ORDERED-DLM: ORDERING TOKEN GENERATION THROUGH SYNTACTIC HIERARCHY

Anonymous authors
Paper under double-blind review

ABSTRACT

Masked diffusion language models generate text by iteratively revealing masked tokens, enabling parallel generation and bidirectional conditioning. This flexibility raises a fundamental scheduling question: when multiple positions are masked, which token should be revealed first? Each reveal not only resolves one position but also changes the context available for predicting what remains. Motivated by information-gain principles in LLM reasoning, we prioritize revealing tokens that reduce uncertainty in predicting the remaining tokens, an objective we call *contextual planning*. To realize this objective, we use syntactic hierarchy to identify tokens with broad structural roles in an unsupervised manner. We introduce Ordered-DLM, a masked diffusion framework that predicts continuous syntactic heights, uses their relative ordering to induce differentiable hierarchy scores, and applies these scores to both forward masking and reverse denoising. The hierarchy is learned jointly with the diffusion model , requiring neither an external parser nor syntactic annotations. Experiments on LM1B and OpenWebText demonstrate lower perplexity than strong diffusion baselines on in-domain and zero-shot benchmarks, as well as lower generative perplexity across sampling budgets, while preserving parallel decoding. These results show that syntactic hierarchy-induced scheduling improves both contextual planning and language-modeling quality.

1 INTRODUCTION

Discrete diffusion language models generate text by repeatedly predicting and revealing masked tokens. Unlike autoregressive models, they can process many positions in parallel and condition on context from both directions (Shi et al., 2024; Lou et al., 2024; Sahoo et al., 2024). Their flexibility introduces a fundamental question: when several positions are masked, which token should be revealed first? Importantly, a revealed token does not only resolve its own position; it also becomes context for predicting the remaining masked tokens. The generation order therefore determines how information is progressively exposed throughout denoising. A similar perspective has recently emerged in LLM reasoning (Ton et al., 2025; Guo, 2025) and agentic (Hu et al., 2026; DENG et al., 2026; Liu et al., 2024; Chan et al., 2025) decision making, where the value of an intermediate step can be characterized by the information it provides for subsequent reasoning. Motivated by this perspective, we seek a generation order in which early token reveals make the remaining masked positions easier to predict. We refer to this objective as *contextual planning*.

Organizing such an informative generation order for general text is a central challenge. Recent works on diffusion language models (Ma et al., 2026; Zeng et al., 2025) show that incorporating structure into the masking process can improve code and mathematical reasoning. Structure is appealing for contextual planning because it provides a way to distinguish tokens with broad contextual roles from local details. However, existing approaches that explicitly impose structural guidance often rely on externally provided structure, such as parser-derived trees, or on ground-truth supervision for the generation order. Such requirements are difficult to satisfy for large-scale text corpus, where explicit structural annotations may be unavailable. We therefore aim to induce the structure underlying the generation order directly from the language-modeling objective in an unsupervised manner.

To obtain such structure without external supervision, we turn to latent syntactic organization, which provides a natural way to organize text into nested constituents. Prior work has shown that such

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

structure can be induced from raw text using continuous syntactic heights (Luo et al., 2019; Shen et al., 2021) or distances (Shen et al., 2018a;b; 2019), whose relative values encode constituent boundaries and nesting . Building on these ideas, we use the relative ordering of predicted syntactic heights to induce a syntactic hierarchy over tokens. Our model predicts a continuous height for each token, converts the resulting height relations into differentiable hierarchy scores, and uses these scores to guide both masking and reverse denoising.

To this end, we introduce Ordered-DLM, a syntactic hierarchy-induced masked diffusion framework. Its scheduler is trained jointly with the diffusion model, allowing it to discover an informative ordering from the language-modeling objective without human annotations or an external parser. As shown in our contextual-planning analysis (Sec. 4.3), the resulting schedule preferentially reveals tokens that provide context for predicting the remaining masked sequence, yielding a contextually informative reveal order. This improved contextual planning leads to consistent gains in language modeling and generation. Ordered-DLM achieves lower perplexity than strong diffusion baselines on in-domain and zero-shot benchmarks, as well as lower generative perplexity across sampling budgets. These results show that hierarchy-induced contextual planning improves both the reveal order and the quality of the generated text.

2 Related Work

2.1 Generation Order and Masking Schedules in Diffusion Language Models

Discrete diffusion language models perform categorical denoising, often with an absorbing mask state (Austin et al., 2021). SEDD and MDLM improve masked-diffusion objectives (Lou et al., 2024; Sahoo et al., 2024). However, structured masking mechanism for generic text has been an open challenge(Wang et al., 2026). DiffusionBERT, InfoDiffusion, and GenMD4 introduce token- or state-dependent corruption schedules (He et al., 2023; Wang et al., 2023; Shi et al., 2024). Generation order has likewise been studied through confidence refinement (Ghazvininejad et al., 2019), blockwise generation (Arriola et al., 2025), and learned policies based on post-training, reinforcement learning, pathwise rewards, or ground-truth supervision (Hong et al., 2026a; Jazbec et al., 2026; Xu et al., 2026; Asano et al., 2026). LoMDM jointly learns a context-dependent order and diffusion model (Hong et al., 2026b), while TreeDiff uses an external AST for code generation (Zeng et al., 2025).

Our method uses a differentiable height predictor to assign each token a scalar value, and a Cartesian-tree analog converts the relative heights into hierarchy scores. These scores control both the forward masking schedule and the reverse reveal priority, allowing tokens that dominate larger spans to provide context before local details. The hierarchy is learned jointly with the diffusion model, without an external parser or token-level annotations.

