DUEL: Exact Likelihood for Masked Diffusion via Deterministic Unmasking

Gilad Turok [1] **Yair Schiff** [1] **Chris De Sa** [1] **Volodymyr Kuleshov** [1]

Abstract

Masked diffusion models (MDMs) generate text by iteratively selecting positions to unmask and then predicting tokens at those positions. Yet MDMs lack proper likelihood evaluation: the evidence lower bound (ELBO) is not only a loose bound on log-likelihood, but, as we show, is also computed under the training distribution rather than the test-time distribution. We resolve this within our DUEL framework, which unifies leading MDM sampling strategies that employ *deterministic* position selection. We prove that DUEL samplers admit **exact likelihood computation under the test-time distribution**—giving MDMs *proper* likelihood, and hence proper perplexity, for the first time. This proper perplexity is the natural analogue of autoregressive perplexity and lets us revisit key questions about MDMs. Under proper evaluation, **MDMs are substantially better than previously thought**: the MDM–autoregressive perplexity gap shrinks by up to 32% on in-domain data and 82% on zero-shot benchmarks. DUEL enables the first principled comparison of fast, parallel samplers across compute budgets—an analysis impossible with the ELBO and unreliable with generative perplexity—identifying a strong default method. Finally, an oracle ordering (with ground-truth access) improves MDM perplexity well beyond ARM — 36.47 vs. 52.11 on AG News — revealing substantial room for improved inference-time orderings.

1. Introduction

Diffusion models have achieved remarkable success in continuous domains such as images and video (Ho et al., 2020; Song et al., 2020). Recently, masked diffusion models (MDMs) have extended this success to discrete domains (Austin et al., 2021; Sahoo et al., 2024; Shi et al., 2024; Lou et al., 2024), and scaled to billions of parameters with strong results across language benchmarks (Nie et al., 2025; Bie et al., 2025; Inception Labs et al., 2025; Song et al., 2025).

Under the any-order autoregressive (AO-ARM) interpretation (Hoogeboom et al., 2021a; Ou et al., 2024; Kim et al., 2025a), MDM generation decomposes into two components. The unmasking policy π performs *position selection*: it determines which masked positions to reveal. The denoising distribution p_θ, parameterized by a neural network x_θ, performs *token prediction*: given a partially masked sequence, it outputs a probability distribution over tokens at each masked position. Together, these define a complete generative procedure: the policy selects positions, tokens are sampled from the denoiser's predictions, and this repeats until the full sequence is generated. Figure 1 illustrates one such step.

This decomposition has an important consequence: for a fixed pretrained network x_θ, MDMs define not one generative model but a *family* of models indexed by the choice of policy π. Each policy produces different samples, implicitly defining its own marginal distribution over text.

Notably, many of the best-performing MDM samplers use *deterministic* unmasking policies. Methods like greedy confidence (Nie et al., 2025), probability margin (Kim et al., 2025a), confidence threshold (Wu et al., 2025), and KLASS (Kim et al., 2025b) each define an *unmasking rule* F—a deterministic function mapping partially masked sequences to positions—based on heuristics applied to the network output. Each rule F induces a deterministic unmasking policy π^F with no randomness in position selection. Crucially, the resulting permutation is *data-dependent*: for a given rule F, the ordering induced by greedy confidence (say) differs across inputs $\mathbf{x}$ because confidence depends on the model's predictions for that specific sequence.

We introduce the **D**eterministic **U**nmasking **E**xact **L**ikelihood (DUEL ⚔) framework to formalize and exploit this

Accepted to FoGen 2026: Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning, an ICML 2026 workshop (non-archival).

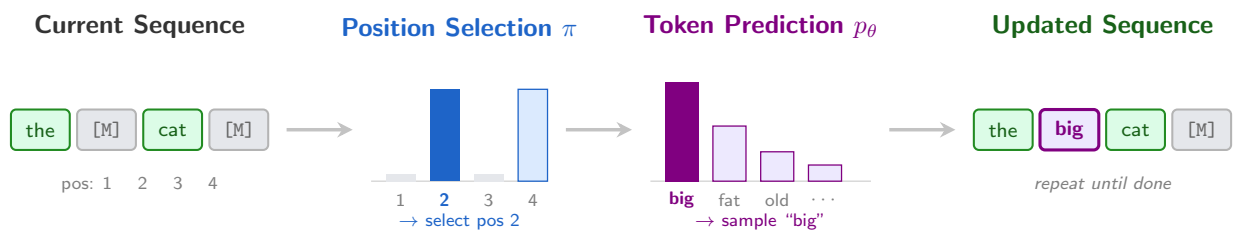

Figure 1. **One step of MDM generation.** Given a partially masked sequence, the unmasking policy π performs *position selection* (choosing position 2), and the denoising network p_θ performs *token prediction* at that position (sampling "big"). This repeats until no masked positions remain.

structure. A DUEL sampler is a pair (x_θ, F) of a denoiser network and a deterministic unmasking rule. We prove that the distribution induced by a DUEL sampler admits exact likelihood computation via a simple algorithm. The key insight: likelihood normally requires marginalizing over all unmasking orderings—a super-exponential sum with $> L!$ terms—but a deterministic policy collapses this to a single term that mirrors the generation procedure. DUEL is a *plug-and-play* evaluation methodology: it computes proper likelihood for the *standard* MDM under its actual test-time policy, with no retraining or architectural changes. Crucially, DUEL corrects how MDMs are *measured*, not the models themselves: every gap reduction we report comes from proper evaluation of an unchanged network. Training MDMs under non-uniform orderings is non-trivial—it removes the uniform-masking decomposition that makes single-pass training efficient, and recent work that pursues this direction (Kim et al., 2026; Peng et al., 2025b) requires substantially more expensive training procedures.

Perplexity is the standard metric for evaluating language models, measuring the exact probability assigned to text under the generative procedure. However, MDMs currently lack an equivalent. The evidence lower bound (ELBO) is both loose and measures the wrong distribution: it bounds likelihood under *uniform random* position selection π^{unif} used for training, not the deterministic policies π^F often used at test time. Generative perplexity requires an external reference model that is biased and expensive, and ignores sample diversity—a model that repeats one good sentence scores well despite mode collapse. DUEL resolves this: it computes exact likelihood under the test-time distribution induced by π^F, giving MDMs a proper perplexity metric for the first time.

In summary, our contributions are as follows:

- **The DUEL framework and proper MDM perplexity.** Pairing a pretrained denoiser x_θ with a deterministic unmasking rule F admits exact likelihood computation (Theorem 4.2), unifying leading MDM sampling strategies under a common formalism. The resulting likelihood is the natural analogue of autoregressive perplexity, evaluating the test-time distribution directly.

- **Reassessing the MDM–ARM perplexity gap.** Across multiple models and benchmarks, DUEL closes up to 32% of the gap in-domain and 82% zero-shot under proper evaluation, showing MDMs are substantially closer to ARMs than the ELBO suggests.

- **Principled sampler analysis.** DUEL enables comparisons impossible with the ELBO and unreliable with generative perplexity: it reliably ranks fast parallel samplers across compute budgets, identifying probability margin (Kim et al., 2025a) as a strong default, and oracle search over orderings reveals substantial inference-time head-room beyond what standard policies achieve.

2. Background

We briefly review masked diffusion models (MDMs) for language generation (Austin et al., 2021; Sahoo et al., 2024; Shi et al., 2024; Lou et al., 2024; Gat et al., 2024; Nie et al., 2025). See Appendix B.1 for extended MDM details.

Forward Process. MDMs define a forward process that progressively masks a clean sequence $\mathbf{x} = (x^{(1)}, \ldots, x^{(L)}) \in \mathcal{V}^L$ over time $t \in [0, 1]$, where $\mathcal{V} = \{1, \ldots, V\}$ is a vocabulary of V tokens. Each position independently either retains

Unmask Rule F	**Selected Positions** $F(\mathbf{z})$	**Intuition**
Left-to-Right	$\{k \text{ smallest } \ell : z_\ell = \mathtt{m}\}$	autoregressive order
Greedy Conf. (Nie et al., 2025)	$\arg\max_{\lvert\mathcal{I}\rvert=k} \sum_{\ell\in\mathcal{I}} P_\ell^{(1)}$	most certain first
Prob. Margin (Kim et al., 2025a)	$\arg\max_{\lvert\mathcal{I}\rvert=k} \sum_{\ell\in\mathcal{I}} (P_\ell^{(1)} - P_\ell^{(2)})$	largest top-1 vs. top-2 gap
Conf. Thresh. (Wu et al., 2025)	$\{\ell \in \mathcal{M} : P_\ell^{(1)} \geq \mu\}$	adaptive parallelism
KLASS (Kim et al., 2025b)	$\{\ell \in \mathcal{M} : P_\ell^{(1)} \geq \mu,\ \mathrm{KL}(\hat{P}_\ell \Vert P_\ell) \leq \nu\}$	confident and stable

Table 1. Unmasking rules defining deterministic policies. $\mathcal{M} = \mathcal{M}(\mathbf{z})$ are masked positions; $\mathbf{P} = \mathrm{softmax}(x_\theta(\mathbf{z}))$ is the token probability matrix; $\hat{P}_\ell$ is the previous step's prediction.

its value or transitions to a special mask token $\mathtt{m}$:

$$q_{t|0}(z^{(\ell)} \mid x^{(\ell)}) = \mathrm{Cat}(\alpha_t\, \mathbf{e}_{x^{(\ell)}} + (1-\alpha_t)\, \mathbf{e}_{\mathtt{m}}),$$

where $\mathbf{e}_v$ is the one-hot vector for token v. The noise schedule satisfies $\alpha_0 = 1$ (clean) and $\alpha_1 = 0$ (fully masked). The mask token is *absorbing*: once masked, a position remains masked, yielding partially masked sequences $\mathbf{z} \in \mathcal{V} \cup \{\mathtt{m}\}$.

Reverse Process. Generation reverses this corruption: starting from a fully masked sequence, the model iteratively predicts and reveals tokens at masked positions $\mathcal{M}(\mathbf{z}) = \{\ell : z_\ell = \mathtt{m}\}$. A *denoising network* x_θ takes a partially masked sequence $\mathbf{z}$ and outputs logits predicting clean tokens at each position. We define the token probability matrix $\mathbf{P} = \mathrm{softmax}(x_\theta(\mathbf{z})) \in \Delta_V^L$, where each row P_ℓ is a distribution over tokens at position ℓ. The reverse transition substitutes these predictions for the unknown ground truth: $p_\theta(z_s^{(\ell)} \mid \mathbf{z}_t) = q_{s|t}(z_s^{(\ell)} \mid \mathbf{z}_t, \mathbf{x} \leftarrow \mathbf{P})$.

Training Objective. The data likelihood marginalizes over latent masked trajectories, $p_\theta(\mathbf{x}) = \sum_{\mathbf{z}_{0:T}} p_\theta(\mathbf{x}, \mathbf{z}_{0:T})$, which is intractable. Following Sahoo et al. (2024), we obtain a tractable ELBO that reduces to the expected negative log-likelihood of true tokens at masked positions:

$$\mathcal{L}_{\mathrm{ELBO}}(\theta) = \mathop{\mathbb{E}}_{t,\mathbf{z}_t}\Big[w_t \sum_{\ell\in\mathcal{M}(\mathbf{z}_t)} \log p_\theta(x^{(\ell)} \mid \mathbf{z}_t)\Big], \tag{1}$$

where $t \sim \mathrm{Unif}[0,1]$ is time, $\mathbf{z}_t \sim q_{t|0}(\cdot \mid \mathbf{x})$ is a random masking of $\mathbf{x}$, and $w_t = \alpha_t'/(1-\alpha_t)$ is the schedule weight.

AO-ARM Interpretation. MDMs admit an equivalent interpretation as any-order autoregressive models (AO-ARMs) (Uria et al., 2014; Yang et al., 2019; Hoogeboom et al., 2021a), decomposing generation into *position selection* via an unmasking policy π_θ and *token prediction* via a denoising distribution p_θ. Recent work formally establishes that the MDM training loss of Equation (1) is equivalent to the AO-ARM ELBO of Equation (5): both reduce to the expected negative log-likelihood of true tokens at masked positions, averaged over random orderings (Zheng et al., 2024; Ou et al., 2024; Kim et al., 2025a). We adopt and extend this perspective in Section 4.

