}}) \text{ is parameterized as in (4) or (6).} \tag{23}$$

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.

In both parameterizations, the decomposition in (23) 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 (4) and the MDM formulation in (6) 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 (5) and (7), respectively.

C.2.2 Additional estimator comparisons on LM1B

We next provide additional estimator comparisons on the LM1B dataset (Tables 7 and 6). Following the same evaluation protocol as in §5.1.1, we find that the results are consistent with the conclusions obtained on OWT in the main paper.

Table 6: **PPL gap at AO-ARM on LM1B**, measured as $\text{TUBE} - \text{ARM}$, the minimum across all regimes.

	LM1B (ARM 22.85)		
L'	γ=0	γ=1	γ=∞
4	2.78	**2.59**	4.22
8	3.21	2.87	5.05
16	3.08	2.65	5.53

C.2.3 Surrogate ψ ablation: variable sample budget behavior and alternative

In the main text (§3.2), we construct the surrogate directly from p_{model} by averaging $p_{\text{model}}(x \mid z)$ over several latent variables. An alternative surrogate choice is to use a separate model whose density is close to $p_{\text{model}}(x)$, while allowing for fast and exact likelihood computation. A natural candidate is an ARM. Since p_{model} is trained on data from p_{data}, we can train the ARM on the same data, expecting $p_{\text{ARM}} \approx p_{\text{data}} \approx p_{\text{model}}$:

$$\psi_{\text{ARM}}(x) = p_{\text{ARM}}(x) = \prod_{t=1}^{L} p_\psi(x^{\pi_t} \mid x^{\pi_{<t}}). \tag{24}$$

Table 7: **Test perplexity (↓) of BD3-LMs on LM1B** under the uniform and confidence-based trajectory policies (in the format of Table 2). Estimators in a row share 24, 64 and 128 orderings per block for $L' = 4, 8, 16$, and values are means over 10 seeds. The reference column reports ELBO_K, which is exact for $L' \in \{4, 8\}$ (marked as $=$) and Monte Carlo-estimated 256-trajectory bound for $L' = 16$ (marked $\leq$). At $\gamma = 0$, uniform sampling recovers the native BD3-LM ELBO in expectation. At $\gamma = \infty$, deterministic AO-ARM yields a unique trajectory, making TUBE equal to the exact DUEL likelihood [Turok et al., 2026]. **Color coding:** red background marks biased estimators, and red text marks cells where the value exceeds ELBO_K, violating the expected bound relation.

		Uniform ($\gamma = 0$)						**Confidence** ($\gamma = 1$)			**Confidence** ($\gamma = \infty$, DUEL)		
		Biased upper estimators			**Ours**	**References**		**Ours**	**References**		**Ours**	**References**	
L'	**Regime**	**CUBO**$_{\beta=2}$	**TVO**$_U$	**IS-VG-B**	**≥ TUBE**	**ELBO**$_K$	**≤ ELBO**	**≥ TUBE**	**ELBO**$_K$	**≤ ELBO**	**≥ TUBE**	**ELBO**$_K$	**≤ ELBO**
4	NFE= 1	133.17	133.17	133.17	**133.17**	133.17	133.17	**133.17**	133.17	133.17	**133.17**	133.17	133.17
	NFE= 2	32.64	37.53	37.06	**33.87**	= 36.95	59.34	**34.33**	= 37.35	59.59	**41.49**	= 42.86	62.00
	NFE= 4	26.81	29.85	29.64	**28.37**	= 29.55	40.70	**28.35**	= 29.48	40.48	**30.61**	= 31.31	41.38
	AO-ARM	24.68	25.99	25.91	**25.63**	= 25.90	28.43	**25.44**	= 25.68	27.81	**27.08**	27.08	27.08
8	NFE= 1	291.66	291.66	291.66	**291.66**	291.66	291.66	**291.66**	291.66	291.66	**291.66**	291.66	291.66
	NFE= 2	39.05	45.66	45.51	**39.36**	= 44.62	85.96	**39.97**	= 45.63	88.96	**69.53**	= 71.21	108.29
	NFE= 4	28.49	32.47	32.38	**30.44**	= 32.01	49.73	**30.47**	= 32.12	50.61	**36.63**	= 38.37	59.56
	NFE= 8	25.93	28.84	28.77	**27.88**	= 28.60	38.37	**27.71**	= 28.42	38.32	**30.89**	= 31.33	41.89
	AO-ARM	24.58	26.37	26.33	**26.07**	= 26.30	29.89	**25.72**	= 25.91	29.11	**27.90**	27.90	27.90
16	NFE= 1	537.47	537.47	537.47	**537.47**	537.47	537.47	**537.47**	537.47	537.47	**537.47**	537.47	537.47
	NFE= 2	47.87	54.10	54.53	**36.18**	≤ 52.89	112.52	**36.66**	≤ 54.26	117.69	**127.02**	≤ 134.18	201.09
	NFE= 4	31.32	34.97	35.03	**29.72**	≤ 34.57	57.20	**29.55**	≤ 34.93	59.78	**43.89**	≤ 52.79	95.07
	NFE= 8	27.04	29.80	29.80	**27.71**	≤ 29.62	41.62	**27.50**	≤ 29.53	42.34	**32.93**	≤ 36.44	57.52
	NFE= 16	25.57	27.91	27.89	**26.77**	≤ 27.80	35.67	**26.49**	≤ 27.50	35.56	**30.26**	≤ 31.13	41.95
	AO-ARM	24.57	26.41	26.39	**25.93**	≤ 26.36	30.66	**25.50**	≤ 25.87	29.82	**28.39**	28.39	28.39

Exact ARM baseline: **22.85** PPL

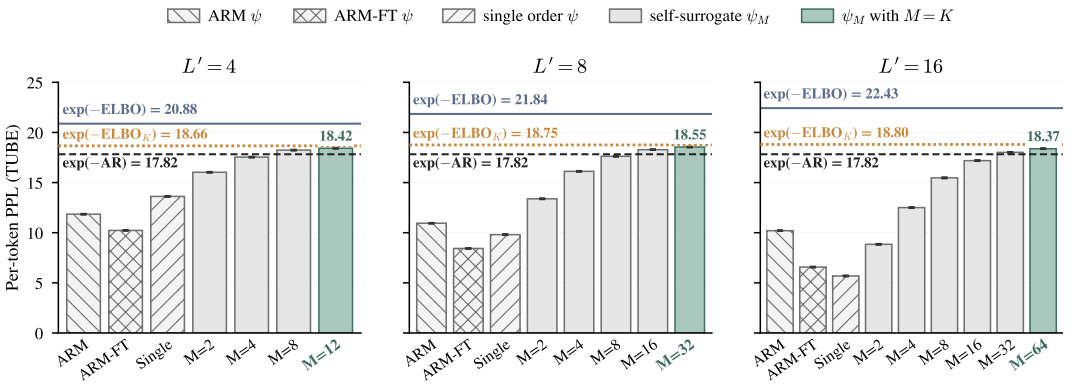


Figure 4: **Test perplexity (↓) of the TUBE with different surrogates ψ on OWT.** Bars show the per-token PPL $= \exp(-\log p_{\mathrm{model}}(x))$ given by the TUBE, with $\hat{p}$ from $K = 12, 32, 64$ orderings for $L' = 4, 8, 16$. ARM and ARM-FT use the exact likelihood of the ARM and of the ARM fine-tuned on $p_{\mathrm{model}}(x)$ samples as ψ_{ARM}. Single uses one ordering as ψ_π. M uses the self-surrogate ψ_M from M other orderings, and $M = K$ (green) is the TUBE of Table 2. Error bars are standard deviations over 10 seeds.

