MDLMPE: Distribution Aware Positional Encoding for Masked Diffusion Language Models

Tong Ling[1], **Hang Lei**[2], **Feng Xiao**[2], **Changhui Sun**[3], **Jiahang Xie**[4], **Hao Liu**[2], **Lu Liu**[2], **Yanlong Du**[2]

[1]University Chinese Academic of Science
[2]HUJING Digital Media & Entertainment Group
[3]State Key Laboratory for Novel Software Technology, Nanjing University, China
[4]School of Data Science, Fudan University, Shanghai, China

arXiv:2608.03769v1 [cs.CL] 4 Aug 2026

Abstract

Masked diffusion language models (MDLMs) enable parallel generation and bidirectional context modeling, but their positional context differs fundamentally from that of autoregressive (AR) models. Whereas AR decoding exposes a contiguous prefix, MDLM denoising produces dynamic, non-contiguous configurations of revealed and masked tokens. Conventional positional encodings such as RoPE capture sequence order and pairwise displacement but remain insensitive to this evolving token-availability structure. To address this limitation, we propose MDLMPE, a positional encoding designed specifically for masked diffusion. To the best of our knowledge, MDLMPE is the first method to make positional representations explicitly aware of the changing revealed/masked configuration. It represents token availability as a binary sequence, applies distance-aware Gaussian weighting, and projects the resulting pattern through a cosine basis to obtain distribution-aware positional features. These features are added to token embeddings and mapped by a lightweight MLP to angular offsets that modulate the standard RoPE phases. Extensive experiments on LLaDA and DREAM demonstrate that MDLMPE generally outperforms conventional positional encoding methods across supervised fine-tuning, pretraining, zero-shot evaluation, and block-diffusion settings. Further ablations show that the complete combination of availability state, Gaussian locality, spectral basis, and embedding injection yields the strongest result. These results establish the evolving token-availability distribution as a useful positional signal for masked diffusion language models.

1 Introduction

Autoregressive (AR) large language models (Radford et al. 2019; Brown et al. 2020; Touvron et al. 2023) have transformed language modeling, yet their left-to-right factorization imposes sequential generation and causal, unidirectional attention. Bidirectional pretraining enriches contextual representations, but AR decoding remains sequential. This limitation motivates masked diffusion language models (MDLMs), which iteratively denoise a masked sequence and support parallel generation, flexible generation orders (Stern et al. 2019), and bidirectional context modeling (Nie et al. 2025; Ye et al. 2025; Song et al. 2025). Systems including LLaDA (Nie et al. 2025), DREAM (Ye et al. 2025), and LLaDA2.0 (Bie et al. 2025) show that this paradigm can scale to billions of parameters while retaining competitive language-modeling ability. As scalability becomes established, a more specific question emerges: how should positional context be represented when the visible sequence state changes throughout denoising?

This question arises because MDLM generation exposes a positional structure that AR decoding never encounters. An AR model reveals a contiguous prefix, whereas an MDLM may recover tokens in a non-contiguous order. A token can retain the same absolute index while the semantic evidence available around it changes as other positions are revealed or re-masked. Its prediction depends not only on where it is located, but also on how the available context is organized. Figure 1 shows this contrast by comparing the monotonic AR prefix with the evolving position availability of MDLM generation. As highlighted by the two red boxes, the two masked tokens remain at fixed sequence positions throughout denoising, while the distribution of surrounding tokens continually changes; existing positional encoding methods cannot reflect this evolving contextual structure.

The limitation becomes clearer when viewed through Rotary Position Embedding (RoPE) (Su et al. 2024), the mainstream positional encoding used by MDLM backbones. RoPE associates each index with a rotation and represents the relation between two positions through their distance-dependent phase difference. This geometry is suited to AR generation: every visible prefix position contains a real token, so pairwise displacement provides a reliable description of where usable context is located. In an MDLM, however, the same indices and displacement may occur under different revealed/masked configurations. The attention relation should therefore reflect both pairwise distance and the distribution of available evidence around the two positions. This observation suggests three requirements for a position-distribution representation: (i) revealed-token density, which estimates the amount of semantic evidence; (ii) target-relative distance to the revealed-token distribution, which indicates contextual relevance; and (iii) the global topology of the revealed/masked pattern, which distinguishes distributional structures that isolated indices cannot express.

This objective also distinguishes MDLMPE from RoPE adaptations for long-context diffusion models (Liu et al. 2025; He et al. 2025). LongLLaDA applies NTK-based extrapolation, while UltraLLaDA modifies RoPE during post-training for 128K contexts. These methods address coordinate behavior beyond the pretraining length; our focus is

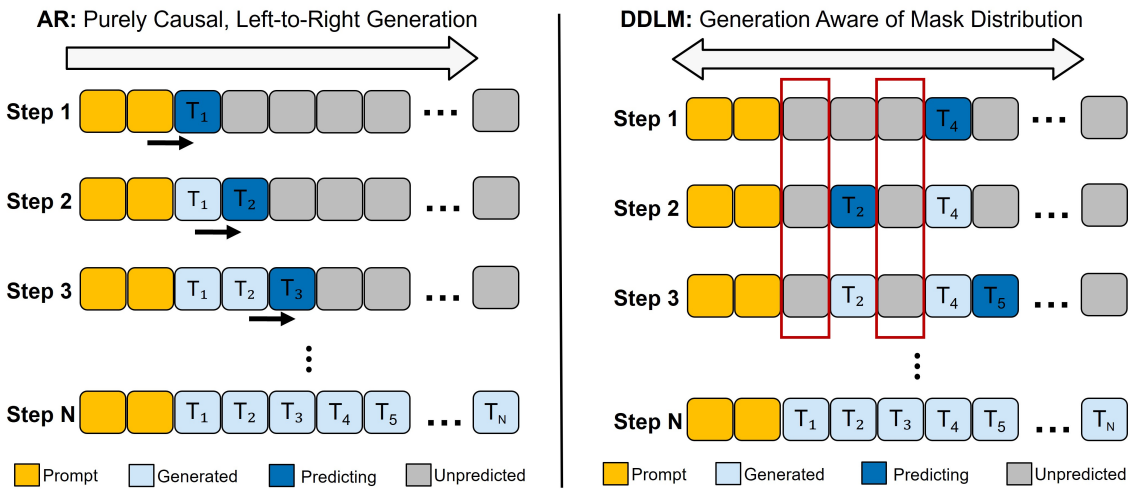

Figure 1: Position availability in AR and masked diffusion generation. AR exposes a contiguous prefix, whereas MDLM denoising reveals a changing, non-contiguous set of response positions.

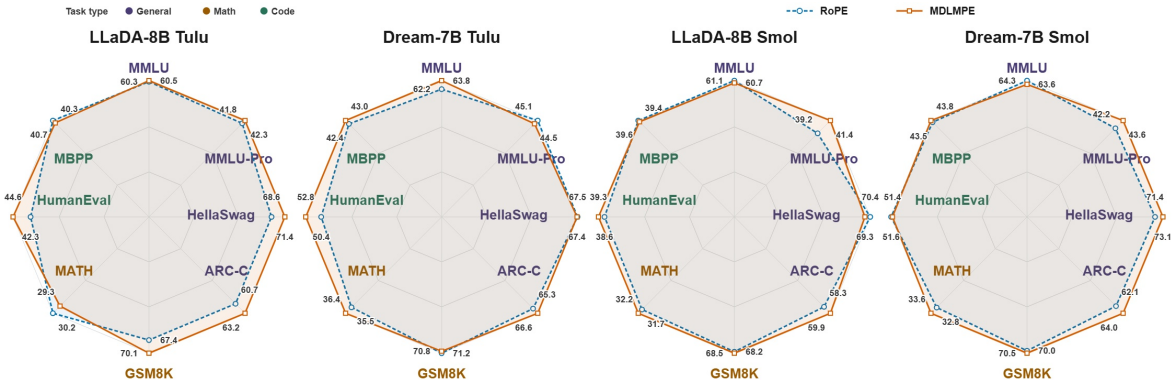

Figure 2: Instruction post-training accuracies for the four table settings, ordered left to right as LLaDA-8B Tulu, DREAM-7B Tulu, LLaDA-8B SmolTalk, and DREAM-7B SmolTalk. Dashed blue circles denote RoPE and solid orange squares denote MDLMPE.

the revealed/masked availability pattern within the current denoising sequence.