3. The DUEL Framework

We introduce DUEL samplers—pairs of a denoiser network and an unmasking rule—that define implicit generative procedures over sequences. Section 3.1 defines these components and shows how they enable sampling. Section 3.2 presents common instantiations of unmasking rules.

3.1. DUEL Samplers

A DUEL sampler combines two components: a denoising network that predicts tokens, and an unmasking rule that selects positions. In practice, this is just a pretrained MDM together with a choice of how to sample from it.

Definition 3.1 (Unmasking Rule)**.** An *unmasking rule* F is a deterministic function mapping a partially-revealed sequence $\mathbf{z}$ to a non-empty subset of masked positions:

$$\emptyset \neq F(\mathbf{z}) \subseteq \mathcal{M}(\mathbf{z}).$$

Since the token probabilities $\mathbf{P} = \mathrm{softmax}(x_\theta(\mathbf{z}))$ are deterministic functions of $\mathbf{z}$, rules depending on $\mathbf{P}$ are valid. The non-emptiness constraint ensures generation progresses at each step.[1] Equivalently, we may write $F(\mathbf{x}^{\sigma_{<t}})$ since $\mathbf{z}$ and the revealed-token pair $(\mathbf{x}^{\sigma_{<t}}, \sigma_{<t})$ carry the same information.[2]

Definition 3.2 (DUEL Sampler)**.** A DUEL *sampler* is a pair (x_θ, F) of a pretrained denoising network x_θ outputting token probabilities $\mathbf{P} = \mathrm{softmax}(x_\theta(\mathbf{z}))$, and an unmasking rule F selecting which positions to reveal. Each rule F induces a deterministic unmasking policy π^F, formalized in Section 4.1, that places all probability mass on the positions F selects.

A DUEL sampler defines an *implicit* generative procedure: we can draw samples without ever writing down the induced distribution in closed form. Starting from the fully masked sequence, we repeatedly: (1) compute token distributions via the denoising network, (2) select positions deterministically via F, and (3) sample and reveal tokens at those positions. Generation terminates when all positions are unmasked. Algorithm 1 formalizes this procedure.

3.2. Instantiations

The DUEL framework unifies leading MDM generation strategies: each corresponds to an unmasking rule F inducing a deterministic policy π^F with exact likelihood via Theorem 4.2. Table 1 catalogues the five rules we consider, with $P_\ell^{(1)}$ and $P_\ell^{(2)}$ the highest and second-highest token probabilities at position ℓ: *left-to-right* reveals the k leftmost masked indices (autoregressive when $k = 1$); *greedy confidence* (Nie et al., 2025) commits to the k most-confident positions; *probability margin* (Kim et al., 2025a) prefers tokens where the top-1 vs. top-2 gap is largest; *confidence threshold* (Wu et al., 2025) reveals every position with $P_\ell^{(1)} \geq \mu$ (adaptive NFE; we tune μ to match each target budget); and *KLASS* (Kim et al., 2025b) adds a stability check via the KL between consecutive predictions. These rules compose with hybrid MDM strategies: in *block diffusion* (Arriola et al., 2025a), F applies within blocks of size L' that are processed autoregressively; in *entropy-bounded sampling* (Ben-Hamu et al., 2025), the entropy-based selection is itself a deterministic function of $\mathbf{z}$.

4. Exact Likelihood Computation

A DUEL sampler (x_θ, F) generates samples via Algorithm 1, but what distribution does it sample from? Section 4.1 develops the formulation needed to make this distribution explicit, and Section 4.2 proves that deterministic unmasking enables its exact likelihood computation.

4.1. Any-Order Autoregressive Formulation

Building on the AO-ARM interpretation of MDMs (Section 2), we introduce *ordered partitions* for parallel unmasking and formalize *deterministic unmasking policies* (extended treatment in Appendix B.2).

Ordered Partitions. We formalize the progression of unmasking steps of target sequence $\mathbf{x} \in \mathcal{V}^L$.

Definition 4.1 (Ordered Partition)**.** An *ordered partition* of $[L] = \{1, \ldots, L\}$ is a tuple $\sigma = (\sigma_1, \ldots, \sigma_T)$ of non-empty subsets with

$$\bigcup_{t=1}^{T} \sigma_t = [L] \qquad \text{and} \qquad \sigma_t \cap \sigma_{t'} = \emptyset$$

for all $t \neq t'$. Each σ_t is the set of positions unmasked at step t, and may depend on $\mathbf{x}$.

Ordered partitions capture both sequential and parallel unmasking—a distinguishing feature of MDMs over ARMs. When each part is a singleton ($|\sigma_t| = 1$), we recover *sequential unmasking* with $T = L$ steps; otherwise we have *parallel unmasking* with $T < L$ steps. We write $\sigma_{<t} \triangleq \sigma_1 \cup \cdots \cup \sigma_{t-1}$ for positions revealed before step t, with $\mathbf{x}^{\sigma_{<t}}$ and $\mathbf{x}^{\sigma_t}$ the corresponding clean-token tuples.

[1] *Tie-breaking.* When the score underlying F admits ties (e.g., equal top-1 confidence at two positions), we resolve them by the lowest position index. This preserves determinism and is what our implementation uses throughout.

[2] History-augmented rules (e.g., KLASS) extend F to depend on auxiliary state that is itself a deterministic function of the trajectory; determinism, and hence Theorem 4.2, is preserved.

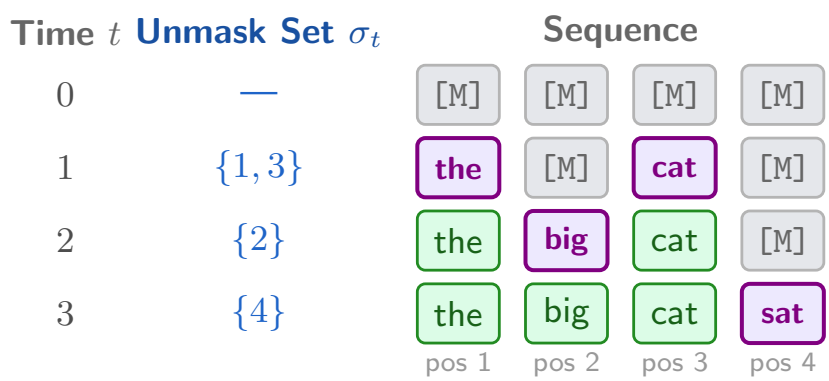

Figure 2. Ordered partition $\sigma = (\{1, 3\}, \{2\}, \{4\})$: parallel reveal at step 1, sequential at steps 2–3.

Joint Factorization. The joint probability of generating sequence $\mathbf{x}$ via ordered partition σ factorizes into position selection and token prediction at each step (derived in Appendix D.1):

$$p_\theta(\mathbf{x}, \sigma) = \prod_{t=1}^{T} \underbrace{\pi_\theta(\sigma_t \mid \mathbf{x}^{\sigma_{<t}})}_{\text{position selection}} \cdot \underbrace{p_\theta(\mathbf{x}^{\sigma_t} \mid \mathbf{x}^{\sigma_{<t}})}_{\text{token prediction}}, \tag{2}$$

where the *unmasking policy* $\pi_\theta(\sigma_t \mid \mathbf{x}^{\sigma_{<t}})$ selects which masked positions to reveal conditioned on the revealed tokens, and the *denoising distribution* $p_\theta(\mathbf{x}^{\sigma_t} \mid \mathbf{x}^{\sigma_{<t}})$ gives the probability of tokens at those positions. Under conditional independence (Assumption D.1), the denoising distribution factorizes over positions:

$$p_\theta(\mathbf{x}^{\sigma_t} \mid \mathbf{x}^{\sigma_{<t}}) = \prod_{\ell \in \sigma_t} p_\theta(x^{(\ell)} \mid \mathbf{x}^{\sigma_{<t}}),$$

where $p_\theta(x^{(\ell)} \mid \mathbf{x}^{\sigma_{<t}}) \triangleq P_\ell[x^{(\ell)}]$ with token probabilities $\mathbf{P} = \mathrm{softmax}(x_\theta(\cdot))$ for token $x^{(\ell)}$ at position ℓ.

Deterministic Unmasking Policies. An unmasking rule F induces a *deterministic unmasking policy* π_θ^F that places all probability mass on the positions returned by F:

$$\pi_\theta^F(\sigma_t \mid \mathbf{x}^{\sigma_{<t}}) = \begin{cases} 1 & \text{if } \sigma_t = F(\mathbf{x}^{\sigma_{<t}}) \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Here σ_t ranges over all possible subsets of masked positions, and π_θ^F is a Dirac delta concentrating on the unique subset $F(\mathbf{x}^{\sigma_{<t}})$. Since F is a function, position selection is entirely deterministic given the revealed tokens. The dependence on θ is indirect: F selects positions based on the network's token probabilities $\mathbf{P} = \mathrm{softmax}(x_\theta(\cdot))$.

Induced Distribution. An AO-ARM defines the data likelihood by marginalizing over ordered partitions: $p_\theta(\mathbf{x}) = \sum_\sigma p_\theta(\mathbf{x}, \sigma)$. Specializing Equation (2) to the deterministic policy π_θ^F yields the distribution induced by a DUEL sampler:

$$p_\theta^{\pi^F}(\mathbf{x}) = \sum_\sigma \prod_{t=1}^{T} \pi_\theta^F(\sigma_t \mid \mathbf{x}^{\sigma_{<t}}) \cdot p_\theta(\mathbf{x}^{\sigma_t} \mid \mathbf{x}^{\sigma_{<t}}). \tag{4}$$

This is the distribution that Algorithm 1 samples from. Crucially, this equation is only made explicit in the *AO-ARM* formulation of MDMs, and not in the standard *latent-variable* formulation.

Computing Equation (4) directly appears intractable: the sum ranges over all ordered partitions σ, each corresponding to a different generation trajectory. The number of such partitions is super-exponential in the sequence length L. Since the denoiser conditions on different contexts under different orderings, each term in the sum requires separate neural network evaluation. Section 4.2 shows that determinism makes this tractable.

Connection to MDM Training. The MDM–AO-ARM connection becomes precise through the unmasking policy. Under *uniform random* position selection with sequential unmasking, for a partially masked sequence $\mathbf{z}$,

$$\pi^{\text{unif}}(\sigma_t \mid \mathbf{z}) = \frac{1}{|\mathcal{M}(\mathbf{z})|},$$

each masked position is equally likely to be unmasked next. This policy arises naturally from ancestral sampling of the MDM reverse process: since the forward process independently masks each position at a random time, reversing this process treats all masked positions symmetrically.

Under uniform position selection π^{unif} with *sequential* unmasking (one token per step), the proposal $q(\sigma) = 1/L!$ over permutations yields a tractable ELBO (Propositions D.3 and D.4):

$$\mathcal{L}_{\text{ELBO}}(\theta) = \mathop{\mathbb{E}}_{q(\sigma)}\left[\sum_{t=1}^{T}\sum_{\ell\in\sigma_t} -\log p_\theta(x^{(\ell)} \mid \mathbf{x}^{\sigma_{<t}})\right]. \tag{5}$$

Ou et al. (2024); Kim et al. (2025a) show this is equivalent to the standard MDM loss of Equation (1), formally establishing MDMs as AO-ARMs trained under uniform random ordering. The ELBO serves as a training objective and is widely used for MDM evaluation; however, it does not yield proper likelihood under the test-time distribution, and hence cannot provide proper perplexity.

4.2. Theoretical Result

Equation (4) sums over super-exponentially many ordered partitions and appears intractable; we show that determinism collapses this sum to a single term. Given any partially masked sequence $\mathbf{z}$, the rule F outputs a unique set of positions, and revealed tokens remain fixed thereafter, so exactly one ordered partition σ^* is consistent with π^F given $\mathbf{x}$. All other partitions receive zero probability, leaving a single computable term.

Theorem 4.2 (DUEL Exact Likelihood)**.** *Let (x_θ, F) be a* DUEL *sampler. Assuming conditional independence in parallel unmasking, Algorithm 2 computes $\log p_\theta^{\pi^F}(\mathbf{x})$ exactly, implementing:*

$$\log p_\theta^{\pi^F}(\mathbf{x}) = \sum_{t=1}^{T}\sum_{\ell\in\sigma_t^*} \log p_\theta(x^{(\ell)} \mid \mathbf{x}^{\sigma^*_{<t}}), \tag{6}$$

where $\sigma^* = (\sigma_1^*, \ldots, \sigma_T^*)$ is the unique ordered partition satisfying $\sigma_t^* = F(\mathbf{x}^{\sigma^*_{<t}})$ at each step. (Proof in Theorem D.5.)