Furthermore, to bring the surrogate ψ_{ARM} closer to p_{model}, one can fine-tune the ARM on synthetic data generated by p_{model}. This can be done using standard likelihood maximization Radford and Narasimhan [2018] or more elaborate procedures Ouyang et al. [2022].

We next compare the proposed self-surrogate with the ARM-based alternative and examine the self surrogate sample budget behavior as the number of sampled orderings M varies. Since the tightness of TUBE depends on the choice of ψ, this allows us to assess both how different surrogate constructions affect the bound and how quickly the self-surrogate improves as more orderings are included.

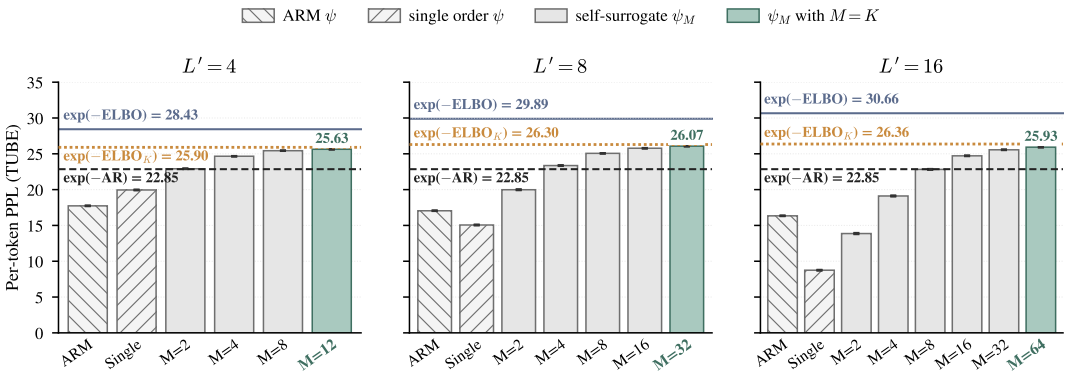


Figure 5: **Test perplexity (↓) of the TUBE with different surrogates ψ on LM1B.** We report $\mathrm{PPL} = \exp(-\log p_{\mathrm{model}}(x))$. The bar plot reports standard deviations over 10 repeated ordering samples with different random seeds. The dotted gold reference line shows ELBO_K. ARM-derived$_\psi$ stands for ARM surrogate ψ_{ARM}, while Mixture$_\psi$ stands for self-surrogate.

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 p_{model} (see Appendix E for fine-tuning details). We vary the number of sampled orderings M on a $\log_2$ grid up to the full ordering budget used in §5.1.1.

Results. Figures 4 and 5 shows that the TUBE with self-surrogate ψ_M gets closest to ELBO_K at every block size with increase of M. The TUBE combined with external ARM surrogate ψ_{ARM} leaves a much wider gap, and fine-tuning it on BM $p_{\mathrm{model}}(x)$ samples, see Figure 4, 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.

C.3 Uniform Diffusion Models (UDMs) and Likelihood Estimator Variance

The ordering-based models considered in the main text constitute a particularly convenient instance of the general LVM formulation in §2.1. More generally, a diffusion model can be viewed as an LVM by taking the full diffusion trajectory $z = x_{1:T}$ as the latent variable. For a reverse diffusion model

$$p_{\mathrm{model}}(x_{0:T}) = p_T(x_T) \prod_{t=1}^{T} p_{\mathrm{model}}(x_{t-1} \mid x_t), \tag{25}$$

the data likelihood is obtained by marginalizing out the latent trajectory,

$$p_{\mathrm{model}}(x_0) = \sum_{x_{1:T}} p_{\mathrm{model}}(x_{0:T}). \tag{26}$$

For a UDM, the forward corruption process independently replaces each token with a uniformly sampled vocabulary element, whereas for an MDM, tokens are replaced with a special mask token [MASK]. Their time-t marginals are given by

$$q_{\mathrm{UDM}}(x_t^d \mid x_0^d) = \alpha_t \mathbb{1}[x_t^d = x_0^d] + (1 - \alpha_t)\frac{1}{|\mathcal{V}|}, \tag{27}$$

$$q_{\mathrm{MDM}}(x_t^d \mid x_0^d) = \alpha_t \mathbb{1}[x_t^d = x_0^d] + (1 - \alpha_t)\mathbb{1}[x_t^d = [\mathrm{MASK}]]. \tag{28}$$

where $\alpha_t \in [0, 1]$ controls the probability of retaining the original token, with $\alpha_0 = 1$ and $\alpha_T \approx 0$.

Using the corresponding forward process $q(x_{1:T} \mid x_0)$ as a proposal, (26) admits the standard importance-sampling representation

$$p_{\mathrm{model}}(x_0) = \mathbb{E}_{x_{1:T} \sim q(\cdot|x_0)} \left[\frac{p_{\mathrm{model}}(x_{0:T})}{q(x_{1:T} \mid x_0)} \right]. \tag{29}$$

Table 8: **Estimator comparison on LM1B** with sequence length $L = 128$ and $K + M = 2048$, using the checkpoints of Gourevitch et al. [2026].

Model	Latent z	$\geq$ **TUBE (Ours)**	$\leq$ **ELBO**$_K$	$\leq$ **ELBO**
UDM	$x_{1:T}$ (naive-IS)	23.98 ± 0.60	29.69	35.70
AUDM	(u, π)	23.47 ± 0.23	27.51	31.67
MDM	$x_{1:T}$ (naive-IS)	21.56 ± 0.39	26.75	31.20
MDM	π	21.09 ± 0.41	26.36	30.49

This immediately gives an unbiased Monte Carlo likelihood estimator,

$$\widehat{p}^{\,\text{naive}}_{\text{model},K}(x_0) = \frac{1}{K}\sum_{k=1}^{K} \frac{p_{\text{model}}(x_0, x_{1:T}^{(k)})}{q(x_{1:T}^{(k)} \mid x_0)}, \qquad x_{1:T}^{(k)} \sim q(\cdot \mid x_0). \tag{30}$$

In practice we start the bridge from the model prior, $r(x_{1:T} \mid x_0) = p_T(x_T)\prod_{t=2}^{T} q(x_{t-1} \mid x_t, x_0)$, which is also a valid proposal: the prior factors cancel in the weight, and every proposed path ends at x_0. Thus, TUBE can be applied directly to both MDMs and UDMs using samples of the full latent trajectory. We refer to this approach as *naive-IS*.