2.2 Self-Supervised Syntax Parsing

Prior work has shown that hierarchical syntactic structure can be induced from raw text through continuous latent representations. PRPN learns syntactic distances jointly with language modeling to capture constituency structure (Shen et al., 2018a), while Shen et al. (2018b) further use syntactic distances to recover constituency trees. Ordered Neurons introduces an architectural inductive bias for learning hierarchical structure (Shen et al., 2019), and Tree Transformer incorporates differentiable tree constraints into self-attention (Wang et al., 2019b). Luo et al. (2019) introduce syntactic heights to represent hierarchical prominence, and StructFormer jointly induces dependency and constituency structure from masked language modeling using latent syntactic structure (Shen et al., 2021). DIORA, URNNG, and Compound PCFG instead induce latent trees through recursive or probabilistic grammar formulations (Drozdov et al., 2019; Kim et al., 2019b;a).

Our work builds on the continuous syntactic-representation perspective, but uses predicted token heights to induce differentiable hierarchy scores for controlling the masking and denoising schedules of a diffusion language model.

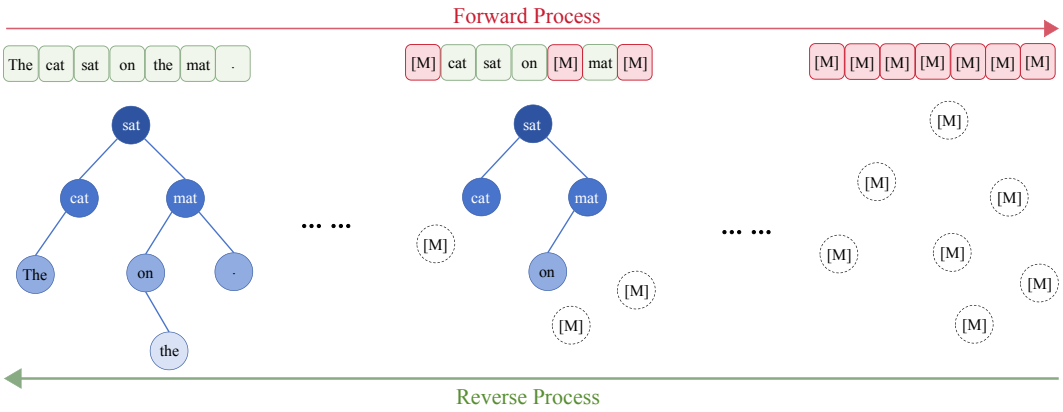

Figure 1: Overview of Ordered-DLM. The forward process uses the learned hierarchy to retain structurally central tokens longer, while the reverse process predicts hierarchy scores from the current masked sequence and uses them to prioritize token reveals.

3 Hierarchy-Induced Masked Diffusion

3.1 Overview

Figure 1 summarizes the forward masking and reverse denoising processes. To connect syntactic hierarchy to masking scheduler, we first induce a hierarchy score $H_i(\mathbf{x})$ from the relative order of the token heights; this score measures how persistently token i acts as a structural anchor across nested constituents. Using the order-expressive parameterization of Hong et al. (2026b), we map this induced hierarchy score to the probability that token i remains unmasked:

$$\alpha_\phi^{(i)}(\mathbf{x}, t) = 1 - t^{H_i(\mathbf{x})}, \tag{1}$$

where $H_i(\mathbf{x}) > 0$ is a differentiable hierarchy score for token x_i. For $t \in (0, 1)$, a larger $H_i(\mathbf{x})$ makes $t^{H_i(\mathbf{x})}$ smaller and therefore keeps token i unmasked with higher probability. The main technical problem is thus how to compute $H_i(\mathbf{x})$ from syntactic heights in a way that preserves hierarchical ordering and remains differentiable for masked diffusion training.

3.2 Inducing Hierarchy from Syntactic Heights

We first present a discrete construction that shows how a hierarchy score $\widetilde{H}_i$ can be computed from syntactic heights. This construction is not yet differentiable, it rather gives an intuitive target that the differentiable relaxation in the next subsection will approximate.

Let $\mathbf{x} = (x_1, \ldots, x_n)$ be a token sequence, and assume for now that each token x_i has a distinct scalar height $h_i \in \mathbb{R}$. The relative order of these heights defines a max-Cartesian tree: within any interval, the token with the largest height is selected as the interval root, and the same rule is recursively applied to the left and right subintervals. Thus, scalar height comparisons are converted into a discrete max-Cartesian-tree hierarchy over spans, as illustrated in Figure 2. We next rewrite this tree rule algebraically. For two positions i and j, define the pairwise dominance indicator

$$\widetilde{\alpha}_{ij} = \begin{cases} 1, & i = j, \\ \mathbf{1}\{h_i > h_j\}, & i \neq j. \end{cases} \tag{2}$$

Thus $\widetilde{\alpha}_{ij} = 1$ means that position i can dominate position j in the max-height ordering: either the two positions are identical, or x_i has a larger height than x_j. For the interval between positions i and j, namely $[\min(i, j), \max(i, j)]$, we then define

$$\widetilde{g}_{ij} = \prod_{k=\min(i,j)}^{\max(i,j)} \widetilde{\alpha}_{ik}. \tag{3}$$

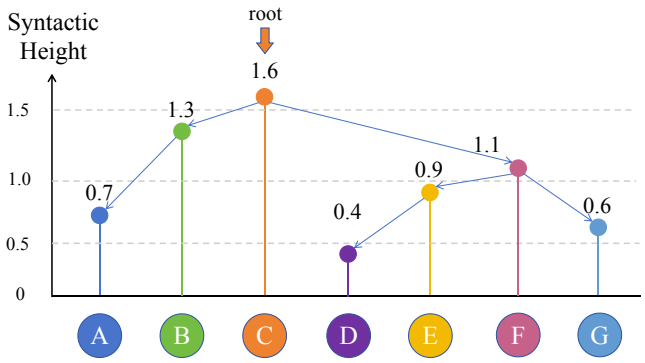

Figure 2: Constructing a max-Cartesian tree from syntactic heights. Each token is assigned a syntactic height; the token with the largest height becomes the root, and the same rule is applied recursively to the remaining left and right spans. The resulting tree organizes tokens into a hierarchy, with higher-height tokens serving as structural anchors for larger spans.