Likelihood Follows Generation. This is the crux of DUEL: *the likelihood computation follows the same path as generation*, simulating the unmasking process but revealing true tokens instead of sampling them. Algorithms 1 and 2 differ by one line and fuse into a single loop that emits a sample and its log-probability simultaneously.

5. Evaluation via DUEL

Perplexity is the standard intrinsic metric for autoregressive language models, but MDMs lack an analogue: their two existing metrics either bound the wrong quantity (ELBO) or rely on an external reference model (generative perplexity). DUEL closes this gap by computing the exact likelihood of $p_\theta^{\pi^F}(\mathbf{x})$—the distribution induced by a deterministic test-time policy (Equation (4))—giving MDMs perplexity on the same probabilistic footing as autoregressive evaluation.

ELBO. The ELBO bounds the wrong distribution: $\log p_\theta^{\pi^{\text{unif}}}(\mathbf{x})$ under *uniform* position selection, not the deterministic policy π^F used at test time. The bound is also loose: the variational gap $\log p_\theta^{\pi^{\text{unif}}}(\mathbf{x}) - (-\mathcal{L}_{\text{ELBO}})$ can be substantial, systematically underestimating model quality even on the ELBO's own terms.

Generative Perplexity. Generative perplexity instead samples from the model and scores those samples under an external reference, typically GPT-2 (Radford et al., 2019), so scores reflect the reference's biases as much as the MDM's quality (Wang et al., 2024). Pynadath et al. (2026) additionally show that it captures only one of two terms in the KL divergence to the reference distribution—the other being entropy—so a model that collapses onto a single high-quality phrase still scores well. Post-hoc diagnostics like entropy and MAUVE (Pillutla et al., 2021) treat symptoms rather than the underlying mismatch.

Table 2. In-domain test perplexity (↓) on OWT and LM1B. **Bold**: best MDM.

(a) OWT. ARM baseline: 17.54.

Model	ELBO	DUEL	Gap Closed
SEDD	≤24.10	**22.58**	23.2%
MDLM	≤22.98	**21.86**	20.6%
BD3-LM L'=4	≤20.73	**19.73**	31.3%
BD3-LM L'=8	≤21.68	**20.37**	31.6%
BD3-LM L'=16	≤22.27	**20.76**	31.9%

(b) LM1B. ARM baseline: 22.83.

Model	ELBO	DUEL	Gap Closed
SEDD	≤32.68	**31.51**	11.9%
MDLM	≤31.78	**29.03**	30.7%
BD3-LM L'=4	≤28.23	**26.97**	23.3%
BD3-LM L'=8	≤29.83	**27.85**	28.3%
BD3-LM L'=16	≤30.60	**28.39**	28.4%

Table 3. Zero-shot gap closed (%); trained on OWT, BD3-LM $L' = 4$. Full PPL in Table 7.

Dataset	SEDD	MDLM	BD3-LM
PTB	31.3%	34.3%	81.8%
Wikitext	40.8%	28.5%	31.4%
LM1B	22.1%	29.0%	29.8%
AG News	25.7%	27.8%	51.7%
Average	30.0%	29.9%	48.7%

Table 4. Zero-shot perplexity (↓), 8B models. **Bold**: better of ELBO/DUEL.

Model	Method	Wikitext	Lambada	AG News
Llama3	Exact	7.94	32.40	41.29
LLaDA	ELBO	≤15.29	≤39.04	≤85.17
	DUEL	**14.50**	**36.00**	**78.91**

What DUEL Enables. DUEL is the first MDM metric sensitive to both the denoiser x_θ and the unmasking policy F without reference-model bias, unlocking two analyses: comparing MDMs and AR models on their respective test-time distributions (Section 6.1), and isolating the contribution of F at fixed x_θ to compare unmasking rules (Section 6.2).

Cost and Caveats. DUEL likelihood costs no more than a single generation: Algorithms 1 and 2 differ by one line and fuse into a single forward-pass loop that emits both a sample and its log-probability. This makes DUEL strictly cheaper than generative perplexity, which generates samples *and then* runs a separate reference-model pass. DUEL is more expensive than the ELBO—which can be Monte-Carlo estimated from a handful of forward passes—but trades that extra cost for exactness on the right distribution rather than a loose bound on the wrong one. Like AR perplexity, DUEL cannot reflect token-sampling heuristics (nucleus, top-k) or downstream task quality.

6. Experiments

We demonstrate DUEL through two categories of experiments. First, we quantify how much of the perplexity gap between MDMs and ARMs is due to evaluation methodology (Section 6.1). Second, we use DUEL to compare sampling strategies—an analysis impossible with the ELBO and unreliable with generative perplexity (Section 6.2).

6.1. Perplexity Gap

Setup. MDMs have consistently lagged behind ARMs in perplexity benchmarks (Nie et al., 2024). We evaluate whether this gap is inflated by evaluation methodology. We compare an ARM baseline, SEDD (Lou et al., 2024), MDLM (Sahoo et al., 2024), and BD3-LM (Arriola et al., 2025a) with block sizes $L' \in \{4, 8, 16\}$, all trained with comparable model sizes, compute budgets, and data. All MDMs use greedy confidence unmasking restricted to a given block size. We evaluate *in-domain* on OWT and LM1B (Tables 2a and 2b), *zero-shot* on PTB, Wikitext, LM1B, Lambada, and AG News (Table 3; full results in Table 7), and at *large scale* with LLaDA-8B (Nie et al., 2025) and Llama3-8B (Grattafiori et al., 2024) on Wikitext, Lambada, and AG News (Table 4). Details in Appendix E.1.

Metric. We define the *perplexity gap* $\Delta = \mathrm{PPL}_{\mathrm{MDM}} - \mathrm{PPL}_{\mathrm{ARM}}$. Δ_{ELBO} inflates the gap on two axes—a loose bound and the wrong policy (π^{unif} instead of π^F)—both of which Δ_{DUEL} corrects. The *gap closed* reports the fraction of Δ_{ELBO} this removes:

$$\text{Gap Closed (\%)} = \frac{\Delta_{\mathrm{ELBO}} - \Delta_{\mathrm{DUEL}}}{\Delta_{\mathrm{ELBO}}} \times 100. \tag{7}$$

Table 5. DUEL perplexity (↓) by unmasking rule on OWT. BD3-LM (L'=16); ELBO ≤23.52. **Bold**: best per NFE.

Unmask Rule F	**128**	**256**	**512**	**1024**
Left-to-Right	240.27	109.00	44.51	**21.46**
Greedy Conf.	165.48	67.15	34.74	22.04
Prob. Margin	**141.31**	**57.66**	**32.33**	22.35
Conf. Thresh.	227.77	116.95	43.45	22.04

Results. In-domain (Table 2), DUEL closes 21–32% of the gap on OWT and 12–31% on LM1B. Zero-shot (Table 3; full perplexities in Table 7), it closes 30% on average, with BD3-LM reaching 82% on PTB. At 8B scale (Table 4), DUEL consistently reduces LLaDA perplexity across Wikitext, Lambada, and AG News, confirming exact evaluation applies at scale; since LLaDA and Llama3 differ in training data and compute, we report raw perplexity only.

Takeaway. Proper perplexity evaluation via DUEL *substantially* closes the MDM–ARM perplexity gap. The gains come entirely from evaluation, not model changes: the ELBO averages over all orderings equally, including poor ones; DUEL evaluates under the deterministic policies that avoid such orderings.

6.2. Comparing Sampling Strategies

DUEL enables controlled sampler comparison (varying F at fixed x_θ) via exact likelihood—impossible with the ELBO, which is invariant to F, and unreliable with reference-biased generative perplexity. We compare fast parallel samplers (Section 6.2.1) and search over orderings for an oracle baseline (Section 6.2.2).

6.2.1. Comparing Fast Samplers

The number of function evaluations (NFE) trades quality for the parallelism that distinguishes MDMs from ARMs. We compare four unmasking rules—left-to-right, greedy confidence, probability margin, and confidence threshold—across NFE budgets along four metrics: DUEL perplexity, generative perplexity (via GPT-2), entropy, and MAUVE.

Setup. We evaluate BD3-LM ($L' = 16$) trained on OWT, varying NFE $\in \{128, 256, 512, 1024\}$. The ELBO (≤23.52) is invariant to unmasking rules and NFE.

Results. Table 5 reports DUEL perplexity. At 1024 NFE, all rules converge to similar perplexity (≈21–22). At lower NFE, probability margin consistently achieves the lowest perplexity, followed closely by greedy confidence. At 128 NFE, probability margin (141.31) outperforms greedy confidence (165.48), the next-best rule.

Takeaway. DUEL yields *consistent* sampler rankings across NFE budgets, identifying probability margin as a strong default under compute constraints. Generative perplexity inverts this—rewarding the low-entropy degenerate text that left-to-right produces at low NFE—while MAUVE saturates near zero, providing no signal (Figure 3; full results in Appendix E.2.1).

6.2.2. Oracle Perplexity

Setup. Since different rules yield different likelihoods (Section 6.2.1), we ask: what is the lowest perplexity achievable over *all* orderings? This question is only meaningful with exact DUEL likelihood—the ELBO cannot distinguish orderings—making it uniquely enabled by DUEL. For block diffusion BD3-LM ($L' = 4$), we exhaustively search all $4! = 24$ permutations per block, selecting the one minimizing negative log-likelihood—the *oracle* perplexity. Details in Appendix E.2.2.

Results. Table 6 reports results on AG News (Zhang et al., 2015) (details in Appendix E.2.2). Among standard rules, all achieve perplexities 54-57, close to the ARM baseline of 52. The oracle—which selects the lowest-perplexity permutation using ground-truth access not available at test time—achieves 36, surpassing the ARM and all MDM rules.

Table 6. Oracle perplexity (↓) on AG News for BD3-LM (L'=4). DUEL uses block-restricted unmasking; oracle searches all $4! = 24$ block permutations.

Model	**Method**	**Rule F**	**Perplexity**
ARM	Exact	—	52.11
BD3-LM	ELBO	—	≤61.67
BD3-LM	DUEL	Left-to-Right	54.94
	DUEL	Greedy Conf.	56.73
	DUEL	Prob. Margin	57.80
	DUEL	Conf. Thresh.	55.68
BD3-LM	DUEL	Oracle	**36.47**

Takeaway. Unmasking order is a powerful lever unique to MDMs: with ground-truth ordering selection, the same model achieves NLL well below the ARM, showing how much head-room standard unmasking policies leave on the table.

Since the oracle requires ground-truth tokens (which are unavailable at test time), developing strategies that approach its performance without such access is a promising direction; recent work on lookahead unmasking (Lee et al., 2025) takes initial steps, using search to select orderings at test time.

7. Discussion

Summary. We introduced DUEL, a framework that gives MDMs a proper perplexity metric—the natural analogue of autoregressive perplexity—via *exact* likelihood computation under the *deterministic* unmasking policies used at test time. Deterministic unmasking collapses the intractable marginalization to a single path, yielding a plug-and-play metric that requires no training modifications, architectural changes, or continuous-time assumptions. Empirically, proper evaluation via DUEL closes up to 32% of the perplexity gap on in-domain test sets and 82% on zero-shot evaluation, showing that previous comparisons systematically underestimated MDM performance; DUEL also enables principled sampler comparison and reveals via oracle search that MDMs can surpass ARM baselines with optimal orderings. Our framework is intentionally restricted to deterministic unmasking policies; stochastic policies (e.g., remasking, random tie-breaking) fall outside Theorem 4.2 and require separate treatment. Promising extensions include remasking paradigms (Wang et al., 2025a; Huang et al., 2025), GRPO-style reinforcement learning (Shao et al., 2024; Wang et al., 2025b), and speculative decoding (Guo & Ermon, 2025).

Related Work. Existing approaches to likelihood or non-uniform orderings in MDMs all modify training or architecture. Peng et al. (2025a;b); Wang et al. (2025c) train planners or orderings under variational bounds, and Kim et al. (2026) (Puma) restructures the forward process to align training with non-uniform test-time unmasking. Exact likelihood has so far required architectural change—Sahoo et al. (2025) via causal attention and AO-ARM factorization—or continuous-time optimal-denoiser assumptions (Jeon et al., 2025); concurrently, Chen et al. (2025a) apply exact per-trajectory likelihood to GRPO-based RL training. Such training under non-uniform orderings is promising but complex: it breaks the uniform-masking decomposition behind efficient single-pass MDM training. DUEL is complementary and plug-and-play—computing exact likelihood for any standard MDM under any deterministic test-time policy, post-hoc, with no training or architectural changes.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgments

This work was partially funded by the National Science Foundation under award CAREER 2145577, and by the National Institute of Health under award MIRA R35GM151243. GT is partially supported by the NSF Graduate

Research Fellowship under Grant No. DGE-213899.