For MDMs, however, the full trajectory contains unnecessary randomness. Conditional on the unmasking ordering π, the sequence of intermediate masked states is determined, allowing the trajectory representation to be reduced to the ordering-based likelihood $p_{\text{MDM}}(x_0 \mid \pi)$ used in the main text. This removes the need to explicitly sample $x_{1:T}$ and leads to a substantially more stable likelihood estimator.

For UDM, the same simplification does not follow directly. Conditioning on an ordering specifies when positions are updated, but the intermediate states $x_{1:T}$ remain random because corrupted positions can take any value in $\mathcal{V}$. Consequently, the ordering alone does not determine the diffusion trajectory, and the naive estimator in (30) retains the additional trajectory-level variance.

A more structured representation can be obtained through the absorbing-state reformulation of UDMs proposed by Gourevitch et al. [2026]. Let

$$u \sim \nu = \text{Uniform}(\mathcal{V}) \tag{31}$$

be an auxiliary sequence of absorbing tokens. Conditional on u, each coordinate follows an absorbing-state corruption process,

$$q_{\text{AUDM}}(x_t^d \mid x_0^d) = \alpha_t \mathbb{1}[x_t^d = x_0^d] + (1 - \alpha_t)\mathbb{1}[x_t^d = u^d], \tag{32}$$

Marginalizing over $u^d \sim \text{Uniform}(\mathcal{V})$ recovers the uniform corruption marginal in (27). Conditional on u, the resulting process has the same absorbing-state structure as a masked diffusion process, so its trajectory can again be represented by an ordering π. This yields

$$p_{\text{AUDM}}(x_0) = \mathbb{E}_{u\sim\nu}\mathbb{E}_{\pi\sim p(\pi)}\left[p_{\text{AUDM}}(x_0 \mid \pi, u)\right]. \tag{33}$$

Hence, the latent variable can be taken as $z = (u, \pi)$ rather than the entire sequence of intermediate states $x_{1:T}$. This representation preserves an unbiased likelihood estimator while removing a substantial part of the trajectory-level randomness present in naive-IS.

Evaluation. We evaluate these representations on LM1B using the released UDM, AUDM and MDM checkpoints of Gourevitch et al. [2026]. We use sequences of length $L = 128$ and set $K = M = 1024$, where K denotes the number of latent samples used to estimate $\widehat{p}_{\text{model,K}}$ and M the number of independent samples used to construct the self-surrogate. We repeat each stochastic evaluation over 10 independent random seeds and report the mean and standard deviation. For reference, we additionally report the mean and standard deviation over 10 random splits of the 2048 samples into the two halves.

The results highlight the importance of the latent representation used for likelihood estimation. For UDM naive-IS, explicitly sampling the full trajectory results in a TUBE PPL standard deviation of 0.60 and an ELBO$_K$ PPL of 29.69. The absorbing-state representation reduces the TUBE standard

deviation to 0.24, while the gap ELBO - ELBO_K PPL decreases from 6.01 for UDM naive-IS to 4.16 for AUDM. For MDM, the importance weight of naive-IS reduces to the ordering likelihood with several positions revealed at the same step, so the two representations give close estimates, and the ordering-based representation used in the main text tightens ELBO_K from 26.75 to 26.36.

Overall, TUBE applies to any diffusion process with a tractable unbiased likelihood estimator, not only to ordering-based models. The choice of latent representation affects finite-sample variance, and designing low-variance estimators for general diffusion processes remains future work.

D 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 (13). Each estimator below uses this single shared sample bank.

D.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 and Turner, 2016] are two equivalent formulations of the same family of upper bounds on $\log p_{\mathrm{model}}(x)$. In our latent-mixture form,

$$\mathrm{CUBO}_\beta(x) \;=\; \frac{1}{\beta}\log \mathbb{E}_{\pi\sim p(\pi)}\big[p_{\mathrm{model}}(x|\pi)^\beta\big] \;\geq\; \log p_{\mathrm{model}}(x), \qquad \beta \geq 1, \tag{34}$$

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

$$\widehat{\mathrm{CUBO}}_{\beta,K}(x) \;=\; \frac{1}{\beta}\log\left(\frac{1}{K}\sum_{k=1}^{K} p_{\mathrm{model}}(x|\pi^{(k)})^\beta\right). \tag{35}$$

Source of bias. The estimator (35) 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. In Prop D.1 we formalize this non reliable bias property.

From empirical point of view the Figure 6 shows configurations of $(|\mathcal{B}|, \beta)$ where this occurs in practice. SPG [Wang et al., 2026] also observes this property 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.

Proposition D.1 (Bias, variance and MSE of the CUBO estimator)**.** *Fix $x \in \mathcal{V}$, $\beta \geq 1$, and $\pi^{(1)}, \ldots, \pi^{(K)} \overset{\text{i.i.d.}}{\sim} p(z)$, and recall the empirical CUBO estimator of Table 1, $\widehat{\mathrm{CUBO}}_{\beta,K}(x) := \frac{1}{\beta}\log\big(\frac{1}{K}\sum_{k=1}^{K} p_{\mathrm{model}}(x \mid z^{(k)})^\beta\big)$. Writing $m_n := \mathbb{E}_{\pi\sim p(z)}[p_{\mathrm{model}}(x \mid z)^n]$, so that $\mathrm{CUBO}_\beta(x) = \frac{1}{\beta}\log m_\beta$ and $m_1 = p_{\mathrm{model}}(x)$, define the* relative variance *of $p_{\mathrm{model}}(x \mid z)^\beta$, the* population gap, *and the third central moment,*

$$\rho_\beta^2 := \frac{m_{2\beta}}{m_\beta^2} - 1, \qquad G_\beta := \mathrm{CUBO}_\beta(x) - \log p_{\mathrm{model}}(x), \qquad \kappa_3 := \mathbb{E}_{z\sim p(z)}\Big[\big(p_{\mathrm{model}}(x \mid z)^\beta - m_\beta\big)^3\Big]. \tag{36}$$

Assume $\mathcal{S}$ (resp. $\mathcal{G}$) is finite and $p_{\mathrm{model}}(x \mid z) > 0$ on the support of $p(z)$, so that $p_{\mathrm{model}}(x \mid z)$ is bounded away from 0 and ∞ almost surely. Then $\rho_\beta^2 \geq 0$ and $G_\beta \geq 0$ by convexity of $n \mapsto \log m_n$, and for every $K \geq 1$,

$$\Big[\widehat{\mathrm{CUBO}}_{\beta,K}(x)\Big] := \Big[\widehat{\mathrm{CUBO}}_{\beta,K}(x)\Big] - \log p_{\mathrm{model}}(x) = G_\beta - \frac{\rho_\beta^2}{2\beta K} + O\big(K^{-2}\big), \tag{37}$$

$$\left[\widehat{\mathrm{CUBO}}_{\beta,K}(x)\right] = \frac{\rho_\beta^2}{\beta^2 K} + \frac{1}{\beta^2 K^2}\left(\frac{5\rho_\beta^4}{2} - \frac{\kappa_3}{m_\beta^3}\right) + O\big(K^{-3}\big)\,, \tag{38}$$

and, combining the two,

$$\mathbb{E}\left[\left(\widehat{\mathrm{CUBO}}_{\beta,K}(x) - \log p_{\mathrm{model}}(x)\right)^2\right] = G_\beta^2 + \frac{\rho_\beta^2}{\beta K}\left(\frac{1}{\beta} - G_\beta\right) + O\big(K^{-2}\big)\,. \tag{39}$$