Because every factor is binary, $\widetilde{g}_{ij} = 1$ if and only if $h_i > h_k$ for every $k \in [\min(i,j), \max(i,j)]$ with $k \neq i$. In other words, $\widetilde{g}_{ij}$ records whether token i serves as the maximum-height root of the whole interval between i and j. If any token inside the interval has a larger height than h_i, that token blocks i, and the product becomes zero.

The hard hierarchy score measures the structural span rooted at token i in the max-Cartesian tree, with a length-dependent normalization:

$$\widetilde{H}_i = \frac{\sum_{1 \leq j \leq n} \widetilde{g}_{ij}}{\log_2 n}. \tag{4}$$

The numerator counts how many endpoint-defined spans $[\min(i,j), \max(i,j)]$, indexed by j, still choose i as their maximum-height root. Therefore, a token receives a large $\widetilde{H}_i$ when it remains the root after the interval is expanded over many neighboring tokens, which indicates that it acts as a persistent structural anchor rather than a local lexical detail. The factor $\log_2 n$ provides a partial rescaling across sentence lengths. It does not change the relative order of tokens within the same sentence, but it tempers the growth of the hierarchy scores as sentence length increases.

3.3 Differentiable Height-to-Hierarchy Relaxation

The hard construction above defines the desired hierarchy, but it is not directly trainable because it uses indicators and binary products. We therefore predict continuous syntactic heights from the input sequence,

$$h_i(\mathbf{x}) = g_\phi^{(i)}(\mathbf{x}), \qquad i \in [n], \tag{5}$$

where g_ϕ is a neural height predictor.

We relax the hard comparison in Eq. 2 as

$$\alpha_{ij} = \begin{cases} 1, & i = j, \\ \frac{1+\mathrm{hardtanh}(h_i(\mathbf{x}) - h_j(\mathbf{x}))}{2}, & i \neq j, \end{cases} \tag{6}$$

The normalized hardtanh maps the pairwise difference to $[0,1]$. It is a piecewise-linear differentiable relaxation of the hard indicator $\mathbf{1}\{h_i > h_j\}$ and provides gradients to the height predictor wherever the hardtanh is not saturated.

We then replace the hard interval-root indicator with a soft interval-survival score,

$$g_{ij} = \prod_{k=\min(i,j)}^{\max(i,j)} \alpha_{ik}, \tag{7}$$

and define the differentiable hierarchy score as

$$H_i(\mathbf{x}) = \frac{\sum_{1 \le j \le n} g_{ij}}{\log_2 n}. \tag{8}$$

This score is a smooth counterpart of Eq. 4. It preserves the max-Cartesian-tree intuition while making the full path from syntactic heights to the OeMDM scheduler differentiable.

3.4 Hierarchy-Induced Masked Diffusion Training

Given the differentiable hierarchy score, the forward noising process follows the OeMDM absorbing-state parameterization. For each position i, we sample

$$q_{\alpha_\phi}(z_t^{(i)} \mid \mathbf{x}) = \mathrm{Cat}\Big(\alpha_\phi^{(i)}(\mathbf{x}, t)\, x_i + \big(1 - \alpha_\phi^{(i)}(\mathbf{x}, t)\big)\, \texttt{[MASK]}\Big), \tag{9}$$

with $\alpha_\phi^{(i)}(\mathbf{x}, t)$ defined in Eq. 1. The clean sequence $\mathbf{x}$ is available during forward noising, but it is unavailable during reverse-time generation. Following OeMDM, we therefore use a reverse scheduler conditioned on the current masked sequence $\mathbf{z}_t$. Let $\widehat{h}_i(\mathbf{z}_t) = g_\psi^{(i)}(\mathbf{z}_t)$, compute $H_i(\mathbf{z}_t)$ using the same soft height-to-hierarchy rules in Eqs. 6–8, and set

$$\hat{\alpha}_\psi^{(i)}(\mathbf{z}_t, t) = 1 - t^{H_i(\mathbf{z}_t)}. \tag{10}$$

The reverse scheduler learns to infer a compatible hierarchy from partial context and therefore predicts which masked positions should be denoised earlier.

