# DA-DLM: Explicitly Modeling Token Dependencies in Diffusion Language Models

Anonymous ACL submission

## Abstract

Diffusion Language Models (DLMs) generate text by iteratively denoising a masked sequence, independently predicting multiple tokens at each step. This conditional independence discards inter-token dependencies and degrades coherence—an issue that parallels the multi-modality problem in Non-Autoregressive Translation (NAT). Drawing on the Directed Acyclic Transformer (DAT), which tackles this problem in NAT via a Directed Acyclic Graph (DAG), we propose DA-DLM, a model that adapts DAG-based dependency modeling to DLMs' iterative setting through a position-oriented DAG design. The position-oriented DAG binds node groups to fixed output positions so that tokens fixed in earlier steps anchor neighboring predictions via learned transitions, and evolves with denoising to focus on remaining uncertainty as anchors accumulate. On language modeling, open-ended generation, and summarization, DA-DLM consistently outperforms Block Diffusion, especially under fewer denoising steps, and matches autoregressive models while preserving the parallel generation advantage.

## 1 Introduction

Diffusion Language Models (DLMs) (Nie et al., 2025; Bie et al., 2025, 2026) have emerged as a promising paradigm for text generation. Most existing DLMs adopt a masked diffusion framework (Sahoo et al., 2024), with Block Diffusion (BD3LM) (Arriola et al., 2025) extending it to a block-wise setting. Starting from a fully masked sequence, they progressively recover the target text over multiple denoising steps, each predicting a set of tokens in parallel. Unlike autoregressive (AR) models that generate tokens sequentially, DLMs are not bound by a fixed generation order and can iteratively refine the sequence with bi-directional context, offering the potential for high-quality and high-speed parallel generation.

However, realizing this potential remains challenging in practice. At each denoising step, DLMs only model token-wise marginals, predicting multiple token independently without capturing dependencies among these simultaneously generated tokens. This conditional independence assumption is a key structural limitation: without coordination, tokens predicted within the same step lack coherence at the sequence level. The problem further intensifies when fewer denoising steps are used to improve efficiency, as each step must predict more tokens in parallel (Liu et al., 2025; Wu et al., 2025; Li et al., 2026).

This type of per-position independent prediction method, along with its associated problem, is not unique to DLMs. In machine translation, Non-Autoregressive Transformers (NATs) (Gu et al., 2018) also independently predict multiple target tokens given a masked or incomplete sequence. It is well-known that NATs suffer from the multi-modality problem (Gu et al., 2018; Huang et al., 2022a): per-position independent predictions mix fragments from different valid translations, mirroring the coherence issue observed in DLMs.

The connection between DLMs and NATs suggests that techniques for handling the multi-modality problem in the latter can inform solutions to the associated issue in the former. In this work, we focus on a particular NAT technique, the Directed Acyclic Transformer (DAT) (Huang et al., 2022b), which tackles this problem by explicitly modeling token dependencies. It generates a Directed Acyclic Graph (DAG) that represents possible tokens and probabilistic transitions between them. By finding a proper path through the DAG, DAT predicts all output tokens in a single step while ensuring they are mutually consistent. However, directly transplanting DAT's mechanism into DLMs is non-trivial. In DLMs' iterative denoising, each step resolves a portion of the remaining uncertainty, relying on tokens fixed in earlier steps

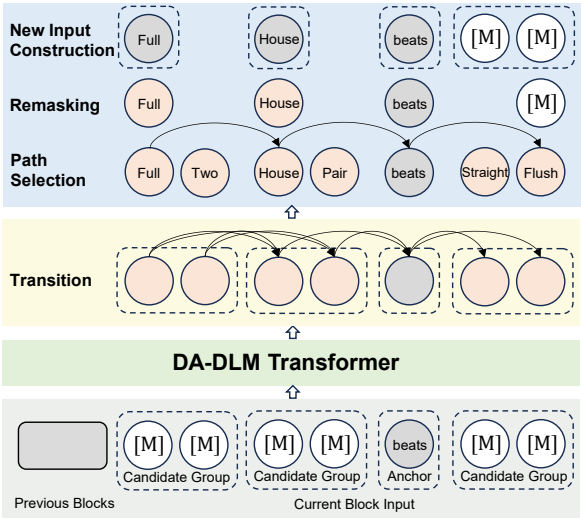


Figure 1: Overview of one denoising step in DA-DLM. Given the current block input with one anchor and three candidate groups (bottom), the model produces per-node emissions and inter-group transitions (middle), selects a path, remasks, and yields a new DAG for the next step.

to effectively anchor subsequent predictions (Sun et al., 2025; Hong et al., 2026), yet DAT's dependency structure—a large DAG with unrestricted transitions—is tailored to single-step generation and weakens the ability of previously fixed tokens to guide subsequent predictions. Without effective anchoring, successive denoising steps drift apart rather than converge toward a coherent output.

In this work, we propose the Directed Acyclic Diffusion Language Model (DA-DLM), which adapts the principle of dependency modeling originated from DAT to DLMs' iterative setting. Built upon BD3LM, DA-DLM constructs a position-oriented DAG at each denoising step, in which every node is tied to a specific output position: Tokens fixed in earlier steps are pinned at their positions as single-node anchors in the DAG, structurally guiding predictions at surrounding positions. Positions still undetermined are each expanded into a group of candidates tied to it, capturing local modalities. Transitions between adjacent groups encode inter-token dependencies, so that jointly selecting one candidate from each group yields a transition path and hence coordinated predictions rather than independent token choices. As denoising progresses, the DAG evolves accordingly to reflect the current state of generation: the number of anchors increases while fewer candidate groups remain, gradually refining the graph to resolve the remaining uncertainty. Figure 1 gives an overview of the model and illustrates how the position-oriented DAG evolves in one denoising step.

We evaluate DA-DLM on language modeling, open-ended generation, and the downstream summarization task. Results show that DA-DLM consistently outperforms BD3LM baselines across all settings, with particularly large gains when fewer denoising steps are used, and matches the autoregressive baseline on summarization while retaining the parallel decoding advantage.

To summarize, this paper makes the following contributions:

- We identify a connection between DLMs and NATs, which opens up a promising avenue for drawing inspiration from NAT research to improve DLMs.
- Inspired by DAT, a NAT solution designed to address the conditional independence limitation, we propose DA-DLM, which introduces explicit token dependency modeling into DLMs' denoising process, replacing independent per-position prediction with coordinated joint modeling over a position-oriented DAG at each denoising step.
- We evaluate DA-DLM across diverse tasks and inference scenarios, demonstrating consistent gains over the BD3LM baseline and competitive performance with autoregressive models.