References

Arriola, M., Gokaslan, A., Chiu, J. T., Yang, Z., Qi, Z., Han, J., Sahoo, S. S., and Kuleshov, V. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025a.

Arriola, M., Schiff, Y., Phung, H., Gokaslan, A., and Kuleshov, V. Encoder-decoder diffusion language models for efficient training and inference. *arXiv preprint arXiv:2510.22852*, 2025b.

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

Balloccu, S., Schmidtová, P., Lango, M., and Dušek, O. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. *arXiv preprint arXiv:2402.03927*, 2024.

Ben-Hamu, H., Gat, I., Severo, D., Nolte, N., and Karrer, B. Accelerated sampling from masked diffusion models via entropy bounded unmasking. *arXiv preprint arXiv:2505.24857*, 2025.

Bie, T., Cao, M., Chen, K., Du, L., Gong, M., Gong, Z., Gu, Y., Hu, J., Huang, Z., Lan, Z., et al. LLaDA 2.0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*, 2025.

Campbell, A., Benton, J., De Bortoli, V., Rainforth, T., Deligiannidis, G., and Doucet, A. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Chang, H., Zhang, H., Jiang, L., Liu, C., and Freeman, W. T. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11315–11325, 2022.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., and Robinson, T. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

Chen, S., Jiao, J., Ratliff, L. J., and Zhu, B. dUltra: Ultra-fast diffusion language models via reinforcement learning. *arXiv preprint arXiv:2512.21446*, 2025a.

Chen, Z., Fang, G., Ma, X., Yu, R., and Wang, X. dParallel: Learnable parallel decoding for dLLMs. *arXiv preprint arXiv:2509.26488*, 2025b.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., and Goharian, N. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*, 2018.

Deschenaux, J. and Gulcehre, C. Beyond autoregression: Fast LLMs via self-distillation through time. *arXiv preprint arXiv:2410.21035*, 2024.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Gat, I., Remez, T., Shaul, N., Kreuk, F., Chen, R. T., Synnaeve, G., Adi, Y., and Lipman, Y. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37:133345–133385, 2024.

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

Grattafiori, A., Dubey, A., Jauhri, A., et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Gulrajani, I. and Hashimoto, T. B. Likelihood-based diffusion language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Guo, G. and Ermon, S. Reviving any-subset autoregressive models with principled parallel sampling and speculative decoding, 2025. URL https://arxiv.org/abs/2504.20456.

Hayakawa, S., Takida, Y., Imaizumi, M., Wakaki, H., and Mitsufuji, Y. Distillation of discrete diffusion through dimensional correlations. *arXiv preprint arXiv:2410.08709*, 2024.

Hayakawa, S., Takida, Y., Imaizumi, M., Wakaki, H., and Mitsufuji, Y. Demystifying maskgit sampler and beyond: Adaptive order selection in masked diffusion. *arXiv preprint arXiv:2510.04525*, 2025.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Hoogeboom, E., Gritsenko, A. A., Bastings, J., Poole, B., Berg, R. v. d., and Salimans, T. Autoregressive diffusion models. *arXiv preprint arXiv:2110.02037*, 2021a.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in neural information processing systems*, 34:12454–12465, 2021b.

Huang, Z., Wang, Y., Chen, Z., and Qi, G.-J. Don't settle too early: Self-reflective remasking for diffusion language models. *arXiv preprint arXiv:2509.23653*, 2025.

Inception Labs, Khanna, S., Kharbanda, S., Li, S., Varma, H., Wang, E., Birnbaum, S., Luo, Z., Miraoui, Y., Palrecha, A., et al. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.

Jazbec, M., Olausson, T. X., Béthune, L., Ablin, P., Kirchhof, M., Monterio, J., Turrisi, V., Ramapuram, J., and Cuturi, M. Learning unmasking policies for diffusion language models. *arXiv preprint arXiv:2512.09106*, 2025.

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

Kim, J., Shah, K., Kontonis, V., Kakade, S., and Chen, S. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. *arXiv preprint arXiv:2502.06768*, 2025a.

Kim, J., Geuter, J., Alvarez-Melis, D., Kakade, S., and Chen, S. Stop training for the worst: Progressive unmasking accelerates masked diffusion training. *arXiv preprint arXiv:2602.10314*, 2026.

Kim, S. H., Hong, S., Jung, H., Park, Y., and Yun, S.-Y. KLASS: KL-guided fast inference in masked diffusion models. *arXiv preprint arXiv:2511.05664*, 2025b.

Kingma, D., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

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

Kuleshov, V. Fast algorithms for sparse principal component analysis based on rayleigh quotient iteration. In *International Conference on Machine Learning*, pp. 1418–1425. PMLR, 2013.

Larochelle, H. and Murray, I. The neural autoregressive distribution estimator. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 29–37. JMLR Workshop and Conference Proceedings, 2011.

Lee, S., Kim, S., Park, J., and Park, D. Lookahead unmasking elicits accurate decoding in diffusion language models. *arXiv preprint arXiv:2511.05563*, 2025.

Li, X., Thickstun, J., Gulrajani, I., Liang, P. S., and Hashimoto, T. B. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022.

Lou, A., Meng, C., and Ermon, S. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 32819–32848, 2024.

Luxembourg, O., Permuter, H., and Nachmani, E. Plan for speed–dilated scheduling for masked diffusion language models. *arXiv preprint arXiv:2506.19037*, 2025.

Marcus, M., Santorini, B., and Marcinkiewicz, M. A. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Nie, S., Zhu, F., Du, C., Pang, T., Liu, Q., Zeng, G., Lin, M., and Li, C. Scaling up masked diffusion models on text. *arXiv preprint arXiv:2410.18514*, 2024.

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

Ou, J., Nie, S., Xue, K., Zhu, F., Sun, J., Li, Z., and Li, C. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, N.-Q., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., and Fernández, R. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 1525–1534, 2016.

Peng, F. Z., Bezemek, Z., Patel, S., Rector-Brooks, J., Yao, S., Bose, A. J., Tong, A., and Chatterjee, P. Path planning for masked diffusion model sampling. *arXiv preprint arXiv:2502.03540*, 2025a.

Peng, F. Z., Bezemek, Z., Rector-Brooks, J., Zhang, S., Zhang, A. R., Bronstein, M., Bose, A. J., and Tong, A. Planner aware path learning in diffusion language models training. *arXiv preprint arXiv:2509.23405*, 2025b.

Pillutla, K., Swayamdipta, S., Zellers, R., Thickstun, J., Welleck, S., Choi, Y., and Harchaoui, Z. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828, 2021.

Pynadath, P., Shi, J., and Zhang, R. Generative frontiers: Why evaluation matters for diffusion language models, 2026. URL https://arxiv.org/abs/2604.02718.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Sahoo, S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J., Rush, A., and Kuleshov, V. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

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

Schiff, Y., Sahoo, S. S., Phung, H., Wang, G., Boshar, S., Dalla-torre, H., de Almeida, B. P., Rush, A., Pierrot, T., and Kuleshov, V. Simple guidance mechanisms for discrete diffusion models. *arXiv preprint arXiv:2412.10193*, 2024.

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

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

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Song, Y., Zhang, Z., Luo, C., Gao, P., Xia, F., Luo, H., Li, Z., Yang, Y., Yu, H., Qu, X., et al. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025.

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

Uria, B., Côté, M.-A., Gregor, K., Murray, I., and Larochelle, H. Neural autoregressive distribution estimation. *Journal of Machine Learning Research*, 17(205):1–37, 2016.

Wang, G., Schiff, Y., Sahoo, S. S., and Kuleshov, V. Remasking discrete diffusion models with inference-time scaling. *arXiv preprint arXiv:2503.00307*, 2025a.

Wang, G., Schiff, Y., Turok, G., and Kuleshov, V. d2: Improved techniques for training reasoning diffusion language models. *arXiv preprint arXiv:2509.21474*, 2025b.

Wang, P., Li, L., Chen, L., Cai, Z., Zhu, D., Lin, B., Cao, Y., Kong, L., Liu, Q., Liu, T., et al. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9440–9450, 2024.

Wang, Y., Schiff, Y., Gokaslan, A., Pan, W., Wang, F., De Sa, C., and Kuleshov, V. InfoDiffusion: Representation learning using information maximizing diffusion models. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36336–36354. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/wang23ah.html.

Wang, Z., Shi, J., Heess, N., Gretton, A., and Titsias, M. K. Learning-order autoregressive models with application to molecular graph generation. *arXiv preprint arXiv:2503.05979*, 2025c.

White, C., Dooley, S., Roberts, M., Pal, A., Feuer, B., Jain, S., Shwartz-Ziv, R., Jain, N., Saifullah, K., Naidu, S., et al. Livebench: A challenging, contamination-free llm benchmark. *arXiv preprint arXiv:2406.19314*, 4, 2024.

Wu, C., Zhang, H., Xue, S., Liu, Z., Diao, S., Zhu, L., Luo, P., Han, S., and Xie, E. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*, 2025.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. XLNet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

Ye, J., Xie, Z., Zheng, L., Gao, J., Wu, Z., Jiang, X., Li, Z., and Kong, L. Dream 7b, 2025. URL https://hkunlp.github.io/blog/2025/dream.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Zheng, K., Chen, Y., Mao, H., Liu, M.-Y., Zhu, J., and Zhang, Q. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *arXiv preprint arXiv:2409.02908*, 2024.

A. Related Works

Masked Diffusion Models. Diffusion models were developed for continuous data (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020) and extended to discrete state-spaces (Hoogeboom et al., 2021b; Austin et al., 2021; Campbell et al., 2022). Lou et al. (2024) proposed SEDD with a score-entropy objective. The absorbing (masking) kernel proved particularly effective, leading Sahoo et al. (2024) and Shi et al. (2024) to develop Masked Diffusion Models (MDM) with simplified training. Zheng et al. (2024) showed MDMs can be interpreted as time-agnostic masked models. MDMs have since scaled dramatically: Nie et al. (2025) introduced LLaDA at 8B parameters, Bie et al. (2025) scaled to 100B, and Inception Labs et al. (2025); Song et al. (2025) demonstrated ultra-fast inference. Arriola et al. (2025a) proposed Block Diffusion interpolating between autoregressive and diffusion paradigms, and Arriola et al. (2025b) introduced E2D2, an encoder-decoder architecture enabling efficient training and KV caching.

Any-order Autoregressive Models. Standard autoregressive models generate left-to-right, but flexible generation orders benefit many tasks. NADE (Larochelle & Murray, 2011; Uria et al., 2016; 2014) pioneered order-agnostic density estimation. XLNet (Yang et al., 2019) proposed permutation language modeling. Hoogeboom et al. (2021a) connected autoregressive and diffusion models. Crucially, MDMs can be reinterpreted as any-order autoregressive models (AO-ARMs): Ou et al. (2024) established equivalence between absorbing diffusion objectives and AO-ARM likelihoods, while Zheng et al. (2024) interpreted MDMs as order-agnostic learners. Kim et al. (2025a) analyzed the exponentially larger masking space MDMs must handle compared to autoregressive models, showing that adaptive inference can mitigate training challenges. Wang et al. (2025c) introduced Learning-Order ARMs (LO-ARMs) that learn context-dependent orderings via a trainable order-policy, achieving state-of-the-art results on molecular graph generation.