We train the hierarchy-induced schedulers with the OeMDM NELBO of Hong et al. (2026b). For any scheduler of the form $\alpha^{(i)}(u, t) = 1 - t^{H_i(u)}$, the OeMDM velocity is

$$A^{(i)}(u, t) = -\frac{\partial_t \alpha^{(i)}(u, t)}{1 - \alpha^{(i)}(u, t)} = \frac{H_i(u)}{t}. \tag{11}$$

Thus the forward and reverse velocities are

$$A_\phi^{(i)}(\mathbf{x}, t) = \frac{H_i(\mathbf{x})}{t}, \qquad \hat{A}_\psi^{(i)}(\mathbf{z}_t, t) = \frac{H_i(\mathbf{z}_t)}{t}. \tag{12}$$

Let $\mathbf{x}_\theta(\mathbf{z}_t, t)$ be the masked diffusion backbone that predicts clean tokens from the partially masked sequence. The training objective is

$$\begin{aligned} \mathcal{L} = \int_0^1 \mathbb{E}_{q_{\alpha_\phi}} \Bigg[\sum_{i=1}^{n} \mathbf{1}\{z_t^{(i)} = \texttt{[MASK]}\} \Big\{ & - A_\phi^{(i)}(\mathbf{x}, t) \log p_\theta(x_i \mid \mathbf{z}_t, t) \\ & + A_\phi^{(i)}(\mathbf{x}, t) \Big(\log A_\phi^{(i)}(\mathbf{x}, t) - \log \hat{A}_\psi^{(i)}(\mathbf{z}_t, t) \Big) \\ & - \Big(A_\phi^{(i)}(\mathbf{x}, t) - \hat{A}_\psi^{(i)}(\mathbf{z}_t, t) \Big) \Big\} \Bigg] dt. \end{aligned} \tag{13}$$

3.5 Blockwise Hierarchy Scoring

For long sequences, evaluating the hierarchy over every pair of token positions is memory intensive. In particular, the full construction requires pairwise quantities α_{ij} and g_{ij} for n^2 position pairs. This is costly for OpenWebText, where each training sequence has length $n = 1024$. We therefore use *Blockwise Hierarchy Scoring* (BHS), which computes local hierarchy scores inside short contiguous blocks.

Given a block length L, partition the sequence into $B = \lceil n/L \rceil$ blocks. The b-th block is

$$\mathcal{B}_b = \{(b-1)L + 1, \ldots, \min(bL, n)\}, \qquad b \in [B]. \tag{14}$$

For a position i, let $b(i)$ be the index of its block and let $L_{b(i)} = |\mathcal{B}_{b(i)}|$ be that block's length. We retain the same pairwise comparison and interval-survival rule from Eqs. 6–7, but only compare i with endpoints j in its own block. The BHS score is

$$H_i^{\mathrm{BHS}}(\mathbf{x}) = \frac{\sum_{j \in \mathcal{B}_{b(i)}} g_{ij}}{\log_2\big(\max\{2, L_{b(i)}\}\big)}. \tag{15}$$

Thus, a token receives a large score when it remains the dominant token as its local block span expands. The final, possibly shorter block is handled by its own length; the $\max\{2, \cdot\}$ term avoids a zero denominator for a singleton final block.

During training, we replace $H_i(\mathbf{x})$ in the forward scheduler with $H_i^{\mathrm{BHS}}(\mathbf{x})$. During generation, the reverse scheduler analogously uses $H_i^{\mathrm{BHS}}(\mathbf{z}_t)$, computed from the heights predicted on the current masked sequence. BHS does not construct cross-block pairwise scores: global dependencies are still modeled by the bidirectional diffusion backbone, while the hierarchy scheduler provides a local hierarchy-induced ordering within each block.

The full construction materializes $O(n^2)$ pairwise scores. With BHS, there are at most $BL^2 = O(nL)$ within-block pairs, reducing the pairwise memory requirement from quadratic to linear in sequence length when L is fixed. BHS therefore makes hierarchy-induced scheduling practical for length-1024 OWT sequences while preserving the same height-to-hierarchy mechanism locally.

4 EXPERIMENTS

4.1 SETUP

Experimental Settings. We evaluate Ordered-DLM following the experimental protocol of prior works (Sahoo et al., 2025; 2024; Hong et al., 2026b). For likelihood evaluation, we train and evaluate on LM1B with and without sentence packing, and on OpenWebText (OWT). We report test perplexity (PPL; $\downarrow$), where diffusion-model perplexities correspond to likelihood upper bounds. To test out-of-domain generalization, we evaluate the OWT-trained model in a zero-shot setting on 7 other datasets. For generation quality, we report generative PPL on OWT using a pretrained GPT-2 Large evaluator. We report both high-precision and low-precision sampling results: high-precision sampling uses `float64` arithmetic, whereas low-precision sampling uses `float32` arithmetic. We follow the training hyperparameters of MDLM (Sahoo et al., 2024). Unless otherwise specified, all diffusion models are trained for 1M steps with batch size 512. For details about model architecture, please refer to Appendix B.

4.2 MAIN RESULTS

Likelihood evaluation. On LM1B and OWT (Table 1), Ordered-DLM consistently improves over prior discrete diffusion models. Ordered-DLM achieves 22.65 PPL on LM1B¶, 24.57 on LM1B, and 19.28 on OWT. Compared with MDLM, which achieves 27.0, 31.8, and 23.2 on the same three benchmarks, Ordered-DLM is better on all three datasets. Compared with the learned-order LoMDM baseline, which achieves 25.4, 27.2, and 20.4, Ordered-DLM is again better on all three benchmarks. We also observe that Ordered-DLM outperforms BD3LM even when BD3LM uses a small block size $L' = 4$, which injects a strong block-wise left-to-right bias, and improves over GenMD4 on every reported benchmark. These results suggest that inducing the masking order from syntactic hierarchy provides a stronger sequence-dependent training signal than generic learned ordering.

Zero-shot likelihood evaluation. We measure zero-shot generalization by evaluating the model trained on OWT on seven additional datasets (Table 2). Ordered-DLM achieves 48.55 PPL on PTB, 24.41 on Wikitext, 44.52 on LM1B, 31.84 on Lambada, 46.18 on AG News, 36.88 on PubMed, and 29.98 on ArXiv. These values are better than MDLM and LoMDM on all seven datasets, and they also outperform the autoregressive Transformer on every zero-shot benchmark in Table 2. The largest gains appear on PTB and LM1B, where Ordered-DLM achieves 48.55 and 44.52, compared with LoMDM's 80.40 and 61.19. This broad improvement suggests that hierarchy-induced noising does not merely overfit the training distribution, but learns an ordering that transfers across domains.