## 2 Background and Motivation

### 2.1 Block Diffusion

Block Diffusion (BD3LM) (Arriola et al., 2025) is the backbone of leading DLMs (Bie et al., 2025, 2026), which we adopt as the foundation for our method. BD3LM partitions a sequence into $B$ blocks of fixed size $N$ and factorizes the likelihood autoregressively across blocks:

$$\log p_\theta(x) = \sum_{b=1}^{B} \log p_\theta(x^b \mid x^{<b}). \quad (1)$$

Within each block, $p_\theta(x^b \mid x^{<b})$ is modeled via continuous-time masked diffusion. The forward noising process $q(\cdot)$ randomly masks tokens in the current block $x^b$ according to the noise schedule $\alpha_t$, producing a partially masked block $x_t^b$ at time $t$. The preceding blocks $x^{<b}$ serve as conditioning that are not subject to masking.

The denoiser reverses this process under the SUBS parameterization (Sahoo et al., 2024): it independently predicts clean tokens at each masked position, while unmasked tokens are carried over unchanged. The training objective adopts the sim-

plified NELBO of Sahoo et al. (2024):

$$\mathcal{L}_{BD3LM}(x;\theta) = \sum_{b=1}^{B} \mathbb{E}_{t,q} \frac{\alpha_t'}{1-\alpha_t} \cdot \sum_i \log p_\theta(x_i^b \mid x_t^b, x^{<b}), \quad (2)$$

where $i$ ranges over all block positions, $\alpha_t' = \mathrm{d}\alpha_t/\mathrm{d}t$; only masked positions contribute.

At inference time, each block is generated by iterative denoising from a fully masked sequence: at each step, the model predicts all masked positions in parallel, a subset of predictions is fixed based on a schedule or confidence criterion, and fixed tokens remain unchanged in subsequent steps.

### 2.2 Cost of Conditional Independence

BD3LM's per-position factorized prediction mirrors the approach of Non-Autoregressive Translation (NAT) (Gu et al., 2018): both independently predict multiple tokens given a masked or incomplete sequence. The cost of this conditional independence has been formally characterized by Huang et al. (2022a) in the NAT literature. They show that for any model predicting $M$ tokens independently given arbitrary conditioning $X$:

$$\min_\theta D_{\mathrm{KL}}[P_{\mathrm{data}}(Y|X) \| P_\theta(Y|X)] \geq C, \quad (3)$$

where $C = \sum_{i=1}^{M} H(y_i|X) - H(Y|X)$ is the conditional total correlation: the dependency information among target tokens that factorized prediction necessarily discards. Since BD3LM employs the same factorized prediction per step, this analytical framework applies. The cost is inherent to per-position independent prediction instead of being specific to either NAT or DLM, manifesting as the multi-modality problem in NAT and as coherence degradation with fewer denoising steps in BD3LM.

### 2.3 Dependency Modeling as Missing Piece

Huang et al. (2022a) further extend their analysis to iterative NATs (Appendix A in Huang et al., 2022a), showing that iterative refinement progressively enriches the per-step conditioning, reducing the conditional independence cost (Eq. 3) at each step; both iterative NATs (e.g., CMLM (Ghazvininejad et al., 2019)) and BD3LM already employ this approach. Their main analysis, on the other hand, focuses on single-step NAT, where the Directed Acyclic Transformer (DAT) (Huang et al., 2022b) stands out by explicitly modeling inter-token dependencies: it constructs a Directed Acyclic Graph (DAG) over the output, where nodes carry token predictions and edges encode probabilistic transitions. The training objective maximizes the data likelihood marginalized over all paths through the DAG, and inference selects the most probable path, yielding coordinated predictions in a single generation step. This dependency-based approach offers a complementary path to iterative refinement and is precisely what BD3LM currently lacks.

However, the iterative setting of BD3LM introduces a requirement absent from single-step generation. In BD3LM, tokens fixed in earlier denoising steps serve as anchors that guide subsequent predictions (Sun et al., 2025; Hong et al., 2026); successive steps build upon previously committed tokens. DAT's position-free DAG, in which paths may skip arbitrary nodes via unrestricted transitions, dilutes the influence of these anchors: fixed tokens cannot reliably anchor their surrounding predictions, causing successive denoising steps to drift apart rather than converge toward a coherent output. Adapting dependency modeling to the iterative setting, therefore, requires a structural redesign that respects position stability and the evolving masking state across steps.

## 3 DA-DLM: Directed Acyclic Diffusion Language Model

As established in §2.3, BD3LM already benefits from iterative refinement but lacks dependency modeling within each step's predictor. We address this by constructing a position-oriented DAG at each denoising step, whose structure reflects the current masking state and evolves as tokens are progressively determined.

### 3.1 Position-Oriented DAG

**Group-wise structure.** To incorporate a DAG into BD3LM's per-step predictor, we require a stable correspondence between nodes and output positions that allows fixed tokens to consistently occupy designated nodes across denoising steps. In a position-free DAG, transitions obey a left-to-right ordering but may skip arbitrary nodes. Fixed tokens can be enforced as path waypoints—nodes that every valid path must pass through—yet the remaining nodes between waypoints are not bound to fixed output positions and may still realign across denoising steps.

The position-oriented DAG addresses this with a group-wise organization. For a block of $N$ positions, the DAG consists of $N$ groups, one per position. Each group $i$ contains $K$ candidate nodes

$\{v_i^1, \ldots, v_i^K\}$, each carrying an independent token prediction distribution. A path $A = (a_1, \ldots, a_N)$, with $a_i \in \{1, \ldots, K\}$, selects exactly one node from each group, producing a sequence of $N$ tokens inherently aligned with the output positions. The group chain is bookended by a start-of-block and an end-of-block node that provide the initial transition distribution and fixed termination.

**Anchors and candidate groups.** The DAG distinguishes two types of groups reflecting the denoising state. A position whose token has been fixed in a previous step reduces to a single-node anchor: every path must pass through this node, pinning the known token at its position. A position still undetermined forms a candidate group with $K$ active nodes, capturing plausible alternatives for that position. This structure evolves with denoising: as tokens are progressively fixed, candidate groups become anchors, and the DAG focuses on the remaining uncertainty, as illustrated in Figure 1.