Likelihood in Masked Diffusion. Several works derive variational bounds for MDMs: Sahoo et al. (2024) derived a Rao-Blackwellized ELBO as weighted masked language modeling losses, Shi et al. (2024) showed the continuous-time objective is a weighted cross-entropy integral, and Ou et al. (2024) connected these to AO-ARM likelihoods. Jeon et al. (2025) established that cross-entropy losses are tight likelihood estimators under optimal denoisers via the I-MDCE relation. However, their framework assumes continuous-time integral decompositions with optimal denoisers—conditions that differ from practical finite-step sampling with trained networks. Peng et al. (2025a) derived an ELBO that decomposes unmasking into a planner and denoiser, using an external planner to guide sampling; Peng et al. (2025b) extended this to training, jointly learning both components via the P-ELBO. Our work differs: rather than deriving bounds or modifying training, we show standard MDM training already enables exact likelihood evaluation under deterministic samplers. In particular, prior fixed-permutation likelihoods (Uria et al., 2014; Kim et al., 2025a) evaluate a model under a single, *input-independent* ordering π and typically require training under that ordering. DUEL instead computes likelihood under *data-dependent* deterministic policies $\pi^F(\mathbf{x})$—the ordering changes per input because it depends on the model's own predictions—and applies to any uniformly-trained MDM with no retraining.

Masked Diffusion Samplers. MDM sampling can be improved along two axes: the *unmasking policy* π (which positions to reveal) and the *denoiser* p_θ (which tokens to predict) with neural network x_θ. *Unmasking policy improvements.* Heuristic policies include random uniform sampling, confidence-based selection (Chang et al., 2022; Nie et al., 2025), entropy-based prioritization (Ben-Hamu et al., 2025), and margin-based methods (Kim et al., 2025a). KLASS (Kim et al., 2025b) adds a KL-stability criterion to filter confident-but-unstable predictions. Truly learned policies include Jazbec et al. (2025), who train a policy network to select positions. Scheduling methods (Luxembourg et al., 2025; Hayakawa et al., 2025) optimize the unmasking trajectory over time. *Denoiser improvements.* Methods improve x_θ through distillation (Deschenaux & Gulcehre, 2024; Hayakawa et al., 2024) or architectural changes enabling parallel decoding (Wu et al., 2025; Chen et al., 2025b). *Remasking strategies.* Orthogonal to both axes, remasking (Wang et al., 2025a; Huang et al., 2025) allows reconsidering previous decisions by re-masking tokens, enabling iterative refinement absent in standard MDMs. Our framework applies to any deterministic unmasking policy where position selection is fixed given model predictions; extending to remasking is a direction for future work.

Evaluating Generative Models. Perplexity is the standard metric for language models, measuring predictive fit to held-out data, though it has known limitations: it ignores generation quality, can be gamed by degenerate distributions, and does not necessarily capture downstream capabilities. For MDMs, autoregressive perplexity cannot be directly computed; instead, the ELBO provides an upper bound on negative log-likelihood, which translates to a bound on perplexity. However, this bound can be loose and measures the wrong distribution (uniform position selection rather than test-time deterministic policies). When exact likelihood is unavailable, practitioners turn to *sample-based metrics*. Generative perplexity scores model samples under an external reference model (typically GPT-2), but inherits the reference model's biases and ignores sample diversity—a model that repeats one high-quality phrase achieves excellent generative perplexity despite mode collapse; concurrent analysis (Pynadath et al., 2026) catalogues additional failure modes of generative perplexity for diffusion language models. Post-hoc fixes like entropy (measuring diversity) and MAUVE (Pillutla et al., 2021) (comparing generated and reference distributions) attempt to address these issues but remain imperfect: entropy is a secondary diagnostic without guidance on how to trade off against quality, while MAUVE inherits reference model bias and is hyperparameter-sensitive. *Benchmarks* (e.g., MMLU, GSM8K) measure task performance but suffer from data contamination (Balloccu et al., 2024) and gaming (White et al., 2024). Our DUEL exact likelihood computation resolves these issues: it provides a principled per-sample metric that evaluates the test-time distribution directly, requires only the MDM itself, and enables direct comparison with autoregressive models

on the same probabilistic footing.

B. Extended Background

We present masked diffusion models (MDMs) and their interpretation as any-order autoregressive models (AO-ARMs). Appendix B.1 introduces the standard latent-variable formulation following Sahoo et al. (2024); Shi et al. (2024). Appendix B.2 recasts MDMs as AO-ARMs, making explicit the decomposition into position selection and token prediction—a perspective implicit in prior work but essential for our exact likelihood results.

B.1. Masked Diffusion Models

Forward Process. Let $\mathcal{V} = \{1, \ldots, V\}$ denote a vocabulary of V tokens, augmented with a mask token $\mathtt{m} \notin \mathcal{V}$ to form $\bar{\mathcal{V}} = \mathcal{V} \cup \{\mathtt{m}\}$. A clean sequence is $\mathbf{x} = (x^{(1)}, \ldots, x^{(L)}) \in \mathcal{V}^L$, and a partially masked sequence $\mathbf{z}_t \in \bar{\mathcal{V}}^L$ has some positions set to $\mathtt{m}$. We write $\mathcal{M}(\mathbf{z}) = \{\ell : z_\ell = \mathtt{m}\}$ for the set of masked positions.

The forward process progressively masks a clean sequence $\mathbf{x}$ through T discrete steps. We index steps by $t \in \{0, 1, \ldots, T\}$, corresponding to normalized time $t/T \in [0, 1]$. Let $\mathbf{e}_v$ denote the one-hot vector for token v, and let $\alpha_t \in [0, 1]$ be a noise schedule with $\alpha_0 = 1$ (no corruption) and $\alpha_T = 0$ (fully masked). The forward transition from step s to t (where $0 \le s < t \le T$) factorizes independently across positions as $q_{t|s}(\mathbf{z}_t \mid \mathbf{z}_s) = \prod_{\ell=1}^{L} q_{t|s}(z_t^{(\ell)} \mid z_s^{(\ell)})$, where

$$q_{t|s}(z_t^{(\ell)} \mid z_s^{(\ell)}) = \mathrm{Cat}\big(\alpha_{t|s}\, \mathbf{e}_{z_s^{(\ell)}} + (1 - \alpha_{t|s})\, \mathbf{e}_{\mathtt{m}}\big),$$

for $\alpha_{t|s} = \alpha_t / \alpha_s$. Each position independently either remains in its current state (with probability $\alpha_{t|s}$) or transitions to the mask token (with probability $1 - \alpha_{t|s}$). The mask token is an *absorbing state*: positions already masked remain masked. The forward marginal $q_{t|0}(\mathbf{z}_t \mid \mathbf{x})$ follows as a special case with $s = 0$ and $\mathbf{z}_0 = \mathbf{x}$:

$$q_{t|0}(z_t^{(\ell)} \mid x^{(\ell)}) = \mathrm{Cat}\big(\alpha_t\, \mathbf{e}_{x^{(\ell)}} + (1 - \alpha_t)\, \mathbf{e}_{\mathtt{m}}\big).$$

The complete forward trajectory distribution factorizes as:

$$q(\mathbf{z}_{0:T} \mid \mathbf{x}) = q(\mathbf{z}_0 \mid \mathbf{x}) \prod_{t=1}^{T} q_{t|t-1}(\mathbf{z}_t \mid \mathbf{z}_{t-1}), \tag{8}$$

where $q(\mathbf{z}_0 \mid \mathbf{x}) = \prod_{\ell=1}^{L} \mathrm{Cat}(\mathbf{e}_{x^{(\ell)}})$ deterministically sets $\mathbf{z}_0 = \mathbf{x}$, then applies noising transitions $q_{t|t-1}$ that progressively mask tokens until step T.

Reverse Process. The reverse transition from step t to $s < t$ factorizes as $q_{s|t}(\mathbf{z}_s \mid \mathbf{z}_t, \mathbf{x}) = \prod_{\ell=1}^{L} q_{s|t}(z_s^{(\ell)} \mid \mathbf{z}_t, \mathbf{x})$, where

$$q_{s|t}(z_s^{(\ell)} \mid \mathbf{z}_t, \mathbf{x}) = \begin{cases} \mathrm{Cat}(\mathbf{e}_{z_t^{(\ell)}}) & \text{if } z_t^{(\ell)} \neq \mathtt{m} \\ \mathrm{Cat}\big(\beta_{s|t}\, \mathbf{e}_{\mathtt{m}} + \gamma_{s|t}\, \mathbf{e}_{x^{(\ell)}}\big) & \text{if } z_t^{(\ell)} = \mathtt{m} \end{cases} \tag{9}$$

with $\beta_{s|t} = (1 - \alpha_s)/(1 - \alpha_t)$ and $\gamma_{s|t} = (\alpha_s - \alpha_t)/(1 - \alpha_t)$. Unmasked positions remain unchanged; masked positions either stay masked (with probability $\beta_{s|t}$) or reveal their true value (with probability $\gamma_{s|t}$).

Since $\mathbf{x}$ is unknown at generation time, we learn a *denoising network* $x_\theta : \bar{\mathcal{V}}^L \to \mathbb{R}^{L \times V}$ that takes a partially masked sequence $\mathbf{z}$ and outputs logits predicting clean tokens at each position. Following the SUBS parameterization (Sahoo et al., 2024), mask logits are set to $-\infty$ and unmasked positions copy their observed values. We define the token probability matrix $\mathbf{P} = \mathrm{softmax}(x_\theta(\mathbf{z})) \in \Delta_V^L$ and the *denoising distribution* $p_\theta(v \mid \mathbf{z}) \triangleq P_\ell[v]$, the probability assigned to token v at position ℓ. The approximate reverse transition substitutes the denoising distribution for the unknown ground truth in Equation (9):

$$p_\theta(z_s^{(\ell)} \mid \mathbf{z}_t) = q_{s|t}(z_s^{(\ell)} \mid \mathbf{z}_t, \mathbf{x} \leftarrow \mathbf{P}).$$

The complete reverse trajectory distribution factorizes as:

$$p_\theta(\mathbf{x}, \mathbf{z}_{0:T}) = p(\mathbf{z}_T) \cdot p(\mathbf{x} \mid \mathbf{z}_0) \prod_{t=1}^{T} p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t),$$

where $p(\mathbf{z}_T) = \prod_{\ell=1}^{L} \mathrm{Cat}(\mathbf{e}_{\mathrm{m}})$ is the fully masked prior, $p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t)$ are the learned denoising transitions applied backward to $t = 0$, and $p(\mathbf{x} \mid \mathbf{z}_0) = \prod_{\ell=1}^{L} \mathrm{Cat}(\mathbf{e}_{z_0^{(\ell)}})$ deterministically emits $\mathbf{x} = \mathbf{z}_0$.

Likelihood and ELBO. The data likelihood marginalizes over latent trajectories: $p_\theta(\mathbf{x}) = \sum_{\mathbf{z}_{0:T}} p_\theta(\mathbf{x}, \mathbf{z}_{0:T})$. This sum is intractable, so we apply importance sampling with the forward process (8) as proposal and Jensen's inequality:

$$\log p_\theta(\mathbf{x}) = \log \mathbb{E}_{q(\mathbf{z}_{0:T}|\mathbf{x})} \left[\frac{p_\theta(\mathbf{x}, \mathbf{z}_{0:T})}{q(\mathbf{z}_{0:T} \mid \mathbf{x})} \right] \geq \mathbb{E}_{q(\mathbf{z}_{0:T}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z}_{0:T})}{q(\mathbf{z}_{0:T} \mid \mathbf{x})} \right].$$

Training Objective. Following Sahoo et al. (2024), the ELBO simplifies to a weighted sum of cross-entropy losses over masked positions in continuous time:

$$\mathcal{L}_{\mathrm{ELBO}}(\theta) = \mathop{\mathbb{E}}_{\substack{t \sim \mathrm{Unif}[0,1] \\ \mathbf{z}_t \sim q_{t|0}(\cdot|\mathbf{x})}} \left[w_t \sum_{\ell \in \mathcal{M}(\mathbf{z}_t)} \log P_\ell[x^{(\ell)}] \right], \tag{10}$$

where $\mathbf{P} = \mathrm{softmax}(x_\theta(\mathbf{z}_t))$ gives token probabilities at each position. The loss sums the log-probabilities $\log P_\ell[x^{(\ell)}]$ of the true tokens $x^{(\ell)}$ at all masked positions $\ell \in \mathcal{M}(\mathbf{z}_t)$, weighted by $w_t = \alpha_t'/(1 - \alpha_t)$ with $\alpha_t' = \mathrm{d}\alpha_t/\mathrm{d}t$. Because α_t decreases monotonically, $w_t \leq 0$ and $\log P_\ell[x^{(\ell)}] \leq 0$, so the integrand is non-negative. In practice, MDMs use time-embedding-free architectures where x_θ depends only on $\mathbf{z}_t$, since the mask pattern implicitly encodes t.