Generative perplexity. Finally, we evaluate generation quality on OWT using generative PPL computed by a pretrained GPT-2 Large evaluator. As shown in Table 3, Ordered-DLM consistently outperforms both MDLM and LoMDM under low-precision sampling. Notably, Ordered-DLM at 256 NFE already slightly surpasses LoMDM at 512 NFE (47.45 vs. 48.29), and the advantage persists as the number of denoising steps increases.

Table 1: Test perplexity (PPL; ↓) on LM1B and OpenWebText. Best diffusion results are bolded. Diffusion-model PPL values are likelihood upper bounds; ¶ marks LM1B without sentence packing; L' is the BD3LM block length.

		LM1B¶	LM1B	OWT
Autoregressive				
Transformer		22.3	22.8†	17.5
Diffusion (Uniform-state / Gaussian)				
D3PM Uniform (Austin et al., 2021)		-	137.9	-
Diffusion-LM (Li et al., 2022)		-	118.6	-
SEDD Uniform (Lou et al., 2024)		40.3	-	29.7
UDLM (Schiff et al., 2025)		31.3	36.7	27.4
Duo (Sahoo et al., 2025)		29.9	33.7	25.2
Diffusion (Absorbing state)				
BERT-Mouth (Wang et al., 2019a)		-	142.9	-
D3PM Absorb (Austin et al., 2021)		-	76.9	-
DiffusionBert (He et al., 2023)		-	63.8	-
SEDD Absorb (Lou et al., 2024)		32.7	-	24.1
MDLM (Sahoo et al., 2024)		27.0	31.8	23.2
Autoregressive + Diffusion				
BD3LM (Arriola et al., 2025)	L'=16	-	30.6	22.3
	L'=8	-	29.8	21.7
	L'=4	-	28.2	20.7
Diffusion (Absorbing state + Learnable scheduler)				
GenMD4 (Shi et al., 2024)		26.9‡	30.0‡	21.8≥
LoMDM (Hong et al., 2026b)		25.4	27.2	20.4
Ordered-DLM		**22.65**	**24.57**	**19.28**

Table 2: Zero-shot perplexity (PPL; ↓) of OWT-trained models on seven held-out datasets. Diffusion values are likelihood upper bounds; best diffusion results are bolded, and diffusion results better than the autoregressive Transformer are underlined.

	PTB	Wikitext	LM1B	Lambada	AG News	Pubmed	Arxiv
Autoregressive							
Transformer	82.05	25.75	51.25	51.28	52.09	49.01	41.73
Diffusion (Uniform-state / Gaussian)							
SEDD Uniform	105.51	41.10	82.62	57.29	82.64	55.89	50.86
Plaid	142.60	50.86	91.12	57.28	-	-	-
UDLM	112.82	39.42	77.59	53.57	80.96	50.98	44.08
Duo	89.35	33.57	73.86	<u>49.78</u>	67.81	<u>44.48</u>	<u>40.39</u>
Diffusion (Absorbing state)							
SEDD Absorb	100.09	34.28	68.20	<u>49.86</u>	62.09	<u>44.53</u>	<u>38.48</u>
D3PM Absorb	200.82	50.86	138.92	93.47	-	-	-
MDLM	95.26	32.83	67.01	<u>47.52</u>	61.15	<u>41.89</u>	<u>37.37</u>
Autoregressive + Diffusion							
BD3LM† ($L' = 4$)	96.81	31.31	60.88	50.03	61.67	<u>42.52</u>	<u>39.20</u>
Diffusion (Absorbing state + Learnable scheduler)							
LoMDM	<u>80.40</u>	27.82	61.19	<u>36.32</u>	53.53	<u>37.73</u>	<u>32.88</u>
Ordered-DLM	**<u>48.55</u>**	**24.41**	**44.52**	**<u>31.84</u>**	**46.18**	**<u>36.88</u>**	**<u>29.98</u>**

This advantage is even more pronounced in the low-NFE regime under high-precision sampling (Table 4). Notably, Ordered-DLM reaches 90.40 PPL with only 64 NFE, lower than the best MDLM result reported across all sampling budgets in the table. Together, these results suggest that our syntactic hierarchy-induced scheduling is particularly beneficial when the denoising budget is limited: by prioritizing structurally informative tokens early, the reverse process makes more effective use of each parallel decoding step.

Table 3: Low-precision generative PPL (↓) on OWT. Scores are computed with a pre-trained GPT-2 Large evaluator on 256 generated samples of length 1024.

#NFE	128	256	512	1024
MDLM	116.71	79.43	55.50	42.56
LoMDM	92.78	73.98	48.29	38.87
Ordered-DLM	**57.19**	**47.45**	**41.69**	**35.36**

Table 4: High-precision sampling results on OWT. We report generative PPL (↓) and entropy (↑) across sampling steps.

#NFE	Ordered-DLM		MDLM	
	Gen PPL (↓)	Entropy (↑)	Gen PPL (↓)	Entropy (↑)
1024	66.89	5.47	104.85	5.63
512	74.88	5.53	104.43	5.63
256	79.59	5.55	112.70	5.66
128	81.36	5.55	120.77	5.67
64	90.40	5.55	143.88	5.70
32	106.63	5.54	196.79	5.75
16	145.00	5.55	343.33	5.81
8	236.64	5.48	830.82	5.91

4.3 Probing the Contextual Value of the Generation Order

The central mechanism of Ordered-DLM is to reveal tokens that serve as useful context before resolving local details. We test this mechanism directly with a functional probe: when the ordering selects a masked token, how much does revealing its ground-truth value reduce uncertainty about the *other* masked tokens?