**Inter-group transitions.** Token dependencies are explicitly modeled through transitions between adjacent groups: from each node in group $i$ to the candidates in group $i+1$. The transition probability is computed via dot-product attention (Huang et al., 2022b), with softmax normalized over the next group's candidates:

$$T(v_i^j \to v_{i+1}^k) = \frac{\exp\big((\mathbf{q}_i^j)^\top \mathbf{k}_{i+1}^k / \sqrt{d}\big)}{\sum_{k'=1}^{K} \exp\big((\mathbf{q}_i^j)^\top \mathbf{k}_{i+1}^{k'} / \sqrt{d}\big)}, \quad (4)$$

where $\mathbf{q}_i^j$ and $\mathbf{k}_{i+1}^k$ are query and key projections of the hidden states at nodes $v_i^j$ and $v_{i+1}^k$, respectively.

Since an anchor at position $i$ is the sole node in its group, all incoming transitions from the preceding group converge to it with probability one. However, its outgoing transitions form a learned distribution over the next group's candidates, allowing the fixed token to directly shape which candidates are favored at the adjacent position in addition to serving as attention context for all positions.

**Path probability.** The joint probability of a target sequence $x^b$ and a path $A$ decomposes into per-node emission probabilities (i.e., token prediction distributions) and inter-group transition probabilities:

$$P_\theta(x^b, A | x_t^b, x^{<b}) = \prod_{i=1}^{N} p_\theta(x_i^b | v_i^{a_i}, x_t^b, x^{<b}) \cdot \prod_{i=1}^{N-1} T(v_i^{a_i} \to v_{i+1}^{a_{i+1}}), \quad (5)$$

where $x_t^b$ is the partially masked block at time $t$ and $x^{<b}$ denotes the preceding blocks. Marginalizing over all paths yields the model likelihood $p_\theta(x^b \mid x_t^b, x^{<b}) = \sum_A P_\theta(x^b, A \mid x_t^b, x^{<b})$.

In contrast to DAT's position-free DAG, where a path selects $M$ nodes from $L \gg M$ nodes via a global $L \times L$ transition matrix, DA-DLM restricts each transition to a local $K \times K$ matrix between adjacent groups, and every path has length of exactly $N$—a structural property that maintains position stability across denoising steps.

### 3.2 Training

**Training objective.** DA-DLM retains BD3LM's block-autoregressive factorization (Eq. 1) and replaces the per-position factorized predictor within each block with the DAG-based joint distribution. Under the same noise-schedule weighting as BD3LM (Sahoo et al., 2024; Arriola et al., 2025), the training objective becomes:

$$\mathcal{L}_{DA-DLM}(x; \theta) = \sum_{b=1}^{B} \mathbb{E}_{t,q} \frac{\alpha_t'}{1 - \alpha_t} \cdot \log \sum_A P_\theta(x^b, A \mid x_t^b, x^{<b}). \quad (6)$$

Compared to BD3LM's training objective (Eq. 2), the per-position independent log-probabilities $\sum_i \log p_\theta(x_i^b \mid \cdot)$ are replaced by a single path-marginalized term $\log \sum_A P_\theta(x^b, A \mid \cdot)$.

**Efficient marginalization via forward algorithm.** The objective requires marginalizing over all paths, which is expensive due to the combinatorial number of paths. We employ a forward algorithm analogous to that in hidden Markov models (details in Appendix A). Because transitions only connect adjacent groups, each recursion step aggregates over at most $K$ predecessors within the preceding group, yielding $O(NK^2)$ time complexity. In contrast, in a position-free DAG (Huang et al., 2022b) with $L{=}NK$ vertices, each node aggregates over all predecessors, resulting in $O(N^3K^2)$ complexity—a factor of $N^2$ higher.

**Diffusion training regime.** Following BD3LM's training protocol (Sahoo et al., 2024; Arriola et al., 2025), each training step samples a timestep $t$ for each block, and the forward masking process corrupts the target $x^b$ into a partially observed $x_t^b$ by independently masking each token with probability $1 - \alpha_t$. The resulting masking pattern directly instantiates the position-oriented DAG: unmasked positions form single-node anchors while masked positions expand into candidate groups, and the model is trained to optimize the path-marginalized

objective (Eq. 6) over this structure. As $t$ varies across training steps, the model encounters DAGs with different anchor densities—from mostly candidate groups under heavy masking to predominantly anchors under light masking—naturally exposing the model to the full spectrum of partial context.

### 3.3 Inference

DA-DLM inherits BD3LM's semi-autoregressive generation. Within each block, generation proceeds by iterative denoising over $T$ steps, starting from a fully masked sequence. At each step, the model constructs a DAG from the current masking state, selects a path to obtain token predictions, applies a remasking strategy to determine which tokens become anchors, and optionally refines local predictions via infilling (illustrated in Figure 1).

**Path selection.** Given emissions and transitions from a forward pass, two decoding strategies operate on the position-oriented DAG.

The Viterbi Strategy finds the jointly optimal path and token sequence (Shao et al., 2022). Analogous to the training forward algorithm, we apply a max-product recursion over the group chain with the same $O(NK^2)$ complexity (details in Appendix A). In a position-free DAG (Shao et al., 2022), the algorithm must additionally search over all possible path lengths and rerank candidates with a tuned length penalty. Our position-oriented structure eliminates this overhead: the path length is fixed to $N$, requiring no length search or additional hyperparameters, and the time complexity reduces from $O(N^3K^2)$ to $O(NK^2)$.

The Sampling Strategy draws a path from the DAG's distribution (Eq. 5) via ancestral sampling: starting from the start-of-block node, we sequentially draw each group's candidate from the current node's outgoing transitions and sample a token from the selected node's emission.

**Remasking.** After predicting tokens along the selected path, a remasking strategy determines which predictions to fix as anchors for the next denoising step. Standard masked diffusion applies random remasking (Nie et al., 2025; Sahoo et al., 2024; Arriola et al., 2025), where the noise schedule determines the expected number of tokens to unmask at each step and the selection is random; confidence-aware parallel decoding—fixing tokens whose prediction confidence exceeds a threshold—is another widely adopted alternative (Wu et al., 2025). We adapt the confidence-based approach to the position-oriented DAG by defining a composite score that incorporates both the emission and transition information along the selected path:

$$s_i = (1 - w) \cdot p_{\text{emit}}(i) + w \cdot p_{\text{trans}}(i), \quad (7)$$

where $p_{\text{emit}}(i)$ is the emission probability of the predicted token, $p_{\text{trans}}(i)$ is the average of the incoming and outgoing transition probabilities at position $i$ along the selected path, and $w$ is the adjusted weight. Positions with $s_i$ above a threshold are fixed as anchors; the rest are remasked.

**Infilling.** As denoising progresses, fixed anchors may partition the remaining masked positions into isolated gaps—contiguous spans bounded by anchors on both sides. When a gap is sufficiently short, the bounding anchors highly constrain the generation space, so we directly fix predictions within such gaps without remasking. Infilling is governed by two parameters: `gap_length` $k$ and `context_ratio` $r$. It takes effect only when the fraction of already-fixed positions exceeds $r$; at that point, all gaps of length $\leq k$ are filled in a single step.