Motivated by this gap, we propose **MDLMPE**, a positional encoding that makes the denoising state an explicit positional signal. MDLMPE represents real and masked tokens as a binary availability sequence. It then applies target-centered Gaussian weighting so that the representation reflects where revealed evidence lies relative to each token; for an attention pair, the two centered views are averaged into a joint distribution. A cosine-basis projection converts this pattern into distribution-aware features. MDLMPE uses them through two complementary paths: token features are added to embeddings, while a lightweight MLP maps pair features to angular offsets for RoPE. The resulting design preserves RoPE's relative-position geometry while making both token representations and attention rotations responsive to the evolving mask pattern.

To determine whether this positional signal is useful both when adapting existing checkpoints and when learning from scratch, we evaluate MDLMPE on LLaDA and DREAM. The matched 7B/8B post-training results summarized in Figure 2 show improvements across most knowledge, reasoning, and code tasks on both Tulu and SmolTalk, with broadly comparable performance on the remaining tasks. This consistency across model architectures and training datasets motivates our subsequent controlled evaluations, which cover 100M-parameter pretraining and zero-shot evaluation, block diffusion, and component ablations. Our contributions are threefold:

- We introduce MDLMPE, a positional encoding for MDLMs that explicitly models the evolving position-distribution structure induced by the revealed/masked token configuration.
- We identify the changing token-availability pattern as a distinctive challenge of MDLM decoding and construct a Gaussian-weighted, cosine-projected representation of this pattern.
- Extensive experiments on LLaDA and DREAM demon-

strate improvements over conventional positional encodings, while controlled ablations characterize how the MDLMPE components combine.

2 Related Work

Positional encoding. Position encoding provides the coordinate system through which attention distinguishes order (Vaswani et al. 2017). Existing designs instantiate this role in different ways: absolute encodings attach vectors to indices (Devlin et al. 2019); relative methods modify attention by pairwise displacement (Shaw, Uszkoreit, and Vaswani 2018; Dai et al. 2019; He et al. 2021); RoPE rotates queries and keys (Su et al. 2024); and ALiBi adds a head-specific distance bias (Press, Smith, and Lewis 2022). Position Interpolation and YaRN further extend pretrained coordinates to longer contexts (Chen et al. 2023; Peng et al. 2024).

Although these methods differ in form, they share a fixed-coordinate premise: an index is mapped to a vector, rotation, or bias independently of the current observation state. Extrapolation methods modify the rotary spectrum or training positions—xPos changes the parameterization (Sun et al. 2023), LongRoPE uses non-uniform rescaling (Ding et al. 2024), and PoSE trains on skipped indices (Zhu et al. 2024)—but retain this premise. LongLLaDA and UltraLLaDA likewise adapt RoPE for long-context diffusion models (Liu et al. 2025; He et al. 2025); MDLMPE addresses a different question by conditioning an in-window position on the observed availability state.

This distinction is largely hidden in AR decoding, where the causal mask and revealed prefix coincide, but becomes consequential in an MDLM that exposes both tokens and masks. An index-only encoding states that two positions are k steps apart without indicating whether either has nearby usable evidence. MDLMPE therefore retains RoPE's index relation while adding the observed/masked configuration to the residual stream and rotary coordinates.

Masked diffusion language modeling. In parallel, diffusion language modeling has progressed from extending continuous denoising (Ho, Jain, and Abbeel 2020; Nichol and Dhariwal 2021; Song et al. 2021) to categorical states (Austin et al. 2021a; Hoogeboom et al. 2021) toward increasingly capable text generators. Diffusion-LM, DiffuSeq, and iterative masked generation establish text settings (Li et al. 2022; Gong et al. 2023; Ghazvininejad et al. 2019; Gu, Wang, and Zhao 2019; Chang et al. 2022); SEDD, MD4, RADD, and TESS refine objectives or parameterizations (Lou, Meng, and Ermon 2024; Shi et al. 2024; Ou et al. 2025; Mahabadi et al. 2024); and LLaDA, DREAM, LLaDA2.0, and Seed Diffusion scale the approach (Nie et al. 2025; Ye et al. 2025; Bie et al. 2025; Song et al. 2025).

Together, these advances improve diffusion objectives, scale, and generation strategies, yet the denoiser still repeatedly infers missing tokens from a partially observed sequence. Diffusion time is usually encoded globally, whereas the spatial layout of the observed subset remains implicit in mask-token activations and attention. DiffusionBERT, Block Diffusion, and DFlash modify the objective, generation factorization, or speculative-drafting mechanism (He et al. 2023; Arriola et al. 2025; Chen, Liang, and Liu 2026). MDLMPE complements these directions: it leaves their generation machinery unchanged and adds a coordinate derived from the realized availability pattern.

3 Method

The preceding discussion identifies a representation gap: masked denoising needs positional signals that follow the changing mask state, distinguish both the amount and location of available evidence, and preserve the geometry of a RoPE-pretrained backbone. MDLMPE meets these requirements with a shared availability encoder and two complementary paths: token-centered features enrich the residual stream, while pair-centered features adjust selected rotary relations.

3.1 Availability-Aware Positional Representation

The first requirement is state sensitivity. Although an MDLM can identify individual `[MASK]` tokens, their spatial arrangement remains entangled with content. We expose this structure explicitly through a minimal content-free state. For $\boldsymbol{x} = (x_1, \dots, x_L)$, define

$$b_t = \mathbb{I}[x_t \neq x_{\text{mask}}]a_t, \qquad b_t \in \{0, 1\}, \tag{1}$$

where $a_t \in \{0, 1\}$ excludes attention-invalid positions. The signal therefore changes as soon as a token is revealed, without encoding lexical content or clean targets.

State alone, however, does not express how strongly each revealed position should influence a target. Because nearby and distant evidence need not be equally useful, a normalized truncated Gaussian converts the global mask pattern into a target-relative neighborhood:

$$w_{i,t} = \frac{\exp\left(-\frac{(i-t)^2}{2\sigma^2}\right)\mathbb{I}[|i-t| \leq W]a_t}{\sum_{u=1}^{L}\exp\left(-\frac{(i-u)^2}{2\sigma^2}\right)\mathbb{I}[|i-u| \leq W]a_u + \varepsilon}, \tag{2}$$

where W is the radius and $\sigma = \max(\rho W, 1)$. Normalization stabilizes boundaries; truncation controls locality and cost. We use $W/L = 3/16$ and $\rho = 1/4$ (e.g., $W = 192$, $\sigma = 48$ at $L = 1024$). The symmetric kernel models bidirectional evidence without a learned left/right preference.

Local weighting makes the representation target-relative, but a weighted count would still collapse equal-density layouts. We therefore preserve their topology on RoPE's multiscale frequency grid. For $F = d_h/2$ rotary pairs, let $c_{t,f} = \frac{1}{2}[1 + \sin(t\omega_f)]$ be a nonnegative shifted-sine coordinate and form

$$\begin{aligned} u_{i,f} &= \sum_{t=1}^{L} w_{i,t} b_t c_{t,f}, \\ d_{i,f} &= \sum_{t=1}^{L} w_{i,t} a_t c_{t,f} + \varepsilon, \qquad r_{i,f} = \frac{u_{i,f}}{d_{i,f}}. \end{aligned} \tag{3}$$

with $0 \leq r_{i,f} \leq 1$. Low frequencies summarize broad availability and higher frequencies retain finer changes, distinguishing the equal-density layouts in Figure 4.