Contextual Entropy Drop. Let C_t be a partially masked context at diffusion time t, and let $M_{t,i}$ be the valid masked positions other than a selected position i. For any position j, denote the denoiser's predictive entropy by

$$e_j(C) = -\sum_{v \in \mathcal{V}} p_\theta(v \mid C, t) \log p_\theta(v \mid C, t). \tag{16}$$

We define the *Contextual Entropy Drop* (CED) of revealing the ground-truth token x_i as

$$\mathrm{CED}(i; C_t) = \frac{1}{|M_{t,i}|} \sum_{j \in M_{t,i}} \left[e_j(C_t) - e_j\big(C_t \cup \{x_i\}\big) \right]. \tag{17}$$

A larger CED means that the revealed token makes the remaining masked sequence easier to predict. Excluding i itself is essential: the probe measures the token's contextual value, rather than how easy that token is to predict.

Priority Context Value. Let $i_r^{(m)}(C_t)$ be the r-th lexical position selected by ordering strategy m, where $r = 0$ is selected first. The *Priority Context Value* (PCV) of its first K selections is the average CED:

$$\mathrm{PCV}_K^{(m)}(C_t) = \frac{1}{K} \sum_{r=0}^{K-1} \mathrm{CED}\Big(i_r^{(m)}(C_t); C_t\Big). \tag{18}$$

This metric asks whether an ordering places contextually informative tokens near the beginning of denoising. We use $K \in \{4, 8\}$, filter to lexical tokens, and compare Ordered-DLM with LoMDM and MDLM. Ordered-DLM and LoMDM use their learned scheduler scores, whereas MDLM reveals the lowest-entropy candidates first. At each timestep, we evaluate 64 shared OWT contexts.

Probe results. The probe evaluates whether Ordered-DLM performs the intended form of contextual planning: revealing tokens that are useful for resolving the remaining sequence before filling in local details. Table 5 reports the absolute PCV values for both $K = 4$ and $K = 8$. Ordered-DLM consistently obtains positive PCV values, whereas LoMDM remains near zero; its values also exceed those of MDLM confidence ordering at most timesteps. These results suggest that the hierarchy-induced order prioritizes contextually informative tokens, providing functional evidence for the proposed context-first generation strategy.

4.4 Conditional Generation

We evaluate conditional generation following Lou et al. (2024). Table 6 compares Ordered-DLM with LoMDM and the reference results reported by Lou et al. (2024). Ordered-DLM obtains a

	$K = 4$			$K = 8$		
t	Ordered-DLM	LoMDM	MDLM	Ordered-DLM	LoMDM	MDLM
0.9	1.693e-3	-2.000e-6	**3.070e-3**	2.180e-3	-2.000e-6	**3.022e-3**
0.8	**2.535e-3**	-1.000e-6	1.588e-3	2.479e-3	-1.000e-6	**2.757e-3**
0.7	**3.742e-3**	-2.000e-6	3.590e-4	**3.785e-3**	-1.000e-6	3.420e-4
0.6	**2.985e-3**	-2.000e-6	2.060e-4	**2.612e-3**	-2.000e-6	1.780e-4
0.5	**2.390e-3**	-2.000e-6	1.350e-4	**2.635e-3**	-2.000e-6	2.820e-4
0.4	**2.614e-3**	-3.000e-6	1.600e-5	**2.948e-3**	-3.000e-6	1.630e-4
0.3	**1.924e-3**	-2.000e-6	1.230e-4	**2.117e-3**	-2.000e-6	1.740e-4
0.2	**2.979e-3**	-8.000e-6	2.610e-4	**3.188e-3**	-7.000e-6	2.180e-4
0.1	**1.563e-3**	-3.000e-6	-2.240e-4	**2.247e-3**	-3.000e-6	-1.700e-5

Table 5: Priority Context Value (PCV) for the first K lexical tokens selected at each diffusion timestep t. PCV is the mean contextual entropy drop caused by revealing the selected tokens, where the entropy is measured on the other masked positions; thus, larger positive values indicate that an ordering reveals tokens that make the remaining sequence easier to predict (better contextual planning). We compare the proposed syntactic hierarchy-induced scheduler (Ordered-DLM), the learned-order LoMDM scheduler, and MDLM confidence ordering. K denotes the number of earliest selections included in the average.

MAUVE score of 0.9771 without annealing, compared with 0.9751 for LoMDM under the same setting. Both learned-order masked diffusion models exceed the reported SEDD standard and infilling variants, indicating that hierarchy-induced scheduling preserves conditional generation quality while improving the match to the reference continuation distribution.

Table 6: Conditional-generation MAUVE (↑). We follow the evaluation protocol of Lou et al. (2024). Baseline results above the midrule are reported by Lou et al. (2024); Ordered-DLM and LoMDM are evaluated in the same conditional-generation setting.