## 4 Experiments

We follow the architecture and training recipe of BD3LM (Arriola et al., 2025); all models are trained on OpenWebText (Gokaslan and Cohen, 2019) (context length 1024). DA-DLM uses $K{=}4$ candidates per group; the DAG-specific modules add $\sim$2.5M parameters ($\sim$2% of the backbone), keeping the model size comparable to BD3LM. Full training details are in Appendix B.

### 4.1 Language Modeling

**Evaluation setup.** Following BD3LM's evaluation protocol (Arriola et al., 2025), we evaluate on two benchmarks: OpenWebText (Gokaslan and Cohen, 2019) (context length 1024) and One Billion Word (LM1B; (Chelba et al., 2013), context length 128). In addition to the OpenWebText models described above, we train a separate set of models on LM1B following the same training procedure. For each dataset, we compare MDLM and BD3LM with block sizes $\in \{16, 8, 4\}$, and the corresponding DA-DLM variants. The reported metric is the noise-schedule-weighted average of negative conditional log-likelihoods (Eq. 2 and Eq. 6); lower values indicate better modeling quality. For MDLM and BD3LM, the loss decomposes into independent per-position terms, and the resulting sum coincides with the negative evidence lower bound (NELBO),

| Models | Block Size | Language Modeling (↓) | |
|---|---|---|---|
| | | LM1B | OpenWebText |
| MDLM | — | 3.47 | 3.15 |
| BD3LM | 16 | 3.46 | 3.11 |
| | 8 | 3.43 | 3.09 |
| | 4 | 3.39 | 3.04 |
| DA-DLM | 16 | **3.30** | **2.92** |
| | 8 | **3.28** | **2.89** |
| | 4 | **3.24** | **2.86** |

Table 1: Results on language modeling tasks. The reported metric is the weighted average of negative conditional log-likelihood, where lower values are better.

| Denoising | Infilling | TPF | MAUVE: Block Size = 16 | |
|---|---|---|---|---|
| | | | BD3LM | DA-DLM |
| FH | – | 1.00 | 0.964 | **0.969** |
| FH | $(2, 0.5)$ | 1.54 | 0.869 | **0.940** |
| FH | $(2, 0.0)$ | 1.78 | 0.710 | **0.910** |
| FH | $(3, 0.5)$ | 1.66 | 0.618 | **0.906** |
| $T = 16$ | – | 1.55 | 0.859 | **0.916** |
| $T = 16$ | $(2, 0.5)$ | 1.99 | 0.587 | **0.839** |
| $T = 16$ | $(2, 0.0)$ | 1.94 | 0.303 | **0.822** |
| $T = 16$ | $(3, 0.5)$ | 2.18 | 0.270 | **0.817** |
| $T = 12$ | – | 1.78 | 0.627 | **0.877** |
| $T = 12$ | $(2, 0.5)$ | 2.21 | 0.404 | **0.714** |
| $T = 12$ | $(2, 0.0)$ | 2.18 | 0.220 | **0.694** |
| $T = 12$ | $(3, 0.5)$ | 2.42 | 0.198 | **0.647** |
| $T = 8$ | – | 2.27 | 0.406 | **0.672** |
| $T = 6$ | – | 2.82 | 0.119 | **0.403** |
| $T = 4$ | – | 4.04 | 0.028 | **0.099** |

Table 2: MAUVE–TPF comparison between BD3LM and DA-DLM on OpenWebText generation with block size 16. MAUVE and TPF for the AR model are 0.984 and 1.00, for reference.

a variational upper bound on the true negative log-likelihood (NLL) (Sahoo et al., 2024; Arriola et al., 2025). DA-DLM's path-marginalized objective uses the same noise-schedule weighting but couples predictions across positions through transition probabilities, preventing the loss from factorizing into a per-position bound on NLL. As a result, the metric is not a consistent bound on NLL across all models, and we do not convert it to perplexity. We accordingly exclude the AR baseline, whose exact NLL is not directly comparable. The two diffusion objectives are nonetheless directly comparable, as both are evaluated under the same noise schedule, weighting, and data.

**Results.** Table 1 reports language modeling performance on both benchmarks. Consistent with Arriola et al. (2025), smaller block sizes yield better performance; DA-DLM consistently outperforms BD3LM at every block size.This advantage stems from dependency modeling: whereas BD3LM predicts each position independently, DA-DLM's transition probabilities couple adjacent predictions along paths, concentrating probability mass on coherent token sequences.

### 4.2 Open-Ended Generation

**Evaluation setup.** We evaluate open-ended generation quality with prefix-conditioned generation on OpenWebText and measure the MAUVE score (Pillutla et al., 2021) (details in Appendix C). We compare an AR baseline, BD3LM with block sizes $\in \{16, 8, 4\}$, and the corresponding DA-DLM variants. To map the quality–efficiency tradeoff, we vary the number of denoising steps $T$ per block and report results as a function of tokens per forward pass (TPF), defined as the average number of tokens determined per forward pass, measured at runtime. We evaluate two sampling mechanisms: (1) first-hitting sampling (FH) (Ou et al., 2025), where each forward pass determines one token (TPF$=1$); and (2) random remasking (Sahoo et al., 2024; Nie et al., 2025), where the noise schedule determines the expected number of tokens to unmask at each step. DA-DLM additionally employs the sampling path selection strategy (§3.3) to decode from the DAG at each denoising step. We also evaluate both models with and without the infilling mechanism (§3.3), parameterized by gap length $k$ and context ratio $r$ (reported as $(k, r)$ in Table 2).

**Results.** Table 2 reports detailed results for block size 16. Under first-hitting sampling (TPF=1), both models achieve comparable MAUVE scores, but as TPF increases—i.e., as each denoising step determines more tokens in parallel—BD3LM's generation quality drops sharply while DA-DLM maintains substantially higher MAUVE. When the number of denoising steps becomes very small, both models eventually degrade.

Infilling further differentiates the two models: for BD3LM, infilling largely follows the existing quality–efficiency tradeoff curve, offering little advantage beyond what reducing denoising steps already provides. For DA-DLM, infilling yields a favorable tradeoff when gap regions are well-constrained by surrounding anchors—for instance, when combined with sufficient denoising steps or when the context ratio gates infilling on adequate surrounding context. DA-DLM's transition probabilities propagate dependency information from anchors into gaps, making infilling an effective accelerator that expands DA-DLM's Pareto frontier to higher TPF values.