Proof of Proposition D.1. Let

$$Y(z) := p_{\mathrm{model}}(x \mid z)^\beta, \qquad \mu := \mathbb{E}_z[Y(z)] = m_\beta,$$

and define

$$\overline{Y}_K := \frac{1}{K}\sum_{k=1}^{K} Y(z^{(k)}), \qquad T_K := \frac{\overline{Y}_K - \mu}{\mu}.$$

Then

$$\widehat{\mathrm{CUBO}}_{\beta,K}(x) = \frac{1}{\beta}\log \overline{Y}_K = \mathrm{CUBO}_\beta(x) + \frac{1}{\beta}\log(1 + T_K).$$

Since

$$\mathrm{Var}_z[Y(z)] = m_{2\beta} - m_\beta^2,$$

we have

$$\rho_\beta^2 = \frac{\mathrm{Var}_z[Y(z)]}{m_\beta^2} \geq 0.$$

Moreover, by the monotonicity of L^p norms, or equivalently by convexity of $n \mapsto \log m_n$,

$$m_\beta^{1/\beta} \geq m_1 = p_{\mathrm{model}}(x),$$

and hence

$$G_\beta = \frac{1}{\beta}\log m_\beta - \log m_1 \geq 0.$$

We next record the moments of T_K. Independence and centering give

$$\mathbb{E}[T_K] = 0, \qquad \mathbb{E}[T_K^2] = \frac{\rho_\beta^2}{K}, \qquad \mathbb{E}[T_K^3] = \frac{\kappa_3}{m_\beta^3 K^2}.$$

Furthermore, the fourth-moment identity for an average of i.i.d. centered variables yields

$$\mathbb{E}[T_K^4] = \frac{3\rho_\beta^4}{K^2} + O(K^{-3}).$$

Because $Y(z)$ is bounded away from zero and infinity, the Taylor expansions below have uniformly controlled remainders. In particular, the higher-order moments of T_K contribute only to the stated remainder terms.

Using

$$\log(1+t) = t - \frac{t^2}{2} + \frac{t^3}{3} - \frac{t^4}{4} + O(t^5),$$

we obtain

$$\begin{aligned}\mathbb{E}[\log(1+T_K)] &= -\frac{1}{2}\mathbb{E}[T_K^2] + \frac{1}{3}\mathbb{E}[T_K^3] - \frac{1}{4}\mathbb{E}[T_K^4] + O(K^{-3}) \\ &= -\frac{\rho_\beta^2}{2K} + \frac{1}{K^2}\left(\frac{\kappa_3}{3m_\beta^3} - \frac{3\rho_\beta^4}{4}\right) + O(K^{-3}).\end{aligned}$$

Consequently,

$$\begin{aligned}\mathrm{Bias}\left[\widehat{\mathrm{CUBO}}_{\beta,K}(x)\right] &= \mathbb{E}\left[\widehat{\mathrm{CUBO}}_{\beta,K}(x)\right] - \log p_{\mathrm{model}}(x) \\ &= G_\beta + \frac{1}{\beta}\mathbb{E}[\log(1+T_K)]\end{aligned}$$

$$= G_\beta - \frac{\rho_\beta^2}{2\beta K} + O(K^{-2}),$$

which proves (37). Notice also that Jensen's inequality gives

$$\mathbb{E}\left[\widehat{\mathrm{CUBO}}_{\beta,K}(x)\right] \le \frac{1}{\beta}\log \mathbb{E}[\overline{Y}_K] = \mathrm{CUBO}_\beta(x),$$

so the finite-K Monte Carlo correction is necessarily nonpositive.

For the variance, the deterministic term $\mathrm{CUBO}_\beta(x)$ does not contribute, and therefore

$$\mathrm{Var}\left[\widehat{\mathrm{CUBO}}_{\beta,K}(x)\right] = \frac{1}{\beta^2}\,\mathrm{Var}\left[\log(1+T_K)\right].$$

Squaring the Taylor expansion gives

$$\log(1+t)^2 = t^2 - t^3 + \frac{11}{12}t^4 + O(t^5).$$

It follows that

$$\begin{aligned}\mathbb{E}\left[\log(1+T_K)^2\right] &= \mathbb{E}[T_K^2] - \mathbb{E}[T_K^3] + \frac{11}{12}\mathbb{E}[T_K^4] + O(K^{-3}) \\ &= \frac{\rho_\beta^2}{K} + \frac{1}{K^2}\left(\frac{11\rho_\beta^4}{4} - \frac{\kappa_3}{m_\beta^3}\right) + O(K^{-3}).\end{aligned}$$

Moreover,

$$\mathbb{E}[\log(1+T_K)]^2 = \frac{\rho_\beta^4}{4K^2} + O(K^{-3}).$$

Thus,

$$\mathrm{Var}\left[\log(1+T_K)\right] = \frac{\rho_\beta^2}{K} + \frac{1}{K^2}\left(\frac{5\rho_\beta^4}{2} - \frac{\kappa_3}{m_\beta^3}\right) + O(K^{-3}),$$

and hence

$$\mathrm{Var}\left[\widehat{\mathrm{CUBO}}_{\beta,K}(x)\right] = \frac{\rho_\beta^2}{\beta^2 K} + \frac{1}{\beta^2 K^2}\left(\frac{5\rho_\beta^4}{2} - \frac{\kappa_3}{m_\beta^3}\right) + O(K^{-3}),$$

which proves (38).

Finally, the bias–variance decomposition gives

$$\mathbb{E}\left[\left(\widehat{\mathrm{CUBO}}_{\beta,K}(x) - \log p_{\mathrm{model}}(x)\right)^2\right] = \mathrm{Bias}\left[\widehat{\mathrm{CUBO}}_{\beta,K}(x)\right]^2 + \mathrm{Var}\left[\widehat{\mathrm{CUBO}}_{\beta,K}(x)\right].$$

Using

$$\mathrm{Bias}\left[\widehat{\mathrm{CUBO}}_{\beta,K}(x)\right]^2 = G_\beta^2 - \frac{G_\beta \rho_\beta^2}{\beta K} + O(K^{-2}),$$

together with the leading variance term, we obtain

$$\mathbb{E}\left[\left(\widehat{\mathrm{CUBO}}_{\beta,K}(x) - \log p_{\mathrm{model}}(x)\right)^2\right] = G_\beta^2 + \frac{\rho_\beta^2}{\beta K}\left(\frac{1}{\beta} - G_\beta\right) + O(K^{-2}),$$

which proves (39). □

Connection to SPG. SPG's "evidence upper bound" (EUBO) [Wang et al., 2026] is CUBO_β specialized to the MDM absorbing forward process [Sahoo et al., 2024]: SPG's Lemma 1 is (34) under $n \leftrightarrow \beta$, and Theorem 1 specializes it to the per-token MDM kernel. This is distinct from the KL-based EUBO of [Ji and Shen, 2019], which requires posterior samples.