Method	Annealing	MAUVE (↑)
GPT-2	Nucleus-0.95	0.955
SSD-LM	Logit Threshold-0.95	0.919
SSD-LM	None	0.312
SEDD Standard	None	0.957
SEDD Infill	None	0.942
LoMDM	None	0.9751
Ordered-DLM	None	**0.9771**

5 Conclusion

We introduced Ordered-DLM, a masked diffusion language model that explores how token generation order can support contextual planning: revealing tokens early when they provide useful context for predicting the remaining masked positions. To realize this idea, Ordered-DLM uses syntactic hierarchy as a learned structural signal. It predicts continuous syntactic heights, converts their relative ordering into differentiable hierarchy scores, and uses these scores to guide both forward masking and reverse denoising. The hierarchy is learned jointly with the diffusion model, requiring neither an external parser nor syntactic annotations. The contextual-planning probe indicates that prioritized tokens tend to reduce uncertainty over the remaining masked sequence, inducing a contextually informative generation order. The resulting schedule improves language-modeling and generation quality. Ordered-DLM achieves lower perplexity on LM1B and OpenWebText, stronger zero-shot performance across seven datasets, and lower generative perplexity across sampling budgets. Overall, these results suggest that syntactic hierarchy can provide a useful structural signal for learning contextually informative reveal orders in masked diffusion language models.

AI USE STATEMENT

We used generative AI tools to assist with the development of theoretical models and conceptual frameworks, provide feedback on research methodology and experimental design, assist with method implementation, translation, and interpretation of experimental results. We also used generative AI tools for experimental code development, manuscript organization and readability, LaTeX troubleshooting, and table formatting. We did not use generative AI tools to generate synthetic datasets, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in writing proofs, propose or refine hypotheses, clean or reformat datasets, or conduct qualitative or thematic data analysis. All AI-assisted outputs were reviewed by the authors, who take full responsibility for the final manuscript, code, scientific claims, experimental results, and references.

ETHICS STATEMENT

This work studies masked diffusion language models using existing text datasets. Such models may reproduce biases or harmful content present in their training data and could be misused to generate misleading text. Our experiments evaluate modeling quality rather than deployment safety; practical applications would require additional safety evaluation and appropriate safeguards.

REPRODUCIBILITY STATEMENT

We describe our method, model architecture, datasets, and evaluation protocols in Sections 3 and 4, with additional details in Appendix B. Our training hyperparameters follow MDLM (Sahoo et al., 2024).

REFERENCES

Marianne Arriola, Aaron Gokaslan, Justin Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations*, volume 2025, pp. 50726–50753, 2025.

Hikaru Asano, Tadashi Kozuno, Kuniaki Saito, and Yukino Baba. Where-to-unmask: Ground-truth-guided unmasking order learning for masked diffusion language models. *arXiv preprint arXiv:2602.09501*, 2026.

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

Kwan Ho Ryan Chan, Yuyan Ge, Edgar Dobriban, Hamed Hassani, and Rene Vidal. Conformal information pursuit for interactively guiding large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 90328–90377. Curran Associates, Inc., 2025. doi: 10.52202/085713-3021. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/823a3d2cf462fb815978314023e48f65-Paper-Conference.pdf.

Mengyi DENG, Zhiwei Li, Xin Li, Tingyu ZHU, Ying Zhao, Zhijiang Guo, and Wei Wang. Uncertainty-aware clarification in LLM agents with information gain. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=7N2ydbDahV.

Andrew Drozdov, Patrick Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. Unsupervised latent tree induction with deep inside-outside recursive auto-encoders. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *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. 1129–1141, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1116. URL https://aclanthology.org/N19-1116/.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6112–6121, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1633. URL https://aclanthology.org/D19-1633/.

Xu Guo. Measuring reasoning utility in llms via conditional entropy reduction. *arXiv preprint arXiv:2508.20395*, 2025.

Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuan-Jing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 4521–4534, 2023.

Chunsan Hong, Seonho An, Min-Soo Kim, and Jong Chul YE. Improving discrete diffusion unmasking policies beyond explicit reference policies. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 108226–108260, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/b0506debbf49e31d25690fbd1e69cd2f-Paper-Conference.pdf.

Chunsan Hong, Sanghyun Lee, and Jong Chul Ye. Unifying masked diffusion models with various generation orders and beyond. *arXiv preprint arXiv:2602.02112*, 2026b.

Senkang Hu, Yong Dai, Yuzhi Zhao, Yihang Tao, Yu Guo, Zhengru Fang, Sam Kwong, and Yuguang Fang. Optimizing agentic reasoning with retrieval via synthetic semantic information gain reward. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=mUU7k4bN97.

Metod Jazbec, Theo X. Olausson, Louis Béthune, Pierre Ablin, Michael Kirchhof, Joao Monteiro, Victor Guilherme Turrisi da Costa, Jason Ramapuram, and marco cuturi. Learning unmasking policies for diffusion language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=F9NDKf5oPy.

Yoon Kim, Chris Dyer, and Alexander Rush. Compound probabilistic context-free grammars for grammar induction. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2369–2385, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1228. URL https://aclanthology.org/P19-1228/.

Yoon Kim, Alexander Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. Unsupervised recurrent neural network grammars. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *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. 1105–1117, Minneapolis, Minnesota, June 2019b. Association for Computational Linguistics. doi: 10.18653/v1/N19-1114. URL https://aclanthology.org/N19-1114/.

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

Zhihan Liu, Hao Hu, Shenao Zhang, Hongyi Guo, Shuqi Ke, Boyi Liu, and Zhaoran Wang. Reason for future, act for now: A principled architecture for autonomous LLM agents. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 31186–31261. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/liu24ab.html.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Hongyin Luo, Lan Jiang, Yonatan Belinkov, and James Glass. Improving neural language models by segmenting, attending, and predicting the future. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1483–1493, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1144. URL https://aclanthology.org/P19-1144/.

Linrui Ma, Yufei Cui, Kai Han, and Yunhe Wang. Mask is what dllm needs: A masked data training paradigm for diffusion llms. *arXiv preprint arXiv:2603.15803*, 2026.