Figure 2 extends this analysis across all three

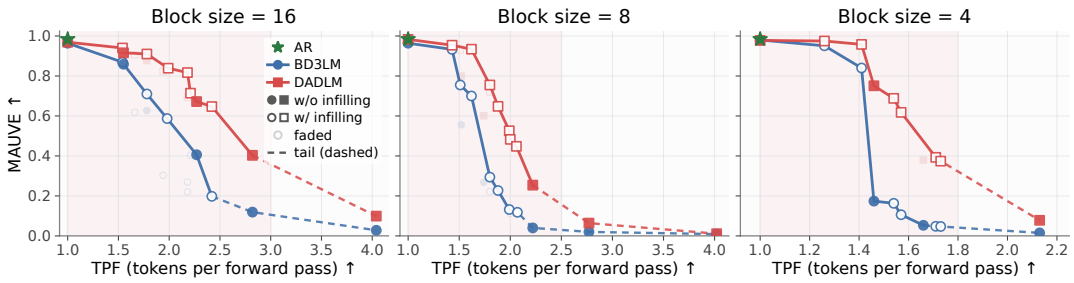


Figure 2: MAUVE–TPF trade-off across different block sizes. Each panel reports the Pareto frontier between generation quality (measured by MAUVE) and computational efficiency (measured by TPF).

block sizes (detailed numerical results in Tables 4–6, Appendix E), where each panel plots the Pareto frontier between MAUVE and TPF: solid lines trace the Pareto frontier and dashed segments mark the high-TPF tail(both models degrade eventually as the number of denoising steps becomes very small); filled markers denote configurations without infilling and open markers denote infilling variants. The shaded region highlights DA-DLM's operating regime—the TPF range over which it maintains a favorable quality–efficiency tradeoff. The trends observed for block size 16 hold consistently: DA-DLM's Pareto frontier consistently surpasses BD3LM's across all block sizes, and infilling configurations extend DA-DLM's frontier further into higher TPF regions. Faded markers indicate configurations dominated within each model's own frontier. Smaller block sizes achieve higher absolute MAUVE under first-hitting sampling, but degrade more steeply under random remasking due to cross-block error compounding: smaller blocks partition the sequence into more blocks, so each block's parallel generation errors accumulate more rapidly through the conditioning prefix, creating an increasing distribution shift from the ground-truth prefixes seen in training. DA-DLM mitigates this by maintaining higher per-block quality, reducing the error that propagates into downstream blocks.

### 4.3 Downstream Task: Summarization

**Evaluation setup.** We evaluate on two summarization datasets—XSum (Narayan et al., 2018) and CNN/DailyMail (Nallapati et al., 2016). Models pre-trained on OpenWebText are fine-tuned for this task. Based on the observation from §4.2 that models with smaller block sizes struggle to maintain generation quality over long sequences, we select a block size of 16 for both BD3LM and DA-DLM. We report ROUGE (Lin and Hovy, 2003) and tokens per forward pass (TPF) as quality and efficiency metrics; decoding configurations are detailed in Appendix D.

**Results.** Table 3 reports ROUGE scores and TPF on both summarization benchmarks. DA-DLM outperforms BD3LM in average ROUGE on both datasets and matches or slightly exceeds the AR baseline, demonstrating that the benefits of dependency modeling transfer to task-specific generation after fine-tuning. The advantage concentrates on ROUGE-1 and ROUGE-L, suggesting that DA-DLM's joint modeling over paths helps the model stay on topic and maintain sequential coherence.

Both BD3LM and DA-DLM achieve TPF well above 1.0 under confidence-aware decoding alone, indicating strong task adaptation. Infilling consistently increases TPF on both datasets with minimal impact on quality: average ROUGE changes by less than 0.05 in either direction.

### 4.4 Ablation: Position-Oriented DAG

**Evaluation setup.** To validate the position-oriented DAG design, we compare DA-DLM against a position-free variant (FREE-DA-DLM) that adopts the original DAG design of DAT (Huang et al., 2022b) in which latent nodes are not bound to specific output positions. We evaluate on the open-ended generation benchmark (§4.2) with block size 16 under varying numbers of denoising steps. Since DAT originally targets one-step parallel generation, we additionally test FREE-DA-DLM in a one-step mode. We note that the position-free structure precludes first-hitting sampling and the infilling mechanism, both of which rely on position correspondence between DAG nodes and output tokens; FREE-DA-DLM's TPF is therefore reported separately. Full results are in Table 7 (Appendix E).

**Results.** FREE-DA-DLM performs dramatically worse than DA-DLM across all configurations (e.g., MAUVE $= 0.187$ vs. $0.916$ at $T{=}16$) and even underperforms BD3LM when $T{>}4$, confirming that

| Models | Strategy | Summarization | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | XSum | | | | | CNN/DailyMail | | | | |
| | | R-1 | R-2 | R-L | Avg | TPF | R-1 | R-2 | R-L | Avg | TPF |
| AR | Greedy | 35.72 | **13.51** | **28.31** | 25.85 | 1.00 | 38.54 | 16.36 | **26.57** | 27.16 | 1.00 |
| BD3LM | Conf 0.9 | 36.05 | 13.26 | 27.64 | 25.65 | 1.07 | 38.21 | **16.85** | 25.95 | 27.00 | 1.81 |
| | Conf 0.9 + Infill | 36.10 | 13.17 | 27.67 | 25.65 | 1.20 | 38.21 | 16.83 | 25.94 | 26.99 | 1.92 |
| DA-DLM | Conf 0.9 | 36.44 | 13.15 | 28.06 | 25.88 | 1.13 | **38.60** | 16.76 | 26.16 | **27.17** | 1.73 |
| | Conf 0.9 + Infill | **36.55** | 13.14 | 28.09 | **25.93** | **1.50** | 38.53 | 16.72 | 26.10 | 27.12 | **2.56** |

Table 3: Results on summarization tasks. 'R-1/2/L' denotes ROUGE-1 / ROUGE-2 / ROUGE-L scores. 'Conf 0.9' means confidence-aware parallel decoding with threshold 0.9, and 'Infill' denotes infilling enabled.

the position-free DAG structure is incompatible with iterative denoising. These results support the analysis in §2: without position binding, the DAG cannot stably anchor predictions to specific output locations as tokens are progressively revealed.

## 5 Related Work

### 5.1 Directed Acyclic Transformers

Since the introduction of DAT (Huang et al., 2022b), a number of follow-up works have improved its theoretical understanding (Huang et al., 2022a), decoding (Shao et al., 2022), architectures (Li et al., 2024, 2025), and training objectives (An et al., 2023; Ma et al., 2023).