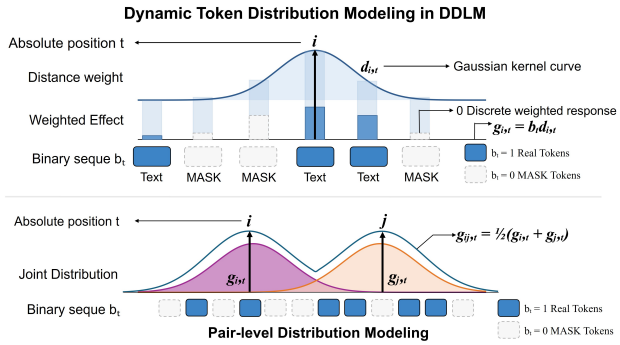

Figure 3: Gaussian local encoding for one position (top) and the pair-level distribution used by the rotary branch (bottom). The token-centered upper construction supplies the embedding branch. For an attention pair (i, j), the implemented rotary branch averages the two weighted sequences as $g_{ij,t} = \frac{1}{2}(g_{i,t} + g_{j,t})$.

Once this topology has been encoded, it must enter the model at the granularity required by each path. Let $\boldsymbol{u}_i = (u_{i,1}, \ldots, u_{i,F})$ and $\boldsymbol{d}_i = (d_{i,1}, \ldots, d_{i,F})$ denote the token-centered availability numerator and its all-valid reference. Token embeddings use $\boldsymbol{z}_i = \boldsymbol{W}_p \boldsymbol{u}_i + \boldsymbol{b}_p$ and $\widetilde{\boldsymbol{e}}_i = \boldsymbol{e}_i + \gamma \boldsymbol{z}_i$, with $\gamma \leq 0.10$ initialized to 0.01. For the pair branch, define $\boldsymbol{A}_i \equiv \boldsymbol{u}_i$ and $\boldsymbol{D}_i \equiv \boldsymbol{d}_i$. The symmetric pair feature is

$$\boldsymbol{r}_{ij} = g(\boldsymbol{A}_i, \boldsymbol{D}_i, \boldsymbol{A}_j, \boldsymbol{D}_j), \qquad [\boldsymbol{r}_{ij}]_f = \frac{A_{i,f} + A_{j,f}}{D_{i,f} + D_{j,f}}, \tag{4}$$

which is equivalent to $u_{ij,f} = \frac{1}{2}(u_{i,f} + u_{j,f})$, $d_{ij,f} = \frac{1}{2}(d_{i,f} + d_{j,f})$, and $r_{ij,f} = u_{ij,f}/d_{ij,f}$ in Figure 3. Thus $\boldsymbol{u}_i$ describes one token's surroundings, whereas $\boldsymbol{r}_{ij}$ summarizes the joint context used by an attention pair. Appendices A and F provide safeguards and deterministic analyses.

3.2 Availability-Conditioned Rotary Adaptation

The pair feature now captures joint context, but replacing RoPE angles with it would discard pretrained displacement geometry. MDLMPE instead uses this information as a bounded residual, beginning with the fixed angularization

$$\boldsymbol{s}_{ij} = \arccos(2\boldsymbol{r}_{ij} - \mathbf{1}) - \tfrac{\pi}{2}\mathbf{1}, \tag{5}$$

where $\arccos$ is applied elementwise. The inverse-cosine map spreads the normalized availability range over a stable angular domain, while the subsequent learnable transformation determines how this signal should interact across frequencies; $\boldsymbol{s}_{ij}$ is therefore an intermediate coordinate rather than a rotary angle. To make the high-level path in Figure 5 explicit, a shared hidden layer applies a linear transformation and SiLU activation to $\boldsymbol{s}_{ij}$, producing $\boldsymbol{\psi}_{ij} \in \mathbb{R}^H$, where $H = \max(16, F/2)$ by default. Each frequency-specific output head then maps $\boldsymbol{\psi}_{ij}$ to the phase residual $\delta_{ij,f} = h_f(\boldsymbol{\psi}_{ij})$:

$$h_f(\boldsymbol{\psi}_{ij}) = \tau_{\max} \tanh(\boldsymbol{w}_{2,f}^\top \boldsymbol{\psi}_{ij} + b_{2,f}). \tag{6}$$

Here $\tau_{\max} = 0.25$, $\boldsymbol{w}_{2,f}^\top$ is the fth row of the shared output layer, and the output heads are zero initialized. This

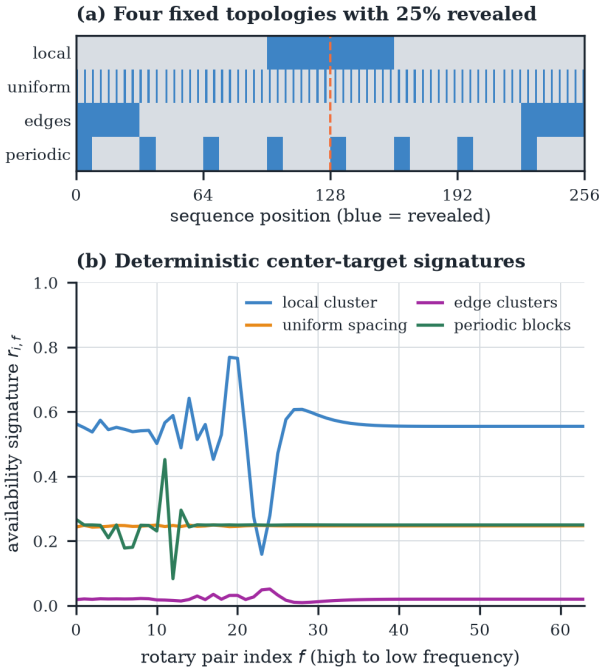

Figure 4: Equal-density availability patterns produce different deterministic MDLMPE signatures under the default geometry. Each length-256 mask reveals exactly 64 positions (25%); the dashed line marks the center target. With $W/L = 3/16$, $\sigma/W = 1/4$, and a fixed RoPE frequency grid, local, uniformly spaced, edge-clustered, and periodic arrangements yield distinct vectors $\{r_{i,f}\}_f$. The figure illustrates representational topology, not learned embeddings or benchmark outcomes.

gives the high-level computation $\boldsymbol{r}_{ij} \to \boldsymbol{\psi}_{ij} \to \delta_{ij,f}$. Let $\bar{\boldsymbol{q}}_{i,f} = \boldsymbol{R}(i\omega_f)\boldsymbol{q}_{i,f}$ and $\bar{\boldsymbol{k}}_{j,f} = \boldsymbol{R}(j\omega_f)\boldsymbol{k}_{j,f}$ denote the ordinary RoPE-rotated two-dimensional query and key pair. The adapted pair contribution is

$$S_{ij,f} = \begin{cases} \bar{\boldsymbol{q}}_{i,f}^\top \bar{\boldsymbol{k}}_{j,f}, & f \in \mathcal{F}_{\mathrm{RoPE}}, \\ \bar{\boldsymbol{q}}_{i,f}^\top \boldsymbol{R}(\delta_{ij,f}) \bar{\boldsymbol{k}}_{j,f}, & f \in \mathcal{F}_{\mathrm{MDLMPE}}, \end{cases} \tag{7}$$

with $|\mathcal{F}_{\mathrm{MDLMPE}}| \approx F/2$ and $S_{ij} = \sum_f S_{ij,f}/\sqrt{d_h}$. Equivalently, the angular relation used by attention is

$$\Delta\phi_{ij,f} = \begin{cases} (j-i)\omega_f, & f \in \mathcal{F}_{\mathrm{RoPE}}, \\ (j-i)\omega_f + \delta_{ij,f}, & f \in \mathcal{F}_{\mathrm{MDLMPE}}. \end{cases} \tag{8}$$

The construction thus preserves the familiar displacement relation and augments it only where the observed context provides additional structure. Every adapted pair retains RoPE's displacement term while adding joint-context information; clipping $\arccos$ prevents endpoint singularities, and bounding $\boldsymbol{\delta}_{ij}$ limits the perturbation.