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

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T Chiu, and Volodymyr Kuleshov. The diffusion duality. In *International Conference on Machine Learning*, pp. 52584–52619. PMLR, 2025.

Yair Schiff, Subham Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-Torre, Bernardo Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 43776–43821, 2025.

Yikang Shen, Zhouhan Lin, Chin-wei Huang, and Aaron Courville. Neural language modeling by jointly learning syntax and lexicon. In *International Conference on Learning Representations*, 2018a.

Yikang Shen, Zhouhan Lin, Athul Paul Jacob, Alessandro Sordoni, Aaron Courville, and Yoshua Bengio. Straight to the tree: Constituency parsing with neural syntactic distance. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1171–1180, Melbourne, Australia, July 2018b. Association for Computational Linguistics. doi: 10.18653/v1/P18-1108. URL https://aclanthology.org/P18-1108/.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1l6qiR5F7.

Yikang Shen, Yi Tay, Che Zheng, Dara Bahri, Donald Metzler, and Aaron Courville. StructFormer: Joint unsupervised induction of dependency and constituency structure from masked language modeling. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7196–7209, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.559. URL https://aclanthology.org/2021.acl-long.559/.

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

Jean-Francois Ton, Muhammad Faaiz Taufiq, and Yang Liu. Understanding chain-of-thought in LLMs through information theory. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 59784–59811. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/ton25a.html.

Alex Wang, Kyunghyun Cho, and CIFAR Azrieli Global Scholar. Bert has a mouth, and it must speak: Bert as a markov random field language model. *NAACL HLT 2019*, pp. 30, 2019a.

Renzhi Wang, Jing Li, and Piji Li. InfoDiffusion: Information entropy aware diffusion process for non-autoregressive text generation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 13757–13770,

Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.919. URL https://aclanthology.org/2023.findings-emnlp.919/.

Yaushian Wang, Hung-Yi Lee, and Yun-Nung Chen. Tree transformer: Integrating tree structures into self-attention. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1061–1070, Hong Kong, China, November 2019b. Association for Computational Linguistics. doi: 10.18653/v1/D19-1098. URL https://aclanthology.org/D19-1098/.

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

Jiawei Xu, Minghui Liu, Aakriti Agrawal, Yifan Chen, and Furong Huang. Scheduling thoughts: Learning the order of thought in diffusion language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=j4UhHaxCpq.

Yiming Zeng, Jinghan Cao, Zexin Li, Yiming Chen, Tao Ren, Zhuochun Li, Dawei Xiang, Xidong Wu, Shangqian Gao, and Tingting Yu. Treediff: Ast-guided code generation with diffusion llms. *arXiv preprint arXiv:2508.01473*, 2025.

A Ablation Study

A.1 Ablation Study

Direction of the pairwise height comparison. Our hierarchy construction uses the pairwise comparison $h_i - h_j$ in Eq. 6, so that a larger height makes a token more likely to dominate any span containing it. The resulting hierarchy score also depends on the heights and positions of the surrounding tokens. A natural concern is whether this direction is arbitrary. To test this, we train two Ordered-DLM checkpoints on LM1B wrapped data with opposite pairwise ordering conventions: the default forward comparison $h_i - h_j$ and the reversed comparison $h_j - h_i$. We then evaluate both checkpoints on the same validation examples and compare their predicted token heights, denoted h_i^{fwd} and h_i^{rev}, over 262K valid token positions. As shown in Table 7, reversing the comparison almost exactly reverses the orientation of the learned height scale: the Pearson and Spearman correlations between h_i^{rev} and h_i^{fwd} are -0.8838 and -0.9848, respectively. The linear fit $h_i^{\mathrm{rev}} \approx -0.98 h_i^{\mathrm{fwd}} + 2.66$ further shows that the two checkpoints learn oppositely oriented height scales, up to a small affine shift and scale mismatch. This result is consistent with sign equivariance. If the reversed convention learns heights that are the negatives of those learned by the forward convention, then its reversed comparison yields the same pairwise dominance values and hence the same hierarchy scores. The observed height reversal is compatible with this symmetry, while the small scale mismatch can alter the magnitude of the soft dominance values.

Table 7: Ablation on the direction of the pairwise height comparison. We compare token-level predicted heights from two checkpoints trained with the forward comparison $h_i - h_j$ and the reversed comparison $h_j - h_i$ on the same LM1B wrapped validation examples.

Metric	Value
Pearson corr. $(h_i^{\mathrm{rev}}, h_i^{\mathrm{fwd}})$	-0.8838
Spearman corr. $(h_i^{\mathrm{rev}}, h_i^{\mathrm{fwd}})$	-0.9848
Linear fit $h_i^{\mathrm{rev}} = a h_i^{\mathrm{fwd}} + b$	$a = -0.9802,\ b = 2.6599$

B MODEL ARCHITECTURE

Following MDLM and LoMDM (Sahoo et al., 2024; Hong et al., 2026b), we use a time-agnostic Transformer as the diffusion backbone: the backbone receives the masked token sequence but no explicit diffusion-time embedding. The forward scheduler g_ϕ predicts one height per position from the input token embeddings. It consists of two additional Transformer layers on top of the embedding layer, followed by a scalar prediction head. The reverse scheduler g_ψ predicts heights from the current masked sequence. It adds one Transformer layer on top of the diffusion backbone's token representations and applies a scalar prediction head at each position. Thus, the forward scheduler learns syntactic heights from the clean sequence available during training, whereas the reverse scheduler infers a compatible hierarchy from the backbone representation available during generation.