PreDAT (Huang et al., 2023) extends DAT from machine translation to diverse text generation tasks via pre-training, sharing our goal of broadening DAG-based generation beyond translation, yet it remains within the encoder-decoder paradigm. Diff-DAT (Nguyen-Tri et al., 2025), focusing on seq2seq machine translation, shares with our work the idea of extending DAT from single-step to iterative generation via a diffusion process. However, both PreDAT and Diff-DAT retain DAT's position-free DAG, which dilutes the anchoring effect of fixed tokens across denoising steps. Consequently, PreDAT remains a single-step generator, and Diff-DAT reports that iterative decoding degrades beyond two steps. DA-DLM addresses this limitation with the position-oriented DAG and further departs from the seq2seq paradigm by building on block diffusion (Arriola et al., 2025) in a decoder-only architecture for general-purpose language modeling.

### 5.2 Dependency Modeling in DLMs

Recent work on the conditional independence limitation in DLMs follows two lines.

The first retains the factorized predictor and supplements it with an external or static dependency source: DCD (Liu et al., 2025) couples the diffusion model's marginals with an external pre-trained autoregressive model at inference time, and CoDD (Li et al., 2026) augments the output distribution with a static Probabilistic Circuit once trained. Neither condition the dependency information on the evolving denoising state.

The second operates on the decoding side: DEMASK (Ringel et al., 2026) and DAPD (Kim et al., 2026) estimate pairwise token dependencies and selectively unmask only approximately independent subsets at each step.

DA-DLM differs from both: its DAG-based transitions are produced by the same backbone conditioned on the current denoising state, making dependency modeling dynamic; and rather than avoiding co-prediction of dependent tokens, it explicitly models their joint distribution through path marginalization.

## 6 Conclusion

We presented DA-DLM, which introduces explicit inter-token dependency modeling into diffusion language models by constructing a position-oriented DAG at each denoising step. By bridging the connection between DLMs and NATs, we adapted the DAG-based joint prediction mechanism from DAT to the iterative denoising setting. The key design is the position-oriented DAG, which binds each node group to a fixed output position so that tokens fixed in earlier steps serve as anchors that reliably guide surrounding predictions—resolving the incompatibility of DAT's position-free DAG with iterative denoising. Experiments on language modeling, open-ended generation, and summarization show that DA-DLM consistently outperforms the BD3LM baseline, with particularly pronounced gains when fewer denoising steps are used, and achieves competitive performance with autoregressive models while retaining the parallel decoding advantage.

## 7 Limitation

Although DA-DLM can be obtained by up-training from an existing BD3LM checkpoint without training from scratch, our experiments are conducted at a relatively small model scale due to computational constraints. The evaluations focus on the traditional DLM benchmarks, conditional log-likelihood, open-ended generation quality, and a single downstream task, but do not yet extend to the broader set of capabilities expected of modern language models, such as mathematical reasoning or code generation. As larger open-source discrete diffusion models become available (Nie et al., 2025; Bie et al., 2025, 2026), up-training on these checkpoints and evaluating on a wider range of benchmarks would provide a more comprehensive validation of the approach.

## References


Chenxin An, Jiangtao Feng, Fei Huang, Xipeng Qiu, and Lingpeng Kong. 2023. Optimizing non-autoregressive transformers with contrastive learning. *arXiv preprint arXiv:2305.13667*.

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

Tiwei Bie, Maosong Cao, Xiang Cao, Bingsen Chen, Fuyuan Chen, Kun Chen, Lun Du, Daozhuo Feng, Haibo Feng, Mingliang Gong, and 1 others. 2026. Llada2. 1: Speeding up text diffusion via token editing. *arXiv preprint arXiv:2602.08676*.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, and 1 others. 2025. Llada2. 0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.

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

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Chunsan Hong, Sanghyun Lee, Chieh-Hsin Lai, Satoshi Hayakawa, Yuhta Takida, Yuki Mitsufuji, Seungryong Kim, and Jong Chul Ye. 2026. Understanding and accelerating the training of masked diffusion language models. *arXiv preprint arXiv:2605.13026*.

Fei Huang, Pei Ke, and Minlie Huang. 2023. Directed acyclic transformer pre-training for high-quality non-autoregressive text generation. *Transactions of the Association for Computational Linguistics*, 11:941–959.

Fei Huang, Tianhua Tao, Hao Zhou, Lei Li, and Minlie Huang. 2022a. On the learning of non-autoregressive transformers. In *International conference on machine learning*, pages 9356–9376. PMLR.

Fei Huang, Hao Zhou, Yang Liu, Hang Li, and Minlie Huang. 2022b. Directed acyclic transformer for non-autoregressive machine translation. In *International Conference on Machine Learning*, pages 9410–9428. PMLR.

Bumjun Kim, Dongjae Jeon, Moongyu Jeon, and Albert No. 2026. Dependency-aware parallel decoding via attention for diffusion llms. *arXiv preprint arXiv:2603.12996*.

Haoran Li, Zhanming Jie, and Wei Lu. 2024. Non-autoregressive machine translation as constrained HMM. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12361–12372, Bangkok, Thailand. Association for Computational Linguistics.

Ian Li, Zilei Shao, Benjie Wang, Rose Yu, Guy Van den Broeck, and Anji Liu. 2026. Breaking the factorization barrier in diffusion language models. *arXiv preprint arXiv:2603.00045*.

Shijie Li, Inigo Jauregi Unanue, and Massimo Piccardi. 2025. DAT-CRF: improving the directed acyclic transformer with CRF integration. In *ECAI 2025*, pages 4313–4320. IOS Press.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 150–157.

Anji Liu, Oliver Broadrick, Mathias Niepert, and Guy Van den Broeck. 2025. Discrete copula diffusion. In *International Conference on Learning Representations*, volume 2025, pages 88953–88979.

Zhengrui Ma, Chenze Shao, Shangtong Gui, Min Zhang, and Yang Feng. 2023. Fuzzy alignments in directed acyclic graph for non-autoregressive machine translation. In *The Eleventh International Conference on Learning Representations*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Quan Nguyen-Tri, Cong Dao Tran, and Hoang Thanh-Tung. 2025. Diffusion directed acyclic transformer for non-autoregressive machine translation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 814–828, Vienna, Austria. Association for Computational Linguistics.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan LI. 2025. Large language diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, pages 50608–50646. Curran Associates, Inc.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. 2025. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *International Conference on Learning Representations*, volume 2025, pages 64972–65009.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems*, volume 34, pages 4816–4828. Curran Associates, Inc.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Liran Ringel, Ameen Ali, and Yaniv Romano. 2026. Dependency-guided parallel decoding in discrete diffusion language models. *arXiv preprint arXiv:2604.02560*.