3.3 Checkpoint-Compatible Integration

The preceding construction is useful in large backbones only if its additional expressivity can be introduced without disturbing the pretrained interface. LLaDA and DREAM therefore share one module that builds availability from token

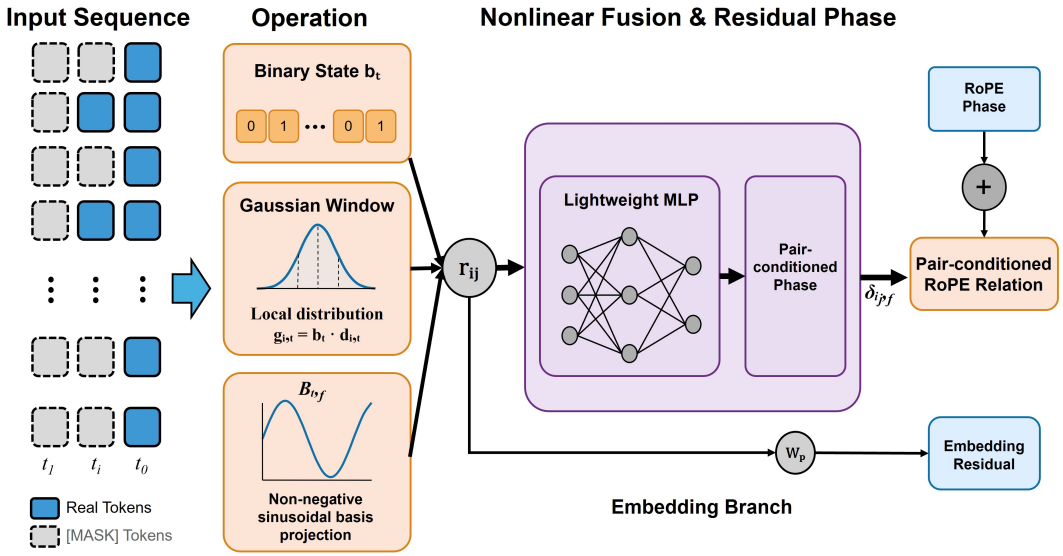

Figure 5: System-level view of the high-level two-branch architecture. The token feature $\boldsymbol{u}_i$ supplies the enabled embedding branch, while the pair feature $\boldsymbol{r}_{ij}$ supplies the shared MLP and bounded RoPE residual; half of the interleaved rotary pairs remain ordinary RoPE. The pair feature $\boldsymbol{r}_{ij}$, its MLP transformation $\boldsymbol{\psi}_{ij}$, and the resulting phase residual $\delta_{ij,f}$ are defined in the main text.

IDs and the validity mask, computes Gaussian summaries once per forward pass, and reuses them across layers. Hidden size, tokenizer, denoising schedule, and output head are unchanged.

Because pair-conditioned phases are evaluated inside attention, fused kernels require an MDLMPE-aware extension; the reference implementation evaluates these phases in query blocks. Local aggregation costs $O(LWF)$, the enabled embedding projection costs $O(LFd)$, and pair-phase evaluation adds $O(L^2FH)$ for the shared map before the selected-frequency score update. A query block of size C bounds extra working storage by $O(CLF)$.

Frequency and head allocation. Within this implementation, one availability representation is shared across the model rather than divided into four view groups. Rotary pairs alternate between RoPE and MDLMPE, giving an approximately 50/50 split: unchanged pairs preserve a direct checkpoint path, while interleaving covers the frequency range. Training from scratch and inference activate all key–value heads. During instruction post-training, the active fraction rises linearly from zero at 10% of updates to one at 90%; selected heads are distributed approximately uniformly, and grouped query heads follow their key–value head.

Implementation details. Numerical safeguards preserve this gradual integration in practice. Availability is constructed before rotary embedding and excludes invalid positions. The basis and Gaussian are fixed and symmetric; denominators and arccos inputs are clamped for mixed precision. Zero-initializing the phase-MLP output starts the rotary path exactly at RoPE, while the bounded embedding gate controls the second residual path.

Finally, the availability signal is always recomputed from the information actually visible to the denoiser. During pretraining, $\boldsymbol{b}$ is obtained from the sampled corruption state; during instruction post-training, it includes the prompt and revealed response tokens but excludes masked response targets. The same operation is repeated after each inference update. Training and inference therefore expose MDLMPE to the same information pattern, preventing the positional feature from leaking clean responses.

3.4 Compatibility with Blockwise Denoising

Having defined the encoder independently of a particular sampler, we next consider whether the same representation remains meaningful under blockwise generation. If q_s is the active-block boundary and $\mathcal{P}$ the prompt, blocked context exposes $\mathcal{P} \cup \{t : t \le q_s\}$, whereas the LLaDA-style regime exposes the full partially denoised response. MDLMPE encodes no block ID; it recomputes $\boldsymbol{b}$ from the state presented to attention and applies the same dual-path construction. It therefore supports both factorizations without changing the objective, schedule, or update rule.

4 Experiments

4.1 Experimental Design

The method makes claims at several levels—checkpoint compatibility, from-scratch positional bias, robustness to changing visibility, and component interaction—so we organize the evaluation around these four questions. For checkpoint adaptation, we conduct instruction post-training of LLaDA-8B and DREAM-7B on Tulu-3 and SmolTalk, using supervised fine-tuning (SFT) as the optimization recipe (Lambert et al. 2024; Allal et al. 2025). To isolate the positional inductive bias, we pretrain matched 100M-parameter models on Open-

Metric	LLaDA-8B				DREAM-7B			
	T RoPE	T MDLMPE	S RoPE	S MDLMPE	T RoPE	T MDLMPE	S RoPE	S MDLMPE
MMLU (5)	60.3	**60.5**	**61.1**	60.7	62.2	**63.8**	**64.3**	63.6
MMLU-Pro (0)	41.8	**42.3**	39.2	**41.4**	**45.1**	44.5	42.2	**43.6**
HellaSwag (0)	68.6	**71.4**	**70.4**	69.3	**67.5**	67.4	71.4	**73.1**
ARC-C (0)	60.7	**63.2**	58.3	**59.9**	65.3	**66.6**	62.1	**64.0**
GSM8K (4)	67.4	**70.1**	68.2	**68.5**	**71.2**	70.8	70.0	**70.5**
MATH (4)	**30.2**	29.3	31.7	**32.2**	35.5	**36.4**	32.8	**33.6**
HumanEval (0)	42.3	**44.6**	38.6	**39.3**	50.4	**52.8**	**51.6**	51.4
MBPP (4)	**40.7**	40.3	**39.6**	39.4	42.4	**43.0**	43.5	**43.8**

Table 1: Instruction post-training results on Tulu (T) and SmolTalk (S), transposed for metric-wise comparison. Every RoPE/MDLMPE column pair starts from the same RoPE-pretrained checkpoint.

WebText (OWT) for 200K steps at length 1,024 (Gokaslan et al. 2019). Component analyses use LLaDA-7B with 50K/2K Tulu train/test examples. All training uses $8\times$ A100 80GB GPUs.

To make these questions comparable, RoPE (Su et al. 2024) serves as the primary baseline, while ALiBi (Press, Smith, and Lewis 2022) and absolute embeddings provide controlled-pretraining diagnostics. Within each comparison, tokenizer, corruption, token budget, optimizer, decoding, and normalization are fixed. Post-training activates MDLMPE heads progressively from 10% to 90% of updates; from-scratch models use it throughout. Appendix B specifies the complete protocol.

4.2 Instruction Post-Training at 7B/8B Scale

With the evaluation protocol fixed, we first ask whether a state-dependent coordinate can be introduced after pretraining while preserving the backbone's learned geometry. The matched results in Table 1 show improvements across knowledge, reasoning, and code tasks. On LLaDA–Tulu, HellaSwag increases from 68.6 to 71.4 and GSM8K from 67.4 to 70.1. On DREAM–Tulu, MATH increases from 35.5 to 36.4 and HumanEval from 50.4 to 52.8; DREAM–SmolTalk also improves HellaSwag from 71.4 to 73.1. Since each comparison shares its starting checkpoint, data order, scheduler, decoding budget, and selection rule, these gains demonstrate that MDLMPE can add availability structure effectively through instruction post-training.

4.3 Controlled Pretraining and Cross-Corpus Evaluation

Post-training establishes checkpoint compatibility, but from-scratch pretraining asks the complementary question of whether availability awareness is useful as an inductive bias in its own right. We therefore train matched 100M-parameter models and report pretraining-validation and zero-shot perplexity in Table 2. For LLaDA-100M, MDLMPE obtains 229.80 on 1BW compared with 239.34 for RoPE, 203.65 on Text8 compared with 214.31, and 26.13 on OWT compared with 27.08. DREAM-100M shows particularly clear improvements on PTB, from 97.05 to 81.24, and on WikiText-2, from 63.74 to 58.82; its WikiText-103 perplexity also reaches 61.64. Together, the in-domain and external-corpus results show that availability-aware positioning contributes during pretraining and carries into zero-shot evaluation across different document structures.