B.2. Interpretation as Any-Order Autoregressive Models

We reformulate MDMs as any-order autoregressive models (AO-ARMs), which cleanly separates generation into two distinct operations: (1) an *unmasking policy* that selects which positions to reveal, and (2) a *denoising network* that predicts tokens at those positions. This decomposition is essential for our exact likelihood results. Prior work has established the equivalence between MDMs and AO-ARMs (Hoogeboom et al., 2021a; Zheng et al., 2024), with recent work explicitly proving that the MDM and AO-ARM training objectives (ELBOs) are equivalent (Ou et al., 2024; Kim et al., 2025a). However, to our knowledge, no prior work explicitly derives the AO-ARM generative model formulation from first principles. We provide this derivation below, with the joint factorization proved in Appendix D.1 and the ELBO formulation in Appendix D.3.

Ordered Partitions. An *ordered partition* $\sigma = (\sigma_1, \ldots, \sigma_T)$ (Definition 4.1) specifies how positions are revealed across T discrete steps, where each $\sigma_t \subseteq [L] = \{1, \ldots, L\}$ is the set of positions unmasked at step t. The parts satisfy:

$$\bigcup_{t=1}^{T} \sigma_t = [L], \quad \sigma_t \cap \sigma_{t'} = \emptyset.$$

When each part contains exactly one position ($|\sigma_t| = 1$ for all t), we have $T = L$ steps and recover *sequential unmasking*; when parts contain multiple positions, we obtain *parallel unmasking* with $T < L$ steps. We write $\sigma_{<t} \triangleq \sigma_1 \cup \cdots \cup \sigma_{t-1}$ for the set of positions revealed before step t, and $\mathbf{x}^{\sigma_{<t}} \triangleq \{x^{(\ell)}\}_{\ell \in \sigma_{<t}}$ for the clean tokens at those positions. The denoising network constructs its input by placing clean tokens at positions $\sigma_{<t}$ and mask tokens elsewhere, with position IDs.

Forward Process. In MDMs, positions are masked independently with probability $1 - \alpha_t$, which induces a uniform distribution over masking orders. To mirror this in the AO-ARM formulation, we define the forward process as sampling ordered partitions uniformly. For sequential unmasking ($|\sigma_t| = 1$ for all t, so $T = L$), this reduces to:

$$q(\sigma) = \frac{1}{L!}.$$

This ensures the AO-ARM training objective remains equivalent to the MDM objective.

Reverse Process. The reverse process generates a sequence through T discrete steps. At each step t, we (i) select which masked positions to reveal, and (ii) predict the tokens at those positions. This defines a joint distribution over sequences and ordered partitions (derived in Appendix D.1):

$$p_\theta(\mathbf{x}, \sigma) = \prod_{t=1}^{T} \underbrace{\pi_\theta(\sigma_t \mid \mathbf{x}^{\sigma_{<t}})}_{\text{position selection}} \cdot \underbrace{p_\theta(\mathbf{x}^{\sigma_t} \mid \mathbf{x}^{\sigma_{<t}})}_{\text{token prediction}},$$

where $\mathbf{x}^{\sigma_t} \triangleq \{x^{(\ell)}\}_{\ell \in \sigma_t}$. The *unmasking policy* $\pi_\theta(\sigma_t \mid \mathbf{x}^{\sigma_{<t}})$ specifies a distribution over which masked positions to unmask at step t. The token prediction term factorizes under conditional independence (Assumption D.1) as $p_\theta(\mathbf{x}^{\sigma_t} \mid \mathbf{x}^{\sigma_{<t}}) = \prod_{\ell \in \sigma_t} p_\theta(x^{(\ell)} \mid \mathbf{x}^{\sigma_{<t}})$, where $p_\theta(x^{(\ell)} \mid \mathbf{x}^{\sigma_{<t}})$ is the denoising distribution at position ℓ. Standard MDMs use uniform random selection:

$$\pi^{\text{unif}}(\sigma_t \mid \mathbf{z}) = \frac{1}{|\mathcal{M}(\mathbf{z})|},$$

where $|\mathcal{M}(\mathbf{z})|$ is the number of masked positions remaining. Alternative policies may use the denoiser's predictions $\mathbf{P}$ to inform position selection. Since $\mathbf{P}$ is a deterministic function of the context $\mathbf{x}^{\sigma_{<t}}$, such dependencies are implicit in the conditioning.

Likelihood and ELBO. The marginal likelihood sums over all ordered partitions: $p_\theta(\mathbf{x}) = \sum_\sigma p_\theta(\mathbf{x}, \sigma)$. This sum is intractable, analogous to the intractable integral over continuous latent trajectories in MDM. Applying importance sampling with a uniform proposal $q(\sigma)$ and Jensen's inequality yields the ELBO:

$$\log p_\theta(\mathbf{x}) = \log \mathbb{E}_{q(\sigma)} \left[\frac{p_\theta(\mathbf{x}, \sigma)}{q(\sigma)} \right] \geq \mathbb{E}_{q(\sigma)} \left[\log \frac{p_\theta(\mathbf{x}, \sigma)}{q(\sigma)} \right].$$

Training Objective. Under a time-embedding-free denoising network with sequential unmasking ($|\sigma_t| = 1$), the MDM objective (10) is equivalent to the AO-ARM loss:

$$\mathcal{L}_{\text{ELBO}}(\theta) = \mathop{\mathbb{E}}_{q(\sigma)} \left[\sum_{t=1}^{T} \sum_{\ell \in \sigma_t} -\log p_\theta(x^{(\ell)} \mid \mathbf{x}^{\sigma_{<t}}) \right],$$

This is the expected negative log-likelihood when tokens are predicted in uniformly random order, derived in Appendix D.3. The equivalence between MDM and AO-ARM loss functions follows most directly from Kim et al. (2025a).

C. Algorithms

The two procedures below define the DUEL sampler and its exact likelihood computation. Both iterate with a denoising network x_θ and unmasking rule F: compute token probabilities $\mathbf{P} = \mathrm{softmax}(x_\theta(\mathbf{z}))$, select positions via $F(\mathbf{z})$, then either sample tokens (Algorithm 1) or accumulate log-probabilities (Algorithm 2). The two algorithms differ by a single highlighted line.

Algorithm 1 DUEL: Sampling

Require: Denoiser network x_θ, unmasking rule F
Ensure: Generated sequence $\mathbf{x}$

$\mathbf{z} \leftarrow (\mathtt{m}, \ldots, \mathtt{m})$ ▷ Start fully masked
while $\mathcal{M}(\mathbf{z}) \neq \emptyset$ **do**
  $\mathbf{P} \leftarrow \mathrm{softmax}(x_\theta(\mathbf{z}))$ ▷ Token probabilities
  $\mathcal{I} \leftarrow F(\mathbf{z})$ ▷ Positions to unmask
  for $\ell \in \mathcal{I}$ **do**
    $x^{(\ell)} \sim \mathrm{Cat}(P_\ell)$ ▷ Sample token
    $\mathbf{z}[\ell] \leftarrow x^{(\ell)}$ ▷ Reveal token
return $\mathbf{x} \leftarrow \mathbf{z}$ ▷ Fully unmasked sequence

Algorithm 2 DUEL: Exact Likelihood

Require: Sequence $\mathbf{x}$, denoiser network x_θ, unmasking rule F
Ensure: Log-likelihood $\log p_\theta^{\pi^F}(\mathbf{x})$

$\mathbf{z} \leftarrow (\mathtt{m}, \ldots, \mathtt{m})$, $\mathtt{ll} \leftarrow 0$ ▷ Start fully masked
while $\mathcal{M}(\mathbf{z}) \neq \emptyset$ **do**
  $\mathbf{P} \leftarrow \mathrm{softmax}(x_\theta(\mathbf{z}))$ ▷ Token probabilities
  $\mathcal{I} \leftarrow F(\mathbf{z})$ ▷ Positions to unmask
  for $\ell \in \mathcal{I}$ **do**
    $\mathtt{ll} \leftarrow \mathtt{ll} + \log P_\ell[x^{(\ell)}]$ ▷ Accumulate
    $\mathbf{z}[\ell] \leftarrow x^{(\ell)}$ ▷ Reveal token
return $\mathtt{ll}$

D. Proofs

D.1. Joint Factorization of the Reverse Process

We present the joint factorization directly for ordered partitions; the sequential case (one position per step) is a special case. The factorization relies on the following assumption about parallel token prediction.

Assumption D.1 (Conditional Independence Within Groups)**.** Given the revealed tokens $\mathbf{x}^{\sigma_{<t}}$, the tokens at positions within σ_t are predicted independently by the denoising network:

$$p_\theta(\mathbf{x}^{\sigma_t} \mid \mathbf{x}^{\sigma_{<t}}) = \prod_{\ell \in \sigma_t} p_\theta(x^{(\ell)} \mid \mathbf{x}^{\sigma_{<t}}),$$

where $p_\theta(x^{(\ell)} \mid \mathbf{x}^{\sigma_{<t}}) \triangleq P_\ell[x^{(\ell)}]$ is the denoising distribution at position ℓ. The denoising network constructs its input by placing the clean tokens $\mathbf{x}^{\sigma_{<t}}$ at their respective positions with mask tokens elsewhere, and outputs token probabilities $\mathbf{P} = \mathrm{softmax}(x_\theta(\cdot))$.

This assumption is standard in MDM generation and holds by construction: the denoising network outputs independent distributions over tokens at each position. Because Algorithm 1 samples each position in σ_t independently from $\mathbf{P}$, this factorization holds exactly for the model's induced distribution $p_\theta^{\pi^F}$—it is only an approximation of the true data distribution p_{data} when intra-group dependencies exist. When each group contains exactly one position ($|\sigma_t| = 1$ for all

t), we have $T = L$ steps and the ordered partition reduces to a permutation over positions; in this case, the assumption is trivially satisfied since each group contains a single token.

Proposition D.2 (Joint Factorization)**.** *Let $\sigma = (\sigma_1, \ldots, \sigma_T)$ be an ordered partition (Definition 4.1), where $\sigma_t \subseteq [L]$ is the set of positions unmasked at step t. The joint distribution of sequence $\mathbf{x}$ and ordered partition σ under the reverse process factorizes as:*

$$p_\theta(\mathbf{x}, \sigma) = \prod_{t=1}^{T} \pi_\theta(\sigma_t \mid \mathbf{x}^{\sigma_{<t}}) \cdot p_\theta(\mathbf{x}^{\sigma_t} \mid \mathbf{x}^{\sigma_{<t}}), \tag{11}$$

where $\sigma_{<t} \triangleq \sigma_1 \cup \cdots \cup \sigma_{t-1}$, $\mathbf{x}^{\sigma_{<t}} \triangleq \{x^{(\ell)}\}_{\ell \in \sigma_{<t}}$ is the set of clean tokens at positions revealed before step t, and $\mathbf{x}^{\sigma_t} \triangleq \{x^{(\ell)}\}_{\ell \in \sigma_t}$. Under Assumption D.1, $p_\theta(\mathbf{x}^{\sigma_t} \mid \mathbf{x}^{\sigma_{<t}}) = \prod_{\ell \in \sigma_t} p_\theta(x^{(\ell)} \mid \mathbf{x}^{\sigma_{<t}})$.

Proof. Applying the chain rule, then factoring each term into position selection and token prediction:

$$\begin{aligned} p_\theta(\mathbf{x}, \sigma) &= \prod_{t=1}^{T} p_\theta(\mathbf{x}^{\sigma_t}, \sigma_t \mid \mathbf{x}^{\sigma_{<t}}, \sigma_{<t}) \\ &= \prod_{t=1}^{T} \pi_\theta(\sigma_t \mid \mathbf{x}^{\sigma_{<t}}, \sigma_{<t}) \cdot p_\theta(\mathbf{x}^{\sigma_t} \mid \sigma_t, \mathbf{x}^{\sigma_{<t}}, \sigma_{<t}) \\ &= \prod_{t=1}^{T} \pi_\theta(\sigma_t \mid \mathbf{x}^{\sigma_{<t}}) \cdot p_\theta(\mathbf{x}^{\sigma_t} \mid \mathbf{x}^{\sigma_{<t}}). \end{aligned}$$

The first equality is the chain rule; at $t = 1$, $p_\theta(\mathbf{x}^{\sigma_1}, \sigma_1 \mid \mathbf{x}^{\sigma_{<1}}, \sigma_{<1}) = p_\theta(\mathbf{x}^{\sigma_1}, \sigma_1)$ since $\mathbf{x}^{\sigma_{<1}} = \emptyset$ and $\sigma_{<1} = \emptyset$. The second factors each term via $p(A, B \mid C) = p(B \mid C) \cdot p(A \mid B, C)$. The third drops $\sigma_{<t}$ (redundant since $\mathbf{x}^{\sigma_{<t}}$ encodes positions via position indices) and drops σ_t from the denoiser (which predicts all masked positions simultaneously without σ_t as input). Applying Assumption D.1 yields Equation (11). □

D.2. Proposal Distribution

For sequential unmasking ($|\sigma_t| = 1$ for all t), the forward masking process induces a uniform distribution over unmasking orders.