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

Chenze Shao, Zhengrui Ma, and Yang Feng. 2022. Viterbi decoding of directed acyclic transformer for non-autoregressive machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4390–4397, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haocheng Sun, Cynthia Xin Wen, and Edward Hong Wang. 2025. Why mask diffusion does not work. *arXiv preprint arXiv:2510.03289*.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. 2025. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*.

## A Algorithm Details

**Forward algorithm.** We recurrently calculate the probability sum of path prefixes that end at candidate $k$ in group $i$ while generating the first $i$ target tokens, denoted as $f_{i,k}$. Since transitions only connect adjacent groups, a path prefix ending at candidate $k$ in group $i$ must come from some candidate $j$ in the preceding group $i-1$, so $f_{i,k}$ can be obtained from $\{f_{i-1,j}\}_{j=1}^{K}$. By advancing one group at a time, we obtain the probability sum of all valid paths as $\sum_{k=1}^{K} f_{N,k}$.

**Viterbi decoding.** Analogous to the forward algorithm, we recurrently calculate the maximum joint score of path prefixes ending at candidate $k$ in group $i$, denoted $g_{i,k}$. Since transitions only connect adjacent groups, $g_{i,k}$ is obtained by taking the best predecessor from group $i-1$, with the score incorporating both the transition to $v_i^k$ and the best token prediction at that node. After processing all $N$ groups, we backtrack from the highest-scoring candidate in the final group to recover the optimal path, and select tokens by taking argmax at each node along the path.

## B General Setup

All models—including the MDLM, BD3LM, and autoregressive (AR) baselines—share a Transformer backbone with 12 layers, 768 hidden dimensions, and 12 attention heads, following the architecture of BD3LM (Arriola et al., 2025). Data is tokenized with the GPT-2 tokenizer (Radford et al., 2019) (vocabulary size 50,257). The AR baseline is trained for half the number of steps to

match the total number of tokens seen (Sahoo et al., 2024).

For diffusion models, training proceeds in two stages: an initial MDLM (Sahoo et al., 2024) phase (850k steps, full-sequence diffusion without block structure) followed by a block-wise up-training phase (150k steps) under varying block sizes. For DA-DLM, the block-wise phase consists of 90k steps of BD3LM training followed by 60k steps of DA-DLM training that introduces the DAG modules (§3), totaling the same 150k up-training budget.

## C Open-Ended Generation Setup

We randomly select validation sequences from OpenWebText, condition on the first 64 tokens as a prompt, and generate the remaining 448 tokens, yielding completions of length 512. We generate 2,500 samples per configuration and compute the MAUVE score against the reference completions, using GPT-2 Large (Radford et al., 2019) as the embedding model. All models employ nucleus sampling with $p$=0.9. The number of denoising steps per block is varied as follows: $\{16, 12, 8, 6, 4\}$ for block size 16; $\{8, 6, 4, 3, 2\}$ for block size 8; $\{4, 3, 2\}$ for block size 4. All diffusion models in our experiments are time-agnostic, i.e., they do not condition on the diffusion timestep (Sahoo et al., 2024; Ou et al., 2025). This allows caching the denoising network's output when no new tokens are unmasked between consecutive steps, so the effective number of forward passes can be smaller than the nominal number of denoising steps, and TPF reflects this reduction. For the infilling mechanism, we test gap lengths $k \in \{2, 3\}$ with and without a context ratio $r$=0.5.

## D Summarization Setup

Documents and summaries are truncated to 768 and 256 tokens, respectively, and diffusion models are fine-tuned with a batch size of 512 for 5k steps; the AR model is fine-tuned for half the number of steps, consistent with the pre-training configuration. Because summarization emphasizes generation accuracy over diversity, we adopt deterministic or high-confidence decoding strategies. The AR model uses greedy decoding. For BD3LM, we adopt confidence-aware parallel decoding (Wu et al., 2025) with a threshold of 0.9. For DA-DLM, we combine Viterbi path selection with confidence-aware parallel decoding, where the composite confidence score (Eq. 7) with $w$=0.5 integrates transition information into the remasking decision. We additionally evaluate both diffusion models with the infilling mechanism (§3.3, $k$=2, $r$=0.5).

## E Detailed Experimental Results

**Training compute alignment.** Although BD3LM and DA-DLM share the same 150k-step block-wise up-training budget, DA-DLM's per-step FLOPs are higher due to the expanded input from the DAG's candidate nodes. To rule out the possibility that DA-DLM's gains merely reflect greater total training compute, we compare against BD3LM-300K, a BD3LM variant trained for 300k block-wise up-training steps to match DA-DLM's total training FLOPs. Table 8 shows that DA-DLM still consistently outperforms BD3LM-300K across all configurations, confirming that the improvements stem from the DAG-based dependency modeling rather than from additional training compute.

| Denoising | Infilling | TPF | MAUVE: Block Size = 16 | |
|---|---|---|---|---|
| | | | BD3LM | DA-DLM |
| FH | – | 1.00 | 0.964 | **0.969** |
| FH | $k = 2, r = 0.5$ | 1.54 | 0.869 | **0.940** |
| FH | $k = 2$ | 1.78 | 0.710 | **0.910** |
| FH | $k = 3, r = 0.5$ | 1.66 | 0.618 | **0.906** |
| $T = 16$ | – | 1.55 | 0.859 | **0.916** |
| $T = 16$ | $k = 2, r = 0.5$ | 1.99 | 0.587 | **0.839** |
| $T = 16$ | $k = 2$ | 1.94 | 0.303 | **0.822** |
| $T = 16$ | $k = 3, r = 0.5$ | 2.18 | 0.270 | **0.817** |
| $T = 12$ | – | 1.78 | 0.627 | **0.877** |
| $T = 12$ | $k = 2, r = 0.5$ | 2.21 | 0.404 | **0.714** |
| $T = 12$ | $k = 2$ | 2.18 | 0.220 | **0.694** |
| $T = 12$ | $k = 3, r = 0.5$ | 2.42 | 0.198 | **0.647** |
| $T = 8$ | – | 2.27 | 0.406 | **0.672** |
| $T = 6$ | – | 2.82 | 0.119 | **0.403** |
| $T = 4$ | – | 4.04 | 0.028 | **0.099** |

Table 4: MAUVE–TPF comparison between BD3LM and DA-DLM on OpenWebText generation with block size 16. Boldface denotes the better MAUVE score between BD3LM and DA-DLM under the same decoding strategy. Here $T$ denotes the number of denoising steps, $k$ denotes the infill gap length, and $r$ denotes the context ratio. MAUVE and TPF for the AR model are $0.984$ and $1.00$, for reference.