Corpus	RoPE	ALiBi	MDLMPE
LLaDA-100M			
1BW	239.34	273.42	**229.80**
PTB	144.55	**135.60**	170.18
Text8	214.31	243.30	**203.65**
WT103	**103.12**	107.09	104.84
WT2	71.07	87.18	**68.10**
OWT	27.08	26.41	**26.13**
DREAM-100M			
1BW	196.50	198.73	**194.54**
PTB	97.05	101.53	**81.24**
Text8	116.94	**103.46**	107.10
WT103	64.49	62.79	**61.64**
WT2	63.74	61.28	**58.82**
OWT	52.37	52.64	**52.29**

Table 2: Controlled pretraining-validation and zero-shot perplexity for 100M-parameter LLaDA and DREAM models. OWT is the training distribution; the remaining corpora evaluate cross-corpus transfer.

4.4 Robustness under Blockwise Post-Training

Beyond the training regime, blockwise generation changes the visibility structure that MDLMPE is designed to represent. Varying the active block determines which response positions are jointly available and thus directly tests topology sensitivity. The two visibility patterns are illustrated in Figure 6, while each comparison in Table 3 shares its checkpoint, prompt, block rule, denoising budget, and extractor. Block size changes $r_{ij,f}$ through the mask state; MDLMPE receives no block ID and leaves the update rule unchanged.

Under this controlled change in visibility, the block-size sweep provides concrete evidence across both backbones. For DREAM-7B at block size 32, HellaSwag increases from 65.1 to 67.7 and ARC-Challenge from 62.8 to 63.3. At block

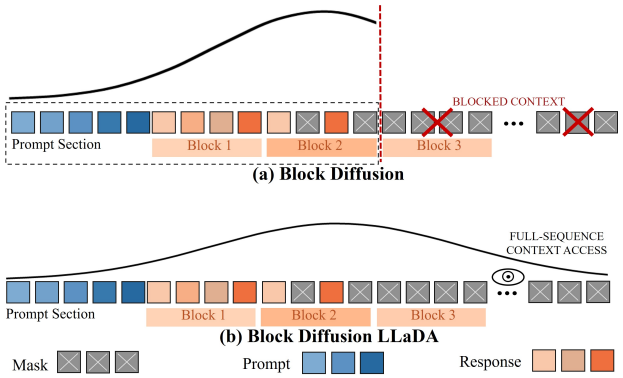

Figure 6: Blocked-context and full-sequence block diffusion. MDLMPE encodes the visibility pattern presented to attention.

Pos.	B	MMLU	ARC	Hella.
DREAM-7B				
RoPE	16	61.3	63.4	66.0
MDLMPE	16	61.8	63.2	67.1
RoPE	32	61.7	62.8	65.1
MDLMPE	32	61.9	63.3	**67.7**
RoPE	64	60.9	63.1	66.6
MDLMPE	64	**62.0**	**63.3**	65.9
LLaDA-8B				
RoPE	16	60.1	60.7	68.4
MDLMPE	16	**60.4**	60.1	68.4
RoPE	32	59.5	60.2	67.4
MDLMPE	32	59.9	60.8	67.9
RoPE	64	59.6	60.0	68.2
MDLMPE	64	60.3	**60.9**	**68.6**

Table 3: Matched block-size results. B denotes block size; all scores are accuracies (%). ARC denotes ARC-Challenge. Within each model panel, data, decoding budget, and evaluation harness are fixed.

Bin.	Gauss.	Basis	Emb.	Train loss↓	Eval PPL↓
–	–	–	–	3.159	17.24
✓	–	–	–	3.193	17.67
✓	✓	–	–	3.154	16.88
✓	✓	✓	–	3.163	17.05
✓	✓	✓	✓	**3.131**	**16.13**

Table 4: Component analysis during LLaDA-7B instruction post-training on 50K Tulu examples, evaluated on 2K examples. The first row is RoPE (Su et al. 2024); the final row is the dual-path MDLMPE method.

size 64, DREAM reaches 62.0 on MMLU and 63.3 on ARC-Challenge. LLaDA-8B at block size 64 improves MMLU from 59.6 to 60.3, ARC-Challenge from 60.0 to 60.9, and HellaSwag from 68.2 to 68.6. Because the same encoder is used without a block identifier, these improvements show that MDLMPE responds directly to the visibility pattern produced by different block configurations.

4.5 Component Analysis

Finally, we connect the end-to-end gains back to the structure of the method by examining how its components combine. The incremental configurations in Table 4 introduce binary availability, Gaussian locality, the shifted-sine basis, and the embedding path in sequence. The reported final row corresponds to the complete dual-path MDLMPE method; the intermediate rows characterize component interactions rather than implying that every added component improves the metric in isolation.

The progression in Table 4 provides a direct link between the representation design and its optimization behavior. Under matched data order, seed, and checkpoint rules, the Gaussian configuration obtains a training loss of 3.154 and an evaluation perplexity of 16.88. The complete dual-path configuration achieves the strongest reported values, with a training loss of 3.131 and an evaluation perplexity of 16.13. These results support combining target-relative Gaussian structure, the multiscale basis, and the two feature-injection paths in the main method. Appendix E further links the aggregate ARC, MMLU, and GSM8K metrics to paired generations.

5 Conclusion

This work introduced MDLMPE, which, to our knowledge, is the first distribution-aware positional encoding method designed for masked diffusion language models. To address the dynamically changing and non-contiguous distributions of revealed and masked positions during denoising, MDLMPE builds upon RoPE to represent token availability through binary availability states, Gaussian local weighting, and a RoPE-aligned sinusoidal basis transformation. The resulting features are incorporated into token embeddings and selected rotary phases. Experiments on LLaDA and DREAM, including supervised fine-tuning of 7B/8B models and pretraining and zero-shot evaluation at the 100M scale, demonstrate that MDLMPE generally outperforms RoPE. These results indicate that dynamic token distributions provide MDLMs with useful positional cues beyond absolute position indices, highlighting the research value and practical potential of MDLMPE. Future work will further evaluate its pretraining effectiveness and scalability on larger MDLMs, explore more effective approaches to token-position-distribution modeling, positional encoding, and feature injection, and examine its generalization across a broader range of model architectures and data distributions.

Supplementary Material

Overview

- **Appendix A: Detailed formulation.**
  - **A.1 RoPE-aligned sinusoidal coordinates** formalizes the revealed-token indicator and the paired sinusoidal reference aligned with the RoPE frequency grid.
  - **A.2 Embedding-level injection** characterizes residual-stream conditioning by availability-distribution features, its complementarity to rotary attention, and its information-independence and stability properties.
  - **A.3 RoPE-aligned sinusoidal basis** examines the shared frequency grid, linear projection, spectral properties, and associated limitations of the fixed coordinate basis.
  - **A.4 Local availability ratios and rotary injection** derives the symmetric Gaussian kernel, boundary-normalized mask-conditioned aggregation, pair availability ratio, and bounded pair-conditioned phase adaptation.
  - **A.5 Safeguards, variants, and complexity** specifies mixed-precision safeguards, initialization and frequency-allocation strategies, cacheable quantities, and asymptotic computational costs.
- **Appendix B: Extended experimental protocol.**
  - **B.1 Data and availability states** specifies the evaluated MDLM families, parameter scales, training corpora, and construction of denoising-state availability indicators.
  - **B.2 Datasets and benchmarks** documents the eight downstream benchmarks and the corpora used for perplexity evaluation.
  - **B.3–B.4 Training controls** details the architectures, optimization configurations, hardware settings, and decoding protocols for scratch training and supervised fine-tuning.
  - **B.5 Metrics and reporting** defines the positional baselines, accuracy and perplexity metrics, ablation controls, and systems-level measurements.
- **Appendix C: Additional ablations.**
  - **C.1–C.3** examine the Gaussian-window ratio, Gaussian bandwidth, and the respective contributions of the embedding projection and sinusoidal basis.
- **Appendix D: Efficiency analysis.**
  - Appendix D derives analytical bounds on the computational and storage overhead of MDLMPE.
- **Appendix E: Qualitative output comparisons.**
  - Three tables present paired ARC, MMLU, and GSM8K examples, followed by a consolidated qualitative analysis.
- **Appendix F: Deterministic mechanism analysis.** Formula-aligned deterministic evaluations establish the exact availability-ratio envelope, frequency–distance transfer, local Jacobian bound, intermediate-coordinate sensitivity, and boundary-normalization invariants independently of any sampled token-recovery trajectory or empirical performance claim.