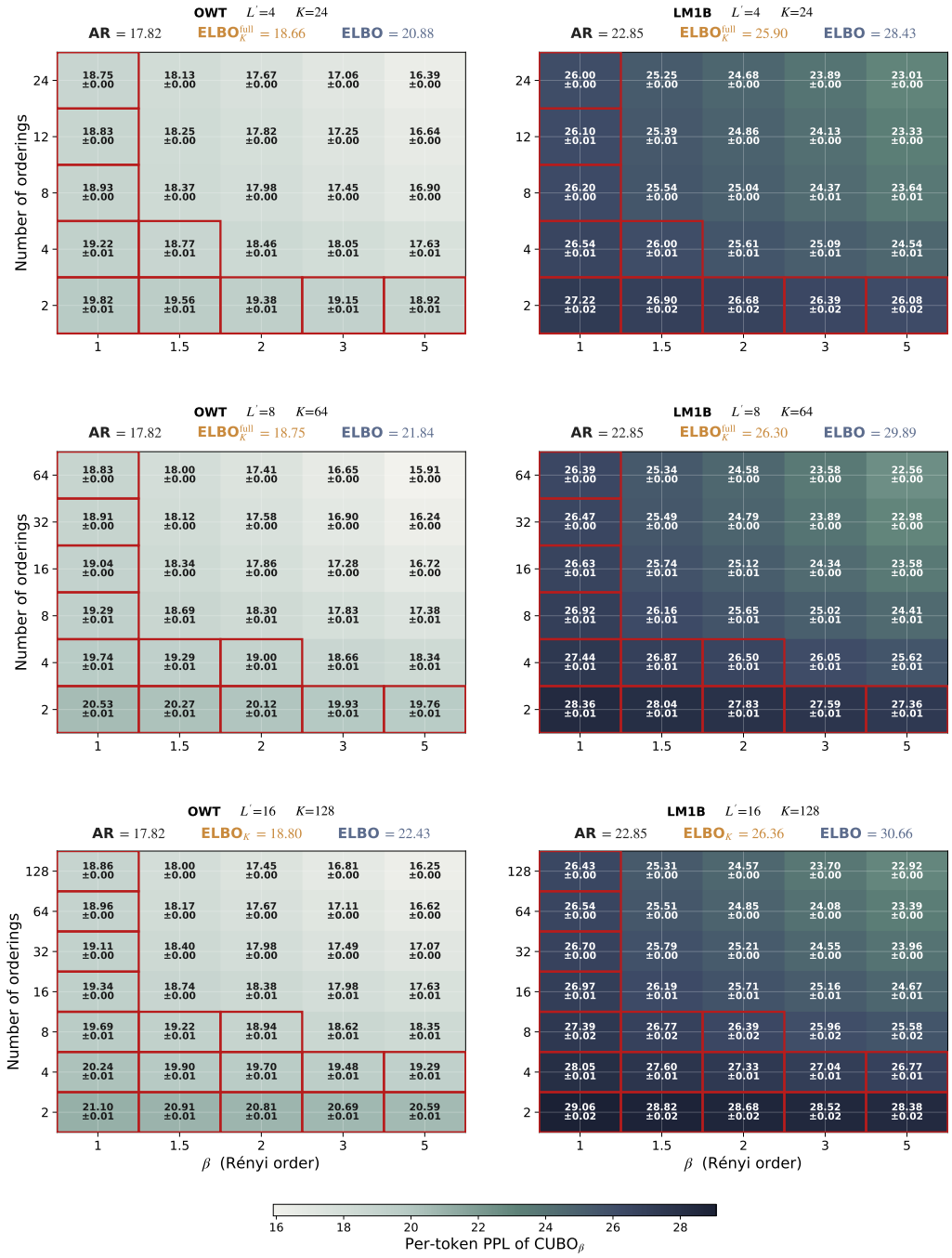


Figure 6: **CUBO$_\beta$ diagnostic on $L' \in \{4, 8, 16\}$.** PPL of CUBO$_\beta$ as a function of the number of orderings and the Rényi exponent β, on OWT (left) and LM1B (right). Red borders mark cells exceeding $\mathrm{ELBO}_K^{\mathrm{full}}$, where CUBO violates its nominal upper-bound guarantee. 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.

D.2 TVO: Thermodynamic-integration Upper Bound

The TVO of [Masrani et al., 2019] starts from the thermodynamic identity

$$\log p_{\mathrm{model}}(x) \;=\; \int_0^1 \mathbb{E}_{\pi \sim q_\beta(\cdot|x)}[\log p_{\mathrm{model}}(x|\pi)]\; d\beta, \qquad q_\beta(\pi|x) \propto p_{\mathrm{model}}(x|\pi)^\beta, \tag{40}$$

where q_β interpolates between the prior $q_0 = p(\pi)$ and the posterior $q_1 = p_{\text{model}}(\pi|x)$. 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:

$$\log p_{\text{model}}(x) \;\le\; \frac{1}{\Lambda}\sum_{\lambda=1}^{\Lambda} \mathbb{E}_{\pi\sim q_{\beta_\lambda}(\cdot|x)}[\log p_{\text{model}}(x|\pi)] \;=:\; \text{TVO}^U_\Lambda(x). \tag{41}$$

Empirical Monte Carlo estimator. Using the same K sampled orderings as in (13) with self-normalized weights $\bar{w}_k^{(\beta)} = p_{\text{model}}(x|\pi^{(k)})^\beta \,/\, \sum_j p_{\text{model}}(x|\pi^{(j)})^\beta$, the empirical MC estimator (Table 1) is

$$\widehat{\text{TVO}}^U_{\Lambda,K}(x) \;=\; \frac{1}{\Lambda}\sum_{\lambda=1}^{\Lambda}\sum_{k=1}^{K} \bar{w}_k^{(\beta_\lambda)}\, \log p_{\text{model}}(x|\pi^{(k)}). \tag{42}$$

Sources of bias. Two structural sources, both visible in Figure 7. *(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, $\mathbb{E}[\widehat{\text{TVO}}^U_{\Lambda,K}] \neq \text{TVO}^U_\Lambda$ at finite K, and the population upper-bound guarantee no longer ensures $\mathbb{E}[\widehat{\text{TVO}}^U_{\Lambda,K}] \geq \log p_{\text{model}}(x)$. *(ii)* The Riemann discretization adds an $\mathcal{O}(1/\Lambda)$ residual that does not vanish at fixed Λ.