| Denoising | Infilling | TPF | MAUVE: Block Size = 8 | |
|---|---|---|---|---|
| | | | BD3LM | DA-DLM |
| FH | – | 1.00 | 0.964 | **0.983** |
| FH | $k = 2, r = 0.5$ | 1.43 | 0.933 | **0.954** |
| FH | $k = 2$ | 1.62 | 0.700 | **0.934** |
| FH | $k = 3, r = 0.5$ | 1.51 | 0.755 | **0.930** |
| $T = 8$ | – | 1.52 | 0.556 | **0.799** |
| $T = 8$ | $k = 2, r = 0.5$ | 1.80 | 0.294 | **0.755** |
| $T = 8$ | $k = 2$ | 1.80 | 0.222 | **0.720** |
| $T = 8$ | $k = 3, r = 0.5$ | 1.88 | 0.227 | **0.648** |
| $T = 6$ | – | 1.74 | 0.268 | **0.601** |
| $T = 6$ | $k = 2, r = 0.5$ | 1.99 | 0.132 | **0.526** |
| $T = 6$ | $k = 2$ | 2.00 | 0.112 | **0.483** |
| $T = 6$ | $k = 3, r = 0.5$ | 2.06 | 0.118 | **0.448** |
| $T = 4$ | – | 2.22 | 0.040 | **0.254** |
| $T = 3$ | – | 2.77 | 0.020 | **0.064** |
| $T = 2$ | – | 4.02 | 0.009 | **0.011** |

Table 5: MAUVE–TPF comparison between BD3LM and DA-DLM on OpenWebText generation with block size 8. Boldface denotes the better MAUVE score between BD3LM and DA-DLM under the same decoding strategy. Here $T$ denotes the number of denoising steps, $k$ denotes the infill gap length, and $r$ denotes the context ratio. MAUVE and TPF for the AR model are $0.984$ and $1.00$, for reference.

| Denoising | Infilling | TPF | MAUVE: Block Size = 4 | |
|---|---|---|---|---|
| | | | BD3LM | DA-DLM |
| FH | – | 1.00 | **0.980** | 0.978 |
| FH | $k = 2, r = 0.5$ | 1.26 | 0.951 | **0.975** |
| FH | $k = 2$ | 1.41 | 0.840 | **0.958** |
| $T = 4$ | – | 1.46 | 0.174 | **0.751** |
| $T = 4$ | $k = 2, r = 0.5$ | 1.54 | 0.163 | **0.688** |
| $T = 4$ | $k = 2$ | 1.57 | 0.106 | **0.618** |
| $T = 3$ | – | 1.66 | 0.053 | **0.380** |
| $T = 3$ | $k = 2, r = 0.5$ | 1.71 | 0.048 | **0.393** |
| $T = 3$ | $k = 2$ | 1.73 | 0.047 | **0.374** |
| $T = 2$ | – | 2.13 | 0.015 | **0.078** |

Table 6: MAUVE–TPF comparison between BD3LM and DA-DLM on OpenWebText generation with block size 4. Boldface denotes the better MAUVE score between BD3LM and DA-DLM under the same decoding strategy. Here $T$ denotes the number of denoising steps, $k$ denotes the infill gap length, and $r$ denotes the context ratio. MAUVE and TPF for the AR model are $0.984$ and $1.00$, for reference.

| Denoising | MAUVE | | | | | |
|---|---|---|---|---|---|---|
| | BD3LM | | DA-DLM | | FREE-DA-DLM | |
| | TPF | MAUVE | TPF | MAUVE | TPF | MAUVE |
| $T = 16$ | 1.55 | 0.859 | 1.55 | **0.916** | 1.60 | 0.187 |
| $T = 12$ | 1.78 | 0.627 | 1.78 | **0.877** | 1.83 | 0.166 |
| $T = 8$ | 2.27 | 0.406 | 2.27 | **0.672** | 2.35 | 0.125 |
| $T = 6$ | 2.82 | 0.119 | 2.82 | **0.403** | 2.91 | 0.096 |
| $T = 4$ | 4.04 | 0.028 | 4.04 | **0.099** | 4.14 | 0.059 |
| $T = 1$ | – | – | – | – | 15.76 | 0.010 |

Table 7: MAUVE–TPF comparison among BD3LM, DA-DLM, and FREE-DA-DLM on OpenWebText generation with block size 16. Since FREE-DA-DLM has a different inference-time profile, we report its TPF separately. Boldface denotes the best MAUVE score among the compared models under the same denoising-step setting. Here $T$ denotes the number of denoising steps. MAUVE and TPF for the AR model are $0.984$ and $1.00$, for reference.

| Denoising | Infilling | TPF | MAUVE: Block Size = 16 | |
|---|---|---|---|---|
| | | | BD3LM-300K | DA-DLM |
| FH | – | 1.00 | 0.959 | **0.969** |
| FH | $k = 2, r = 0.5$ | 1.54 | 0.843 | **0.940** |
| FH | $k = 2$ | 1.78 | 0.713 | **0.910** |
| FH | $k = 3, r = 0.5$ | 1.66 | 0.615 | **0.906** |
| $T = 16$ | – | 1.55 | 0.782 | **0.916** |
| $T = 16$ | $k = 2, r = 0.5$ | 1.99 | 0.556 | **0.839** |
| $T = 16$ | $k = 2$ | 1.94 | 0.334 | **0.822** |
| $T = 16$ | $k = 3, r = 0.5$ | 2.18 | 0.338 | **0.817** |
| $T = 12$ | – | 1.78 | 0.725 | **0.877** |
| $T = 12$ | $k = 2, r = 0.5$ | 2.21 | 0.385 | **0.714** |
| $T = 12$ | $k = 2$ | 2.18 | 0.262 | **0.694** |
| $T = 12$ | $k = 3, r = 0.5$ | 2.42 | 0.222 | **0.647** |
| $T = 8$ | – | 2.27 | 0.326 | **0.672** |
| $T = 6$ | – | 2.82 | 0.132 | **0.403** |
| $T = 4$ | – | 4.04 | 0.025 | **0.099** |

Table 8: MAUVE–TPF comparison between BD3LM-300K and DA-DLM on OpenWebText generation with block size 16. Boldface denotes the better MAUVE score between BD3LM-300K and DA-DLM under the same decoding strategy. Here $T$ denotes the number of denoising steps, $k$ denotes the infill gap length, and $r$ denotes the context ratio. MAUVE and TPF for the AR model are $0.984$ and $1.00$, for reference.