A Detailed Formulation

A.1 RoPE-Aligned Sinusoidal Coordinates

For a batch element with length L, let x_t be the token at position t, x_{mask} the mask token, and a_t the attention-validity indicator. The revealed-token indicator is

$$b_t = \mathbb{I}[x_t \neq x_{\text{mask}}]a_t, \qquad b_t \in \{0, 1\}. \tag{9}$$

For head dimension d_h, there are $F = d_h/2$ rotary pairs. We use RoPE's inverse-frequency grid, $\omega_f = \theta^{-2f/d_h}$, and retain the paired sinusoidal reference coordinate

$$\boldsymbol{p}_{t,f} = \big[\sin(t\omega_f),\ \cos(t\omega_f)\big]. \tag{10}$$

The main availability coordinate is the nonnegative, phase-shifted form $c_{t,f} = \frac{1}{2}[1 + \sin(t\omega_f)]$. The paired expression documents the underlying RoPE geometry, while the implemented ratio uses the nonnegative coordinate. The next two subsections motivate embedding injection and the shared RoPE spectrum.

A.2 Embedding-Level Conditioning of Availability Distributions

Let $\boldsymbol{e}_i \in \mathbb{R}^d$ denote the token embedding at position i, $\boldsymbol{u}_i \in \mathbb{R}^F$ the unprojected availability aggregate in Eq. 18, and $\boldsymbol{z}_i = \boldsymbol{W}_p\boldsymbol{u}_i + \boldsymbol{b}_p \in \mathbb{R}^d$. We inject the projected feature through a gated residual path,

$$\widetilde{\boldsymbol{e}}_i = \boldsymbol{e}_i + \gamma\boldsymbol{z}_i, \tag{11}$$

where $0 \leq \gamma \leq \gamma_{\max} = 0.10$ is a learned bounded gate initialized to 0.01. If $\|\boldsymbol{z}_i\|_2 \leq B_z$, then the deviation from the pretrained input satisfies

$$\big\|\widetilde{\boldsymbol{e}}_i - \boldsymbol{e}_i\big\|_2 = \gamma\|\boldsymbol{z}_i\|_2 \leq \gamma_{\max}B_z. \tag{12}$$

Thus, setting $\gamma = 0$ exactly recovers the baseline embedding, while near-zero initialization makes the initial adaptation a controlled perturbation rather than a replacement of the pretrained residual stream.

Embedding injection makes this state information available to every first-layer linear path:

$$\begin{bmatrix}\boldsymbol{q}_i\\ \boldsymbol{k}_i\\ \boldsymbol{v}_i\end{bmatrix} = \begin{bmatrix}\boldsymbol{W}_Q\\ \boldsymbol{W}_K\\ \boldsymbol{W}_V\end{bmatrix}\widetilde{\boldsymbol{e}}_i = \begin{bmatrix}\boldsymbol{W}_Q\\ \boldsymbol{W}_K\\ \boldsymbol{W}_V\end{bmatrix}\boldsymbol{e}_i + \gamma\begin{bmatrix}\boldsymbol{W}_Q\\ \boldsymbol{W}_K\\ \boldsymbol{W}_V\end{bmatrix}\boldsymbol{z}_i. \tag{13}$$

Rotary injection directly changes query–key geometry, whereas Eq. 13 also exposes the local state to value and subsequent feed-forward paths. Consequently, two denoising states with the same token and absolute index but different revealed neighborhoods can receive different inputs before attention chooses a key.

The residual complements rather than duplicates the `[MASK]` embedding. Let y_t be the underlying clean token and let $\mathcal{B}(b_{1:L})$ denote the Gaussian aggregation and projection that produces $\boldsymbol{z}_i$. Since it uses only the binary state in Eq. 9,

$$\boldsymbol{z}_i = \mathcal{B}(b_{1:L}), \qquad \mathcal{B}(b_{1:t-1}, 0, b_{t+1:L}) \text{ is invariant to } y_t. \tag{14}$$

The feature therefore supplies availability geometry, while $\boldsymbol{e}_i$ and the content-dependent projections retain token semantics. In particular, it cannot encode the identity of an unrevealed target; the same construction is applied to the current corruption state during training and to the current denoising state during inference.

The possible adverse effects are correspondingly bounded. A transient pattern changes only the residual term in Eq. 11; the bounded gate, normalized aggregation, and per-step recomputation let optimization attenuate it. Confusing availability with semantic importance would require the binary

feature to replace content, which Eq. 14 excludes and Eq. 13 does not impose. Finally, ordinary RoPE pairs and zero-initialized phase residuals preserve explicit compatibility paths. Appendix C tests whether this residual information improves results beyond matched parameter counts.

A.3 RoPE-Aligned Spectral Basis for Availability Projection

For frequency f, the main coordinate is a nonnegative phase-shifted cosine, equivalently written as a shifted sine,

$$\begin{aligned} c_{t,f} &= \frac{1}{2}\big[1+\cos(t\omega_f - \tfrac{\pi}{2})\big] \\ &= \frac{1}{2}\big[1+\sin(t\omega_f)\big], \end{aligned} \qquad 0 \le c_{t,f} \le 1, \tag{15}$$

where $\omega_f = \theta^{-2f/d_h}$ is precisely the frequency used by the corresponding RoPE pair. Given the normalized nonnegative weights from Eq. 22, the availability mass and its all-position reference are

$$\begin{aligned} A_{i,f} &= \sum_{t=1}^{L} w_{i,t} b_t c_{t,f}, \\ D_{i,f} &= \sum_{t=1}^{L} w_{i,t} a_t c_{t,f} + \varepsilon, \qquad r_{i,f} = \frac{A_{i,f}}{D_{i,f}}. \end{aligned} \tag{16}$$

Because $b_t \in \{0,1\}$ and $w_{i,t}, c_{t,f} \ge 0$, the numerator is bounded by the unclamped denominator:

$$0 \le A_{i,f} \le \sum_{t=1}^{L} w_{i,t} a_t c_{t,f} \le D_{i,f}, \qquad 0 \le r_{i,f} \le 1. \tag{17}$$

This is why the main ratio uses a shifted coordinate instead of a signed sinusoid: signed contributions could cancel, so a small aggregate would not distinguish scarce revealed evidence from balanced positive and negative phases.

Stacking the coordinates gives $\boldsymbol{c}_t = [c_{t,0}, \dots, c_{t,F-1}]^\top$. The feature before and after the learned linear projection is

$$\boldsymbol{u}_i = \sum_{t=1}^{L} w_{i,t} b_t \boldsymbol{c}_t, \qquad \boldsymbol{z}_i = \boldsymbol{W}_p \boldsymbol{u}_i + \boldsymbol{b}_p. \tag{18}$$

Hence a one-position change at t has the exact influence

$$\frac{\partial \boldsymbol{u}_i}{\partial b_t} = w_{i,t}\boldsymbol{c}_t, \qquad \left\| \frac{\partial \boldsymbol{u}_i}{\partial b_t} \right\|_2 \le w_{i,t}\sqrt{F}. \tag{19}$$

The Gaussian weight therefore localizes the effect of a state change, while the fixed vector $\boldsymbol{c}_t$ makes $\boldsymbol{W}_p$ a learned mixture of multiscale spatial summaries. Low-frequency channels describe broad availability imbalance; higher-frequency channels respond to local revealed–masked transitions. The computation remains a weighted vector sum and does not materialize an $L \times L \times F$ tensor.