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

IS-VG-B [Struski et al., 2023] constructs an upper bound on $\log p_{\text{model}}(x)$ 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.

The starting two-point inequality is

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

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

$$\text{IS-VG-B}_s(x) \;:=\; \underbrace{\mathbb{E}[\log \widehat{p}_{\text{model},s}(x)]}_{=\,\text{ELBO}_s} \;+\; \underbrace{\log\, \mathbb{E}\left[\frac{\widetilde{p}_{\text{model},s}(x)}{\widehat{p}_{\text{model},s}(x)}\right]}_{\text{correction term}} \;\geq\; \log p_{\text{model}}(x), \tag{43}$$

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 (13). 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:

$$\widehat{\text{IS-VG-B}}_{s,n_p}(x) \;=\; \frac{1}{n_p}\sum_{j=1}^{n_p} \log\left(\frac{1}{s}\sum_{i=1}^{s} p_{\text{model}}(x|\pi^{(j,i)})\right) + \log\left(\frac{1}{n_p}\sum_{j=1}^{n_p} \frac{\sum_{i=1}^{s} p_{\text{model}}(x|\widetilde{\pi}^{(j,i)})}{\sum_{i=1}^{s} p_{\text{model}}(x|\pi^{(j,i)})}\right), \tag{44}$$

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.

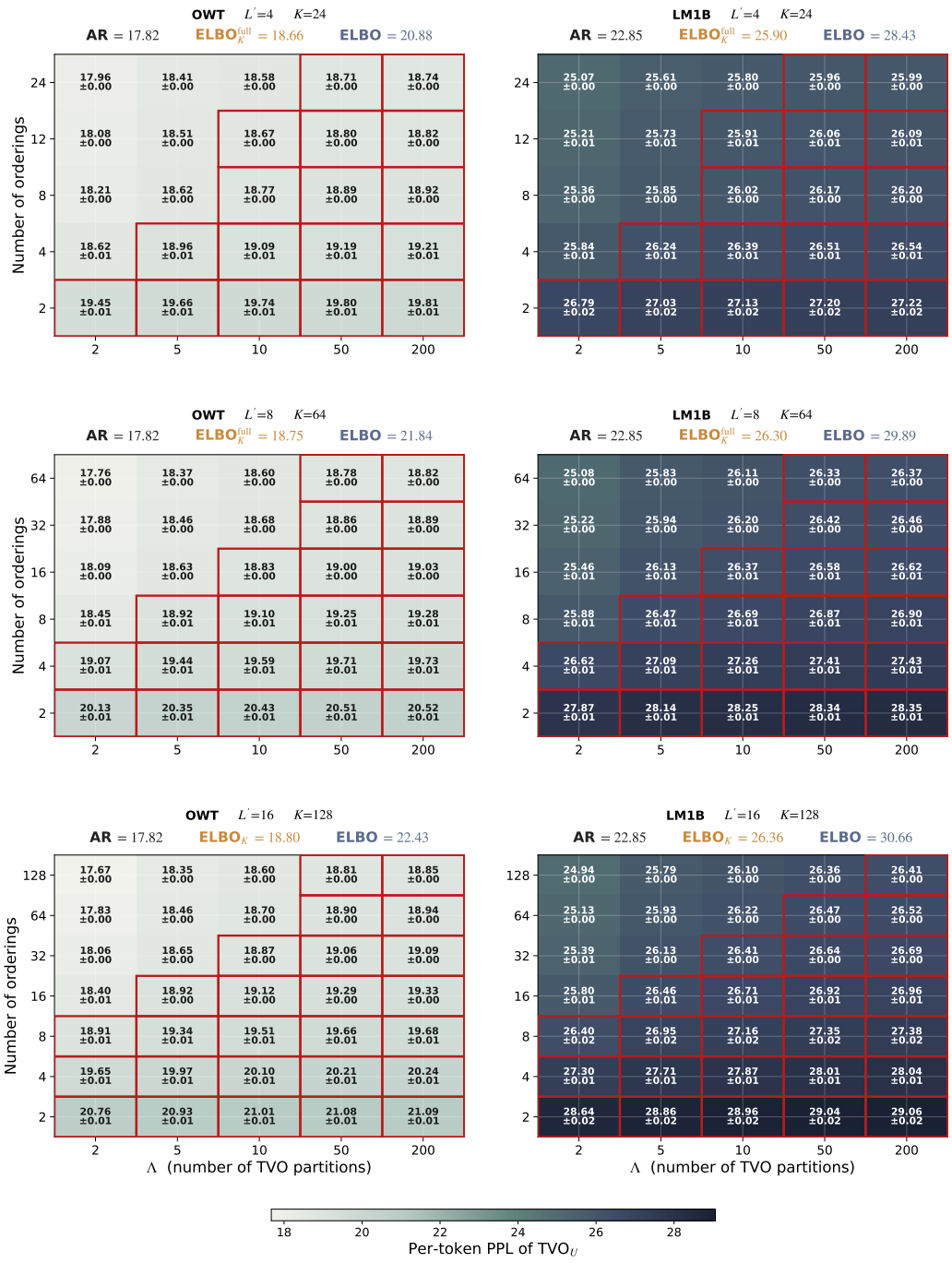


Figure 7: **TVO$_U$ diagnostic on $L' \in \{4, 8, 16\}$.** Per-token PPL of TVO$_U$ as a function of the number of orderings and the number of Riemann partitions L, on OWT (left) and LM1B (right). Red borders mark cells exceeding $\mathrm{ELBO}_K^{\mathrm{full}}$. Only the full-budget row converges correctly as $L \to \infty$.

Source of bias. The first term in (44) is an MC estimate of ELBO_s, which by Jensen lies below $\log p_{\mathrm{model}}(x)$ at any finite s, this gap is part of the construction, since the correction is designed precisely to lift the bound back above $\log p_{\mathrm{model}}(x)$. The second term is meant to do that lifting, but is itself a log of a Monte Carlo mean: letting $r^{(j)} = \sum_{i=1}^{s} p_{\mathrm{model}}(x|\widetilde{\pi}^{(j,i)}) \,/\, \sum_{i=1}^{s} p_{\mathrm{model}}(x|\pi^{(j,i)})$ 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, as shown in Figure 8. [Struski et al., 2023] acknowledge this in their Limitations: the empirical estimator "results in nonrigorous bounds."