Proposition D.3 (Uniform Proposal Distribution)**.** *Under sequential unmasking, the forward masking process induces a uniform distribution over unmasking orders:*

$$q(\sigma) = \frac{1}{L!} \quad \textit{for all permutations } \sigma.$$

Proof. We relate the forward *masking* process to the reverse *unmasking* order.

Step 1: Forward masking process. Starting from $\mathbf{z}^{(0)} = \mathbf{x}$, at each step $t = 1, \ldots, L$, we select one of the $L - t + 1$ currently unmasked positions uniformly at random and mask it. Let τ_t denote the position masked at step t. The probability of any specific masking order $\tau = (\tau_1, \ldots, \tau_L)$ is:

$$q(\tau) = \prod_{t=1}^{L} \frac{1}{L - t + 1} = \frac{1}{L} \cdot \frac{1}{L-1} \cdots \frac{1}{1} = \frac{1}{L!}.$$

Step 2: Unmasking order reverses masking order. The unmasking order σ specifies the sequence in which positions are *revealed* during generation. This is the reverse of the masking order: position τ_L (last to be masked) is the first to be unmasked.

Step 3: Uniformity is preserved under reversal. Since reversal is a bijection on permutations, and $q(\tau) = 1/L!$ for all τ, we have $q(\sigma) = 1/L!$ for all σ. □

D.3. Deriving the AO-ARM Training Objective (ELBO)

Although the ELBO training objective for MDMs has been established in prior work (Hoogeboom et al., 2021a; Ou et al., 2024; Kim et al., 2025a), we provide the derivation from first principles here for completeness. We derive the result for sequential unmasking (one position per step), where σ is a permutation of $[L]$ and $\sigma_t \in [L]$ denotes the single position unmasked at step t. Sequential unmasking is required for the uniform proposal $q(\sigma) = 1/L!$ from Proposition D.3, which enables the policy–proposal cancellation in Step 2 of the proof.

Proposition D.4 (ELBO Training Objective)**.** *Let $p_\theta(\mathbf{x}, \sigma)$ be the joint distribution from Proposition D.2 under sequential unmasking with uniform policy, and let $q(\sigma) = 1/L!$ be the proposal from Proposition D.3. Then:*

$$\log p_\theta(\mathbf{x}) \geq -\mathcal{L}_{ELBO}(\theta; \mathbf{x}),$$

where

$$\mathcal{L}_{ELBO}(\theta; \mathbf{x}) = \mathop{\mathbb{E}}_{q(\sigma)} \left[\sum_{t=1}^{L} -\log p_\theta(x^{(\sigma_t)} \mid \mathbf{x}^{\sigma_{<t}}) \right], \tag{12}$$

where $\sigma_t \in [L]$ is the position unmasked at step t.

Proof. We proceed in two steps.

Step 1: Importance sampling and Jensen's inequality. The marginal likelihood requires summing over all $L!$ unmasking orders. We rewrite this as an expectation under the uniform proposal $q(\sigma) = 1/L!$:

$$p_\theta(\mathbf{x}) = \sum_\sigma p_\theta(\mathbf{x}, \sigma) = \sum_\sigma q(\sigma) \cdot \frac{p_\theta(\mathbf{x}, \sigma)}{q(\sigma)} = \mathbb{E}_{\sigma \sim q} \left[\frac{p_\theta(\mathbf{x}, \sigma)}{q(\sigma)} \right].$$

Taking logarithms and applying Jensen's inequality:

$$\log p_\theta(\mathbf{x}) = \log \mathbb{E}_{\sigma \sim q} \left[\frac{p_\theta(\mathbf{x}, \sigma)}{q(\sigma)} \right] \geq \mathbb{E}_{\sigma \sim q} \left[\log \frac{p_\theta(\mathbf{x}, \sigma)}{q(\sigma)} \right] \triangleq -\mathcal{L}_{\text{ELBO}}(\theta; \mathbf{x}).$$

Step 2: Substituting factorizations and simplifying. From Propositions D.2 and D.3, we expand the log-ratio. With a uniform unmasking policy $\pi^{\text{unif}}(\sigma_t \mid \mathbf{x}^{\sigma_{<t}}) = 1/(L - t + 1)$ (where $L - t + 1$ is the number of remaining masked positions), the policy terms cancel with the proposal:

$$\log \frac{p_\theta(\mathbf{x}, \sigma)}{q(\sigma)} = \sum_{t=1}^{L} \underbrace{\log \frac{1/(L-t+1)}{1/(L-t+1)}}_{=0} + \sum_{t=1}^{L} \log p_\theta(x^{(\sigma_t)} \mid \mathbf{x}^{\sigma_{<t}}).$$

The first sum vanishes, yielding equation (12). □

D.4. Theoretical Properties of DUEL

We prove the main theoretical result for DUEL samplers: exact likelihood computation. Policy-dependence—that different deterministic rules induce different distributions over the same denoiser x_θ—is established empirically in Section 6.2, where on a fixed pretrained model the rules in Table 1 yield substantially different DUEL perplexities (Table 5).

Theorem D.5 (DUEL Exact Likelihood)**.** *Under a deterministic policy π_θ^F, exactly one ordered partition $\sigma^* = (\sigma_1^*, \ldots, \sigma_T^*)$ satisfies $p_\theta^{\pi^F}(\mathbf{x}, \sigma^*) > 0$. The log-likelihood is:*

$$\log p_\theta^{\pi^F}(\mathbf{x}) = \sum_{t=1}^{T} \sum_{\ell \in \sigma_t^*} \log p_\theta(x^{(\ell)} \mid \mathbf{x}^{\sigma^*_{<t}}), \tag{6}$$

where $\sigma_t^* = F(\mathbf{x}^{\sigma^*_{<t}})$ is the set of positions selected at step t.

Proof of Theorem 4.2. The key insight is that a deterministic policy eliminates the intractable sum over orderings: exactly one ordering has nonzero probability, so the sum collapses to a single computable term.

Step 1: The deterministic rule induces a unique trajectory. Given any sequence $\mathbf{x}$ and a deterministic rule F, there is exactly one unmasking trajectory $\sigma^* = (\sigma_1^*, \ldots, \sigma_T^*)$ that the generative process can follow. To see this, note that generation starts from the same fully masked state and proceeds deterministically: at step t, the rule F examines the current revealed tokens $\mathbf{x}^{\sigma^*_{<t}}$ and returns a fixed set of positions $\sigma_t^* = F(\mathbf{x}^{\sigma^*_{<t}})$. We then reveal the true tokens at those positions, yielding the next set of revealed tokens. Since F is a function (not a distribution) and revealed tokens remain fixed (the absorbing property), each step is uniquely determined by the previous state.

Step 2: All other orderings have zero probability. Consider any ordering $\sigma \neq \sigma^*$. Since σ differs from σ^*, there must be some first step t where $\sigma_t \neq \sigma_t^*$. At this step, the revealed tokens are identical (both orderings have revealed the same positions up to step $t-1$), so the rule returns the same positions: $F(\mathbf{x}^{\sigma_{<t}}) = F(\mathbf{x}^{\sigma^*_{<t}}) = \sigma_t^*$. But since $\sigma_t \neq \sigma_t^*$, we have $\sigma_t \neq F(\mathbf{x}^{\sigma_{<t}})$. By the definition of the deterministic policy (Equation (3)), $\pi_\theta^F(\sigma_t \mid \mathbf{x}^{\sigma_{<t}}) = 0$, which makes the entire joint probability $p_\theta^{\pi^F}(\mathbf{x}, \sigma) = 0$.

Step 3: The marginal reduces to a single term. Since all orderings except σ^* contribute zero probability, the marginal likelihood simplifies:

$$p_\theta^{\pi^F}(\mathbf{x}) = \sum_\sigma p_\theta^{\pi^F}(\mathbf{x}, \sigma) = p_\theta^{\pi^F}(\mathbf{x}, \sigma^*).$$

Expanding the surviving term using the joint factorization (Equation (2)):

$$p_\theta^{\pi^F}(\mathbf{x}, \sigma^*) = \prod_{t=1}^{T} \underbrace{\pi_\theta^F(\sigma_t^* \mid \mathbf{x}^{\sigma^*_{<t}})}_{=1} \cdot \prod_{\ell \in \sigma_t^*} p_\theta(x^{(\ell)} \mid \mathbf{x}^{\sigma^*_{<t}}).$$

Each policy term equals 1 because $\sigma_t^* = F(\mathbf{x}^{\sigma^*_{<t}})$ by construction. Taking logarithms yields Equation (6). □

E. Experimental Details

We provide details for each experiment. Training and architecture follow Sahoo et al. (2024) and Arriola et al. (2025a).

E.1. Perplexity Gap Experiments

Data. We evaluate on two in-domain datasets: OpenWebText (OWT; Gokaslan et al., 2019) with GPT-2 tokenizer (Radford et al., 2019), and One Billion Words (LM1B; Chelba et al., 2013) with BERT tokenizer (Devlin et al., 2019). For zero-shot evaluation, we train on OWT and evaluate on Penn Tree Bank (Marcus et al., 1993), Wikitext (Merity et al., 2016), LM1B, Lambada (Paperno et al., 2016), AG News (Zhang et al., 2015), and Scientific Papers (PubMed and ArXiv subsets) (Cohan et al., 2018). All sequences are concatenated and wrapped following Arriola et al. (2025a). We use the full validation/test split for all evaluations. For large-scale experiments, we evaluate on Wikitext, Lambada, and AG News.

Models. For test and zero-shot perplexity, we evaluate: (1) an autoregressive model (ARM) baseline, (2) SEDD (Lou et al., 2024), (3) MDLM (Sahoo et al., 2024), and (4) BD3-LM (Arriola et al., 2025a) with block sizes $L' \in \{4, 8, 16\}$. All 110M models use 12-layer DiT, hidden dim 768, 12 heads, with sequence length $L = 1024$ (OWT) or $L = 128$ (LM1B), trained for 1M steps with batch size 512, making them directly comparable. For large-scale experiments, we evaluate LLaDA-8B-Base (Nie et al., 2025) with sequence length $L = 2048$ and include Llama3-8B-Base (Grattafiori et al., 2024) as a reference; note these models differ in training data and compute budget, so we do not compute gap closed for this comparison.

Likelihood Estimators. We compare three methods for computing negative log-likelihood, listed as "Method" in our tables.

- **Exact** (ARM baseline). The autoregressive model computes exact log-likelihood via the chain rule of probability:

$$\log p(\mathbf{x}) = \sum_{\ell=1}^{L} \log p(x^{(\ell)} \mid x^{(1)}, \ldots, x^{(\ell-1)}).$$

Table 7. Zero-shot perplexity (↓) for models trained on OWT, evaluated on unseen datasets. ARM baseline in first row. **Bold**: better of ELBO/DUEL per model-dataset pair.

Model	Method	PTB	Wikitext	LM1B	Lambada	AG News	PubMed	ArXiv
ARM	Exact	81.07	25.32	51.14	52.13	52.11	48.59	41.22
SEDD	ELBO	≤96.33	≤35.98	≤68.14	≤48.93	≤67.82	≤43.13	≤37.89
	DUEL	**91.56**	**31.63**	**64.39**	**46.01**	**63.78**	**39.31**	**35.48**
MDLM	ELBO	≤90.96	≤33.22	≤64.94	≤**48.29**	≤62.78	≤43.13	≤37.89
	DUEL	**87.57**	**30.97**	**60.94**	48.42	**59.81**	**41.96**	**37.57**
BD3-LM	ELBO	≤96.81	≤31.31	≤60.88	≤50.03	≤61.67	≤42.52	≤39.20
L'=4	DUEL	**83.94**	**29.43**	**57.98**	**48.56**	**56.73**	**41.41**	**38.11**

Each conditional is computed in a single, batched forward pass using causal masking, so the full log-likelihood requires one forward pass over the sequence.