The alignment with RoPE is structural rather than cosmetic. The original displacement phase $(j-i)\omega_f$, basis coordinate $c_{t,f}$, pair coordinate $s_{ij,f}$, and final residual $\delta_{ij,f}$ are indexed by the same frequency ordering. Thus the model preserves the spectrum represented by a RoPE-pretrained checkpoint instead of introducing an unrelated position scale. It also permits the availability coordinates and Gaussian buffers to reuse the RoPE frequency cache.

Using a periodic basis does not make the encoding the sole source of absolute position. Let $\boldsymbol{R}(\alpha)$ denote the standard 2×2 rotation by angle α, and write $\alpha_{ij,f} = (j-i)\omega_f$. For any selected rotary pair, the distance between the baseline and residual-conditioned rotation matrices is bounded by

$$\begin{aligned} \|\boldsymbol{R}(\alpha_{ij,f} + \delta_{ij,f}) - \boldsymbol{R}(\alpha_{ij,f})\|_2 &= 2\left|\sin\left(\frac{\delta_{ij,f}}{2}\right)\right| \\ &\le 2\sin(\tau_{\max}/2). \end{aligned} \tag{20}$$

Multiple frequencies, local support, and the retained ordinary RoPE pairs resolve periodic ambiguity, while Eq. 20 applies directly to the main pair-conditioned residual with $\tau_{\max} = 0.25$ radians.

Alignment can correlate the residual with the original phase, and a fixed basis is less flexible than a fully learned coordinate map. These are controlled by the unchanged RoPE pairs, the bounded residual, and the shared cross-frequency phase MLP. The embedding projection $\boldsymbol{W}_p$ is separate from this phase MLP. Appendix C compares the fixed and learnable alternatives with matched accounting; the selected basis is therefore a bounded inductive bias with an explicit baseline path rather than an irreversible change to positional geometry.

A.4 Local Availability Ratios and Rotary Injection

Let W be the window radius and $\sigma = \max(\rho W, 1)$, where the defaults are $W/L = 3/16$ and $\rho = 1/4$. Before normalization, the symmetric weight from target position i to source position t is

$$\begin{aligned} \widetilde{w}_{i,t} = \exp\left[-\frac{(i-t)^2}{2\sigma^2}\right] \\ \times \mathbb{I}[|i-t| \le W] a_t, \end{aligned} \tag{21}$$

where a_t removes padding or otherwise invalid attention positions. The main method has no learned left/right scalar. The row-normalized matrix is

$$w_{i,t} = \frac{\widetilde{w}_{i,t}}{\sum_{u=1}^{L} \widetilde{w}_{i,u} + \varepsilon}. \tag{22}$$

Normalization makes a ratio insensitive to positions retained solely because of boundary truncation. Masked positions are suppressed before the nonnegative basis aggregation in Eq. 16. The token-centered matrices $\boldsymbol{A}$ and $\boldsymbol{D}$ are reused by both branches. For an attention pair (i,j), the rotary branch forms the symmetric feature

$$[\boldsymbol{r}_{ij}]_f = \frac{A_{i,f} + A_{j,f}}{D_{i,f} + D_{j,f}}, \qquad 0 \le [\boldsymbol{r}_{ij}]_f \le 1. \tag{23}$$

This ratio is equivalent to averaging the two token-centered numerator sequences and denominator references before division. It is mapped to a centered intermediate coordinate by

$$\boldsymbol{s}_{ij} = \arccos(2\boldsymbol{r}_{ij} - \mathbf{1}) - \tfrac{\pi}{2}\mathbf{1}. \tag{24}$$

The intermediate vector is fused across frequencies by the shared MLP,

$$\begin{aligned} \boldsymbol{\psi}_{ij} &= \mathrm{SiLU}(\boldsymbol{W}_1 \boldsymbol{s}_{ij} + \boldsymbol{b}_1)\,, \\ \boldsymbol{\delta}_{ij} &= \tau_{\max} \tanh(\boldsymbol{W}_2 \boldsymbol{\psi}_{ij} + \boldsymbol{b}_2)\,, \end{aligned} \tag{25}$$

where $\tau_{\max} = 0.25$, the hidden width is $H = \max(16, F/2)$, and $(\boldsymbol{W}_2, \boldsymbol{b}_2)$ are initialized to zero. The angular relation used by attention is

$$\Delta\phi_{ij,f} = \begin{cases} (j-i)\omega_f, & f \in \mathcal{F}_{\mathrm{RoPE}}, \\ (j-i)\omega_f + \delta_{ij,f}, & f \in \mathcal{F}_{\mathrm{MDLMPE}}. \end{cases} \tag{26}$$

The two frequency sets alternate and each contains approximately half of the rotary pairs. The implementation stores the token-centered $\boldsymbol{A}, \boldsymbol{D} \in \mathbb{R}^{L \times F}$ summaries once per example; pair ratios and MLP outputs are evaluated in query blocks inside attention, so the full $L \times L \times F$ pair tensor is not materialized.

A.5 Safeguards, Variants, and Complexity

The ratio denominator is clamped by ε, and the argument of $\arccos$ is clipped away from ± 1 before mixed-precision evaluation. These safeguards prevent division by zero and endpoint-gradient instabilities. The zero-initialized final MLP layer makes $\boldsymbol{\delta}_{ij} = \mathbf{0}$ at initialization, exactly recovering RoPE in the rotary branch. A learnable nonnegative affine basis can replace the fixed coordinate for an ablation, but the main method keeps the aligned fixed basis. With a full window, aggregation costs $O(L^2F)$; a radius-W grouped-convolution or banded reduction costs $O(LWF)$. The enabled embedding projection costs $O(LFd)$, and pair-conditioned phase evaluation costs $O(L^2FH)$. Geometry buffers depend only on (L, W, ρ, θ) and are cached; the token-centered state is computed once per model forward and reused by all layers, while pair phases are evaluated in query blocks with $O(CLF)$ extra working storage for query-block size C.

B Extended Experimental Protocol

B.1 Data and Availability-State Construction

We evaluate LLaDA (Nie et al. 2025) and DREAM (Ye et al. 2025) at two scales: matched 100M models trained from scratch and RoPE-pretrained LLaDA-8B/DREAM-7B checkpoints adapted by SFT. Pretraining uses OpenWebText (OWT) (Gokaslan et al. 2019); SFT uses Tulu-3 (Lambert et al. 2024) and SmolTalk (Allal et al. 2025). All ablations use LLaDA-7B on Tulu SFT with 50,000 training examples and 2,000 test examples. All training experiments use $8\times$ NVIDIA A100 80GB GPUs. Each example is tokenized once by the native tokenizer of the evaluated checkpoint, then padded, truncated, and corrupted under the same rule for all compared encodings. At a denoising step, the availability indicator is reconstructed from the actual model input: prompt tokens and revealed response tokens have value one, masked response tokens have value zero, and padding positions are excluded by the attention-validity mask. This avoids treating a nominal diffusion timestep as a substitute for the realized revealed/masked arrangement.

B.2 Datasets and Evaluation Benchmarks

Main-paper Table 1 groups eight downstream benchmarks into three categories. General tasks comprise MMLU (Hendrycks et al. 2021a), MMLU-Pro (Wang et al. 2024), HellaSwag (Zellers et al. 2019), and ARC-Challenge (Clark et al. 2018). Math and reasoning tasks comprise GSM8K (Cobbe et al. 2021) and Hendrycks MATH (Hendrycks et al. 2021b). Code tasks comprise HumanEval (Chen et al. 2021) and MBPP (Austin et al. 2021b).

Main-paper Table 2 reports perplexity on the One Billion Word Benchmark (1BW) (Chelba et al. 2014), Penn Treebank (PTB) (Marcus, Santorini, and Marcinkiewicz 1993), Text8 (Mahoney 2011), and WikiText-2 and WikiText-103 (Merity et al. 2017), with OWT (Gokaslan et al. 2019) used for pretraining and in-domain validation. These references identify the data sources; all compared encodings use identical preprocessing and evaluation within each matched experiment.