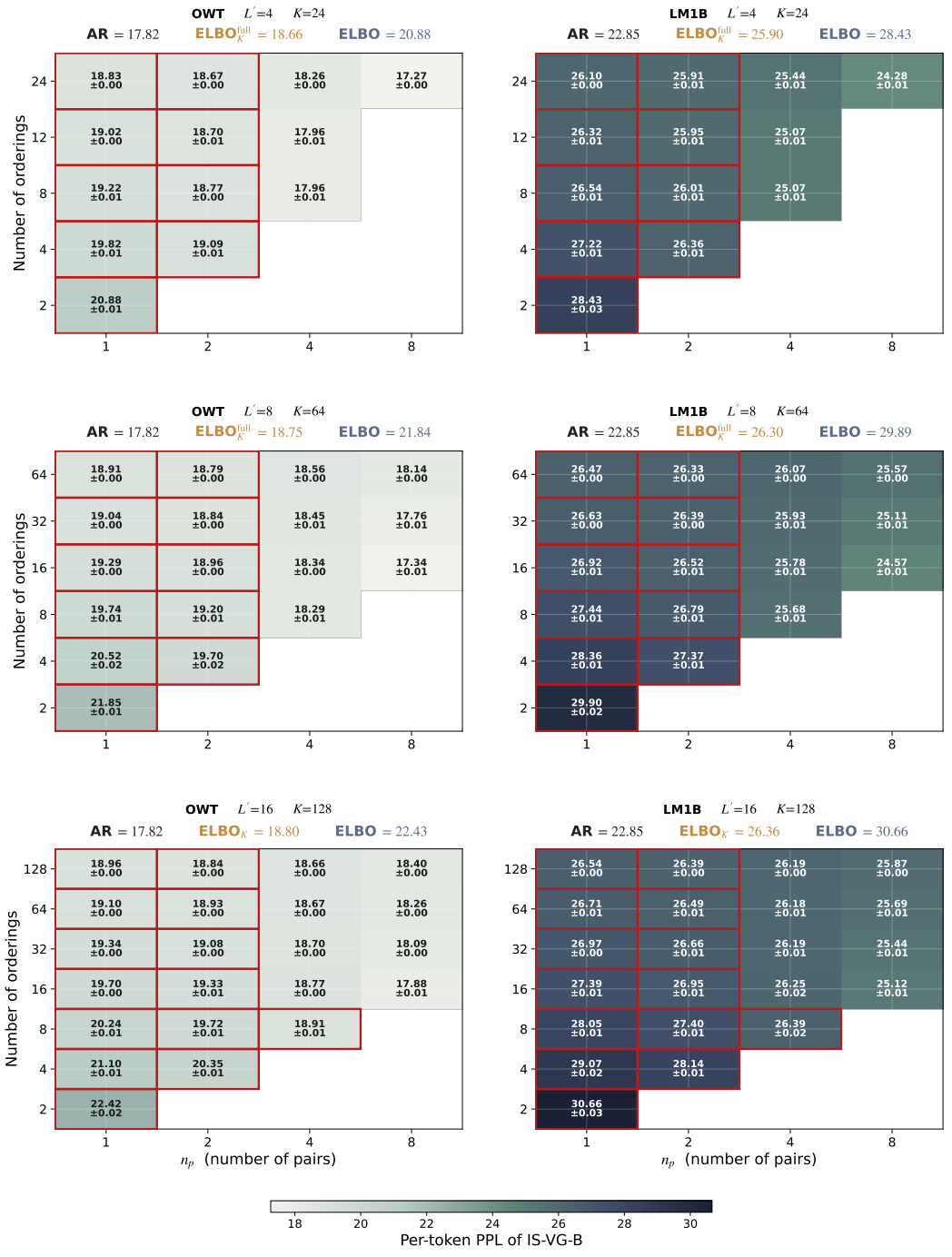


Figure 8: **IS-VG-B diagnostic.** Per-token PPL of IS-VG-B as a function of independent pairs n_{pairs} and within-pair samples k, on OWT (left) and LM1B (right). Red borders mark cells exceeding $\text{ELBO}_K^{\text{full}}$. Small n_{pairs} pushes the estimate above ELBO_K, large n_{pairs} below AR.

E Experimental details

The full implementation, including all .npy with Monte-Carlo orderings and figure-regeneration scripts, is available in the supplementary material (https://github.com/arsenyivanov/tube).

E.1 Models and datasets

Block diffusion and masked diffusion. We evaluate the BMs of Arriola et al. [2025] (https://github.com/kuleshov-group/bd3lms). On OWT we use the public checkpoints `kuleshov-group/bd3lm-owt-block_size{4,8,16}`, the MDLM [Sahoo et al., 2024] checkpoint `kuleshov-group/mdlm-owt`, and the ARM `kuleshov-group/ar-noeos-owt` as the exact ARM baseline (https://huggingface.co/collections/kuleshov-group/bd3-lms-67be95f81b96b15fec50d53f).

Large-scale block diffusion. We evaluate LLaDA2.0-mini [Bie et al., 2025] (https://huggingface.co/inclusionAI/LLaDA2.0-mini), a 16B-parameter MoE block diffusion model with block size 32, and compare it with Ling-mini-2.0 [Team et al., 2025] (https://huggingface.co/inclusionAI/Ling-mini-2.0), the autoregressive model it was converted from.

Uniform diffusion. For Appendix C.3 we use the LM1B checkpoints of UDM, AUDM and MDM released by Gourevitch et al. [2026] (https://github.com/samsongourevitch/rev_udm).

LM1B training details. For LM1B, we follow the original BD3-LM training procedure. We first train an MDM 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 4 is a one-off OWT experiment and is not available for LM1B, so the corresponding bar is missing from the LM1B panel of Figure 5.

E.2 Sampling orderings

Evaluation data. For OWT we use the validation split of BD3-LM (the last 100,000 documents of OpenWebText), tokenized with the GPT-2 tokenizer and wrapped into rows of 1024 tokens without BOS/EOS. For LM1B we use the test split with the BERT tokenizer, wrapped into rows of 128 tokens. From each set we draw a fixed random evaluation subset: 1000 rows for OWT and 5000 rows for LM1B, which gives 256,000 and 160,000 blocks at $L' = 4$. The same rows are used by every model and estimator. Token 0 of every row serves as context and is not scored.

Exact reveal-state tables ($L' \in \{4, 8\}$). Since BD3-LM is not time-conditioned, its prediction for a block depends only on the set S of revealed positions. For every block we run one forward pass per set S and store $\log p(x_i \mid x_S)$ and the confidence $\log \max_v p(v \mid x_S)$ for all $2^{L'}$ sets. One pass scores all blocks of a row at once, because each block reads the previous blocks from the clean copy of the sequence. From these tables the exact likelihood of every regime and policy is computed by dynamic programming over the $2^{L'}$ states, and trajectories are sampled on CPU.

Sampled trajectories ($L' = 16$ and larger models). For $L' = 16$ the table is too large, so we sample trajectories with the model. A trajectory reveals positions block by block, scores each revealed true token from the state before its step, and records $\log p(x \mid \pi)$ for every block. Table 9 represent it.