- **ELBO** (MDMs). The ELBO provides an upper bound on the true negative log-likelihood. Because the MDM loss (Equation (1)) involves an expectation over random timesteps and masked positions, computing it exactly is intractable; instead, we use a K-sample Monte Carlo estimate of the ELBO. For MDLM/SEDD, the loss weights each masked position by L/n_t where n_t is the number of masked positions; for BD3-LM, the loss factorizes over blocks with preceding blocks revealed (see Arriola et al., 2025a, Eq. 8). We use $K = 1$ Monte Carlo sample with a low-discrepancy sampler (Kingma et al., 2021) for the 110M models, and $K = 128$ samples for LLaDA-8B.

- **DUEL** (MDMs). Computes exact likelihood under a deterministic unmasking policy by summing the log-probabilities of revealed tokens along the single path induced by the unmasking rule (Algorithm 1 in the main paper).

Metrics. Each likelihood estimator above produces a negative log-likelihood $\mathrm{NLL} = -\log p(\mathbf{x})$ for a sequence $\mathbf{x}$ of length L. We convert this to perplexity: $\mathrm{PPL} = \exp(\mathrm{NLL}/L)$. To compare MDMs against the ARM baseline, we define the *perplexity gap* $\Delta = \mathrm{PPL}_{\mathrm{MDM}} - \mathrm{PPL}_{\mathrm{ARM}}$. Under the ELBO, $\Delta_{\mathrm{ELBO}} = \mathrm{PPL}^{\mathrm{ELBO}}_{\mathrm{MDM}} - \mathrm{PPL}_{\mathrm{ARM}}$; under DUEL, $\Delta_{\mathrm{DUEL}} = \mathrm{PPL}^{\mathrm{DUEL}}_{\mathrm{MDM}} - \mathrm{PPL}_{\mathrm{ARM}}$. The *gap closed* (%) measures the fraction of the ELBO gap eliminated by exact evaluation:

$$\text{Gap Closed} = \frac{\Delta_{\mathrm{ELBO}} - \Delta_{\mathrm{DUEL}}}{\Delta_{\mathrm{ELBO}}} \times 100\%.$$

Note that in zero-shot experiments (Table 3) we exclude Lambada, PubMed, and ArXiv datasets and do not report the gap closed metric. This is because the MDM already outperforms the ARM baseline under the ELBO on these datasets, making $\Delta_{\mathrm{ELBO}} \leq 0$ and gap closed ill-defined; nonetheless, DUEL still yields similar-magnitude perplexity reductions on these datasets (Table 7).

Protocol. We evaluate on the full validation/test split for both ELBO and DUEL. For the 110M models, DUEL uses block-greedy unmasking with $k = 1$ token per step. Each BD3-LM is evaluated at its native training block size ($L' \in \{4, 8, 16\}$); SEDD and MDLM, which are trained globally, are evaluated under block-greedy unmasking with $L' = 4$ to match the smallest BD3-LM block size for a like-for-like comparison. For LLaDA-8B-Base, DUEL uses block-greedy unmasking with $k = 1$ and block size $L' = 32$.

DUEL perplexity outperforms ELBO perplexity on every model and dataset comparison except one. The sole exception is MDLM on Lambada, where the ELBO (≤48.29) is slightly lower than DUEL (48.42); this is not contradictory, as they measure different distributions ($p_\theta^{\pi^{\mathrm{unif}}}$ vs. $p_\theta^{\pi^F}$).

E.2. Comparing Sampling Strategies

This section provides experimental details for Section 6.2.

E.2.1. Comparing Fast Samplers

Data. OpenWebText (Gokaslan et al., 2019) validation split (1,000 samples, $L = 1024$, GPT-2 tokenizer).

Model. BD3-LM (Arriola et al., 2025a) with block size $L' = 16$, trained on OWT.

Unmasking Rules. We compare four deterministic rules, all restricted to operate within blocks of size $L' = 16$. Let $\mathcal{M} = \mathcal{M}(\mathbf{z})$ denote the masked positions and write $k' \triangleq \min(k, |\mathcal{M}|)$ so that the boundary case $|\mathcal{M}| < k$ is handled explicitly.

1. **Left-to-Right**: $F(\mathbf{z}) = \{k' \text{ smallest } \ell \in \mathcal{M}\}$
2. **Greedy Confidence**: $F(\mathbf{z}) = \arg\max_{\mathcal{I} \subseteq \mathcal{M},\, |\mathcal{I}|=k'} \sum_{\ell \in \mathcal{I}} P_\ell^{(1)}$
3. **Probability Margin** (Kim et al., 2025a): $F(\mathbf{z}) = \arg\max_{\mathcal{I} \subseteq \mathcal{M},\, |\mathcal{I}|=k'} \sum_{\ell \in \mathcal{I}} \big(P_\ell^{(1)} - P_\ell^{(2)}\big)$, where $P_\ell^{(1)}, P_\ell^{(2)}$ are the top-2 token probabilities
4. **Confidence Threshold** (Wu et al., 2025): Unmask all $\ell \in \mathcal{M}$ where $P_\ell^{(1)} > \tau$; if none exceed τ, unmask the most confident position

Metrics. We report four metrics:

1. **DUEL perplexity** (↓): Exact likelihood computed on OWT validation data
2. **Generative perplexity** (↓): Generate 1,000 samples, evaluate under GPT-2 Large (Radford et al., 2019)
3. **Entropy** (↑): Token-level entropy of generated text; low entropy indicates repetitive or degenerate output
4. **MAUVE** (↑) (Pillutla et al., 2021): Distribution similarity between 1,000 generated samples and 1,000 reference sequences from OWT

Protocol. All samplers target the same NFE budgets $\{128, 256, 512, 1024\}$ for like-for-like comparison. For left-to-right, greedy, and probability margin, this is achieved by fixing the tokens unmasked per step $k \in \{8, 4, 2, 1\}$, giving exactly $\lceil L/k \rceil$ NFE for sequence length $L = 1024$.

Confidence threshold instead adaptively selects how many tokens to unmask based on whether predictions exceed a threshold τ, so the per-sequence NFE varies and the same τ produces different mean NFEs under evaluation (on held-out data) and generation. We therefore calibrate τ separately for each setting to hit the same NFE budgets:

- DUEL perplexity: $\tau \in \{0.05, 0.07, 0.15, 0.99\}$ achieves empirical mean NFE $\{140, 242, 514, 1003\}$ (averaging `val/num_decoding_steps` over validation batches).
- Sample generation (generative perplexity, entropy, MAUVE): $\tau \in \{0.045, 0.08, 0.16, 1.0\}$ achieves mean NFE $\{131, 278, 518, 921\}$.

Table 5 reports confidence threshold at these calibrated NFE budgets; Figure 3 plots all available data points.

Results. Figure 3 presents the full comparison across all four metrics.

DUEL *perplexity.* Rankings are consistent across NFE budgets: probability margin achieves the lowest perplexity at low NFE, followed by greedy confidence, with left-to-right and confidence threshold performing worst. All methods converge to similar perplexity (≈21–22) at 1024 NFE, approaching the ELBO baseline (≤23.52). The degradation from 1024 to 128 NFE ranges from 6× (probability margin) to 11× (left-to-right).

Generative perplexity. Rankings are inconsistent: left-to-right achieves the *lowest* generative perplexity at 128 and 256 NFE despite having one of the worst DUEL perplexities at these budgets (worst at 128 NFE; second-worst at 256 NFE, behind only confidence threshold). This apparent contradiction is explained by the entropy results below. The magnitude of degradation is also distorted, showing 30–45× increase from 1024 to 128 NFE.

Entropy. Left-to-right exhibits anomalously low entropy (≈4.1) at intermediate NFE (256–512), indicating degenerate, repetitive text. This explains its misleadingly favorable generative perplexity: repetitive text often achieves low

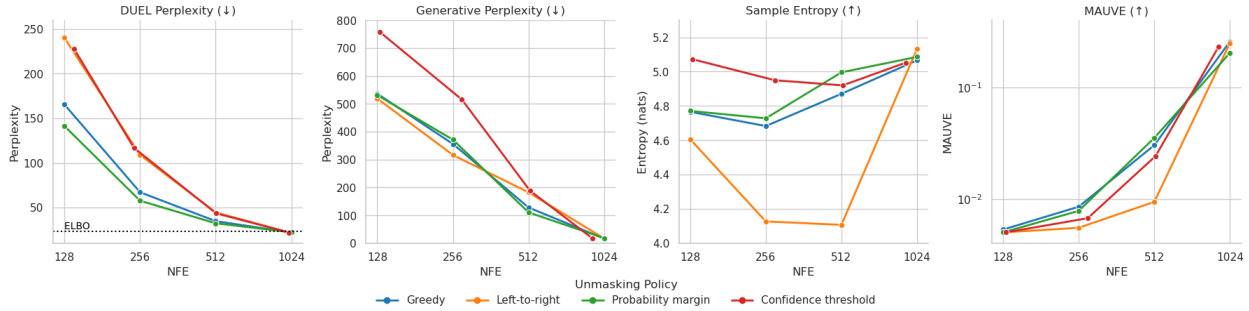

Figure 3. **Comparing fast samplers across four metrics.** Dashed line in DUEL perplexity: ELBO ($\leq$23.52). DUEL perplexity yields consistent rankings across NFE (probability margin best at low NFE), while generative perplexity rankings cross repeatedly. Entropy reveals that left-to-right produces degenerate low-entropy text at low NFE despite achieving favorable generative perplexity. MAUVE saturates near zero at low NFE for all rules, providing little discriminative signal.

perplexity under autoregressive models. In contrast, probability margin and greedy confidence maintain higher entropy ($\approx$4.7–5.0) across NFE budgets, indicating more diverse output. All methods converge to entropy $\approx$5.1 at 1024 NFE.

MAUVE. All methods achieve MAUVE ≈ 0.005 at 128 NFE, providing minimal discriminative signal. Scores improve to 0.20–0.26 at 1024 NFE, but remain below typical thresholds for high-quality generation. The log-scale behavior indicates MAUVE is most useful for distinguishing high-quality regimes, not for comparing degraded generation.

Takeaways. Sample-based metrics can be misleading: generative perplexity rewards degenerate text, entropy requires careful interpretation, and MAUVE saturates at low quality. DUEL perplexity provides consistent, interpretable rankings that correctly identify probability margin as optimal at low NFE budgets.

E.2.2. Oracle Perplexity

Data. AG News (Zhang et al., 2015) zero-shot evaluation (GPT-2 tokenizer, $L = 1024$).

Model. BD3-LM with block size $L' = 4$, trained on OWT (same model as Section 6.1).

Method. For BD3-LM, DUEL processes each block of size L' independently. Within a block, the unmasking order is a permutation $\pi \in S_{L'}$ of the masked positions, and each permutation induces a different factorization of the block likelihood via Equation (6). The oracle exhaustively evaluates all $L'!$ permutations per block and selects the one that minimizes the block's negative log-likelihood:

$$\mathrm{NLL}_{\text{oracle}} = \sum_{b=1}^{L/L'} \min_{\pi \in S_{L'}} \mathrm{NLL}^{(b)}_{\text{DUEL}}(\pi),$$

where $\mathrm{NLL}^{(b)}_{\text{DUEL}}(\pi)$ is the negative log-likelihood contribution of block b under ordering π, computed by running Algorithm 2 with a left-to-right unmasking rule applied to the permuted block positions. Concretely, for each block b:

1. Enumerate all $L'!$ permutations $\pi_1, \ldots, \pi_{L'!}$ of the L' positions in block b.
2. For each permutation π_i, run DUEL with the unmasking order defined by π_i: unmask positions in the order $\pi_i(1), \pi_i(2), \ldots, \pi_i(L')$, recording the log-probability of each ground-truth token as it is revealed.
3. Select $\pi^* = \arg\min_{\pi_i} \mathrm{NLL}^{(b)}_{\text{DUEL}}(\pi_i)$.

For $L' = 4$, this evaluates $4! = 24$ permutations per block, incurring a $24\times$ cost relative to a single DUEL evaluation.

Unmasking Rules. In addition to the oracle, we evaluate four standard deterministic rules for comparison: left-to-right, greedy confidence ($k = 1$), probability margin ($k = 1$), and confidence threshold ($\tau = 0.95$). All rules are restricted to block size $L' = 4$; see Appendix E.2.1 for rule definitions.