E.3 Estimator details

All estimators in Tables 2–7 use the same number of trajectories per block, 24, 64 and 128 for $L' = 4, 8, 16$. TUBE splits them into two disjoint halves of $K = M$ trajectories, one for $\widehat{p}_{\text{model},K}$ and one for the self-surrogate ψ_M. CUBO, TVO$_U$ and IS-VG-B use all of them. Every estimate is repeated 10 times, each repetition drawing fresh trajectories per block (from the exact table for $L' \in \{4, 8\}$ and from the 256-trajectory bank for $L' = 16$), and the tables report the mean and the standard deviation over the repetitions. Table 10 summarizes the configuration.

The heatmaps in Appendix D sweep the number of trajectories per block over $\{2, 4, 8, 12, 24\}$, $\{2, 4, \ldots, 64\}$ and $\{2, 4, \ldots, 128\}$ for $L' = 4, 8, 16$, together with $\beta \in \{1, 1.5, 2, 3, 5\}$ for CUBO, $\Lambda \in \{2, 5, 10, 50, 200\}$ for TVO$_U$ and $n_{\text{p}} \in \{1, 2, 4, 8\}$ for IS-VG-B.

Table 9: **Latent samples per model.** Deterministic regimes (AO-ARM with $\gamma = \infty$, NFE $= 1$) use a single trajectory.

Model	Regimes	Latent samples	Rows
BD3-LM, $L' \in \{4, 8\}$	all	table of $2^{L'}$ states	all
BD3-LM, $L' = 16$	AO-ARM, NFE $\in \{1, \ldots, 16\}$, $\gamma \in \{0, 1, \infty\}$	256 per block	all
MDLM (OWT)	AO-ARM, NFE $\in \{128, 512\}$, $\gamma = 0$	1024 per seq.	100
LLaDA2.0-mini	AO-ARM ($\gamma = 0$), confidence threshold	1024 / 512 per NFE	400
UDM, AUDM, MDM	see Appendix C.3	2048 per seq.	100

Table 10: **Estimator configuration in the main tables.** Number of trajectories per block used by each estimator.

Estimator	$L' = 4$	$L' = 8$	$L' = 16$
TUBE ($K + M$)	$12 + 12$	$32 + 32$	$64 + 64$
CUBO$_{\beta=2}$	24	64	128
TVO$_U$ ($\Lambda = 200$ partitions)	24	64	128
IS-VG-B ($n_{\mathrm{p}} = 2$, 4 equal groups)	4×6	4×16	4×32
ELBO (mean of $\log p(x \mid \pi)$)	24	64	128
ELBO$_K$ reference	exact	exact	256
Repetitions for $\pm$ std	10	10	10

In addition, after the computation of TUBE Monte Carlo estimation for each x, we perform clipping, i.e., $\min\{\widehat{\mathrm{TUBE}}_{\psi,K}(x), 0\}$.

E.4 Compute

All evaluations are collected on NVIDIA H200 GPUs (140 GB) with PyTorch 2.7.1 and CUDA 12.6. Table 11 reports the wall-clock times for the validation splits and all estimators.

Table 11: **GPU time of the likelihood banks** (H200, GPU-hours).

Bank	OWT	LM1B
ARM (exact likelihood)	< 0.1	< 0.1
BD3-LM, $L' = 4$ (table of 2^4 states)	0.1	< 0.1
BD3-LM, $L' = 8$ (table of 2^8 states)	1.2	≈ 0.5
BD3-LM, $L' = 16$ (256 trajectories, 16 regimes)	43.7	≈ 10
MDLM (1024 trajectories, 100 sequences)	88.1	–
UDM, AUDM, MDM (2048 samples, 100 sequences)	–	11.2

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract claims a variational upper bound on the log-likelihood with an unbiased Monte Carlo estimator (TUBE, §3) and a two-sided evaluation of MDM likelihoods. The comparison with the ARM is exact for BD3-LM with $L' \in \{4, 8\}$ and a finite-budget estimate for larger blocks and models (§5).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section A discusses the restriction to BMs and the dependence of the tightness of TUBE on the surrogate ψ. The exact likelihood is available only for $L' \in \{4, 8\}$, larger blocks and full-sequence MDMs give wide intervals, and the LLaDA2.0 MDM rows evaluate a confidence-threshold policy rather than its exact sampler.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The TUBE bound (Definition 3.1) follows from the tangent inequality for log shown in §3.1, with the assumptions ($p, \psi > 0$, ψ independent of $\widehat{p}_{\mathrm{model},K}$) stated explicitly. The bias and variance of the estimator are analysed in Proposition 3.1, and the baseline upper-bound estimators (CUBO, TVO, IS-VG-B) are derived in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Models and checkpoints are listed in Appendix E.1, evaluation data and latent samples in Appendix E.2, and estimator settings in Appendix E.3. The released code prepares the evaluation data, collects the likelihood banks and computes every table and figure from them.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is available at `https://github.com/arsenyivanov/tube` with the environment and instructions to reproduce every result. OWT and LM1B are public. The OWT checkpoints of Arriola et al. [2025], LLaDA2.0-mini, Ling-mini-2.0 and the checkpoints of Gourevitch et al. [2026] are public (Appendix E.1). The LM1B BD3-LM and ARM checkpoints are trained with the default protocol of the BD3-LM repository.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Setup paragraphs in §5.1.1, §5.1.2 and §5.2 describe the evaluation protocol. Appendix E.1, Appendix E.2 and Appendix E.3 give the checkpoints, the evaluation data and latent samples, and the per-estimator settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Tables 2 and 7 and Figure 4 report the mean and one standard deviation over 10 repetitions, each drawing fresh trajectories per block. For the full-sequence and uniform-diffusion experiments, the standard deviation is over 10 random splits of one bank of samples, and the MDLM table reports single estimates from one bank.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix E.4 reports the hardware (NVIDIA H200, 140 GB), the software environment and the GPU time of every likelihood sampling (Table 11). All estimators are computed on CPU from the banks. Preliminary experiments are not included.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper is methodological, evaluating likelihood bounds for existing pre-trained language models on public datasets. It conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section A discusses positive applications. The work is foundational, and any negative impacts would be those general to language modeling rather than specific to likelihood estimation.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The only newly trained models are small (about 170M parameters) MDM, BD3-LM and ARM models trained on the public LM1B benchmark for evaluation. Their capabilities are comparable to existing public LM1B models and pose no specific misuse risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use the code and OWT checkpoints of BD3-LM and MDLM [Arriola et al., 2025, Sahoo et al., 2024] (Apache-2.0), the code and checkpoints of Gourevitch et al. [2026] (MIT), LLaDA2.0-mini [Bie et al., 2025] (Apache-2.0), Ling-mini-2.0 [**?**] (MIT), and the OWT [Gokaslan and Cohen, 2019] and LM1B [Chelba et al., 2014] datasets under their public terms. Vendored code keeps its original license file.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The TUBE evaluation code is released at `https://github.com/arsenyivanov/tube` with a README, a requirements file and instructions for every step.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human subjects research.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: All research ideas, methodology, derivations and experimental analysis were developed by the authors. LLMs were used as writing and coding assistants and are not part of the core methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.