B.3 Matched Scratch-Training Protocol

The 100M models use hidden size 768, 12 layers, 12 attention heads, four key-value heads, a 126,464-token vocabulary, RMSNorm, SiLU, tied embeddings, and maximum length 1,024. OWT pretraining runs for 200K steps with learning rate 1×10^{-4}, effective batch size 16, AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0.01), cosine decay, and 2K learning-rate-warmup steps. This optimizer schedule does not activate the MDLMPE head-level warmup used for SFT; MDLMPE is active from the first pretraining update. The RoPE baseline and every MDLMPE variant start from the same architecture and initialization protocol. A matched run changes only the position module; data order, token budget, optimizer states, sequence length, corruption sampler, and validation cadence remain fixed. This control is important because a change in mask sampling can itself change diffusion loss even when the model weights are unchanged.

B.4 SFT Adaptation and Decoding Controls

SFT uses learning rate 2×10^{-5}, five epochs, prompt-loss masking, group-by-length batches, and bfloat16 FSDP. RoPE and MDLMPE start from the same source checkpoint and use the same backbone training policy, learning-rate scheduler, stopping rule, and checkpoint-selection criterion. The projection, gate, and phase-MLP parameters are present only in MDLMPE and are included in its parameter accounting. As part of checkpoint-compatible MDLMPE integration, its rotary heads are activated progressively; this method-specific head schedule does not alter the optimizer, data order, token budget, or decoding controls. For total update count N and update n, the active MDLMPE-head fraction is

$$\alpha(n) = \begin{cases} 0, & n/N < 0.1, \\ \dfrac{n/N - 0.1}{0.8}, & 0.1 \le n/N < 0.9, \\ 1, & n/N \ge 0.9. \end{cases} \tag{27}$$

Thus SFT begins with RoPE, linearly activates MDLMPE on key–value heads between 10% and 90% of training, and finishes with MDLMPE active on all heads. For an active

count h, the selected key–value indices are approximately uniform over $[0, H_{\mathrm{KV}} - 1]$ rather than always taking a prefix; grouped query heads inherit the state of their associated key–value head. The warmup is disabled for 100M scratch pretraining, where MDLMPE is active throughout, and evaluation/inference uses all heads. Prompt template, candidate order, generation budget, denoising schedule, seed policy, and answer normalization are fixed at evaluation. The availability feature is recomputed after each iterative update from the current corrupted sequence.

B.5 Metrics and Evaluation Harness

RoPE (Su et al. 2024) is the primary baseline; ALiBi (Press, Smith, and Lewis 2022) and absolute embeddings are diagnostics at 100M. Perplexity is computed with each model's native diffusion objective under matched masking schedules, reporting the same token subset for every encoding. Multiple-choice accuracy uses an identical prompt template, candidate serialization, decoding budget, answer parser, and invalid-output treatment within a matched comparison. The benchmark suite covers language modeling and knowledge, scientific, arithmetic, and commonsense decisions; PPL is therefore interpreted together with task accuracy rather than as a sufficient proxy for downstream utility.

B.6 Reporting, Ablations, and Systems Measurement

Component and hyperparameter ablations retain the same parameter accounting unless the purpose is explicitly to test added learnable capacity. Throughput and memory are measured only after warmup with fixed sequence length, batch size, dtype, attention implementation, device, and denoising steps; peak allocated memory is reset before each measurement, and throughput is averaged over a fixed interval. A configuration is described as robust only when its direction agrees across the matched language-modeling and task-level metrics, while a quality–systems trade-off is reported as such rather than being folded into a single score.

C Additional Ablations

All studies use LLaDA-7B on the Tulu SFT split with 50,000 training and 2,000 test examples and report training loss and evaluation perplexity.

C.1 Window Ratio

The window ratio governs global coverage versus local emphasis. Holding other parameters fixed, we compare progressively local W/L settings in Table 5.

Window ratio	Train loss↓	Eval PPL↓
1	3.152	16.77
3/4	3.155	16.72
1/2	3.143	16.59
1/4	3.136	16.38
3/16	**3.131**	**16.13**
1/8	3.134	16.26

Table 5: Gaussian-window-ratio ablation.

Table 5 indicates that a window ratio of $3/16$ yields the lowest training loss and evaluation perplexity among the tested settings, while more global and more local windows both incur higher perplexity.

C.2 Gaussian Standard Deviation

The Gaussian bandwidth governs positional decay within the retained window. We vary ρ in $\sigma = \rho W$ while holding W fixed; Table 6 compares concentrated and smoother local weighting.

ρ	Train loss↓	Eval PPL↓
0.20	3.142	16.33
0.25	**3.131**	**16.13**
0.33	3.132	16.21
0.50	3.144	16.40

Table 6: Gaussian-bandwidth ablation.

Table 6 identifies $\rho = 0.25$ as the best setting among the tested bandwidths for both reported metrics.

C.3 Embedding Projection and Sinusoidal Basis

The RoPE-aligned sinusoidal basis and embedding projection $\boldsymbol{W}_p$ may contribute independently. We therefore evaluate a 2×2 factorial design while retaining the same binary state, Gaussian weights, and pair-conditioned rotary branch. When the sinusoidal basis is disabled, the positional coordinate is represented by an MLP rather than omitted. Table 7 reports the four configurations.

Emb. proj.	Basis	Train loss↓	Eval PPL↓
No	No	3.160	17.18
Yes	No	3.134	16.42
No	Yes	3.163	17.05
Yes	Yes	**3.131**	**16.13**

Table 7: Embedding-projection and sinusoidal-basis ablation.

Table 7 shows that the configuration combining the embedding projection and sinusoidal basis attains the lowest training loss and evaluation perplexity among the evaluated settings.

D Efficiency Analysis

This section derives the computational and storage complexity of the pair-conditioned implementation. The analysis separates cached geometric quantities from denoising-state-dependent tensors and avoids reliance on empirical systems measurements.

D.1 Cached Local Aggregation

Let $\boldsymbol{C} \in \mathbb{R}^{L \times F}$ contain the nonnegative basis coordinates and let $\boldsymbol{K}_W \in \mathbb{R}^{L \times L}$ denote the conceptual sparse, row-normalized truncated Gaussian operator. At one denoising state, the token-centered numerator and reference matrices

are

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{K}_W \operatorname{diag}(\boldsymbol{b})\boldsymbol{C}, \\ A_{i,f} &= \sum_{|t-i|\le W} K_{i,t} b_t c_{t,f}, \\ D_{i,f} &= \sum_{|t-i|\le W} K_{i,t} a_t c_{t,f} + \varepsilon. \end{aligned} \tag{28}$$

The support of $\boldsymbol{K}_W$ gives $\operatorname{nnz}(\boldsymbol{K}_W) \le L(2W+1)$ and hence

$$\begin{aligned} T_{\text{local}} &= O(LWF), \\ T_{\text{embed}} &= O(LFd), \\ T_{\text{pair}} &= O(L^2FH), \\ T_{\text{add}} &= T_{\text{local}} + T_{\text{embed}} + T_{\text{pair}}. \end{aligned} \tag{29}$$

The first term is the local reduction, the second is the enabled embedding projection, and the third evaluates $\boldsymbol{r}_{ij}$ and the shared cross-frequency phase MLP for all attention pairs. The Gaussian kernel, basis, and frequency assignment depend only on (L, W, ρ, θ) and are cached. Padding-aware $\boldsymbol{A}$ and $\boldsymbol{D}$ vary with the denoising state and are computed once per forward pass. A grouped one-dimensional convolution or banded reduction evaluates the local operator, while query-block evaluation of $\boldsymbol{r}_{ij}$ avoids materializing the full $L \times L \times F$ pair tensor.

D.2 Runtime Decomposition

For N_ℓ unchanged transformer layers, write the per-step costs as

$$\begin{aligned} T_{\text{RoPE}} &= N_\ell T_{\text{block}} + T_{\text{pos}}, \\ T_{\text{MDLMPE}} &= N_\ell T_{\text{block}} + T_{\text{pos}} + T_{\text{local}} + T_{\text{embed}} + T_{\text{pair}}. \end{aligned} \tag{30}$$

MDLMPE does not change layer count, hidden size, attention-mask shape, denoising steps, or backbone query/key/value dimensions. Because the residual phase is pair conditioned, however, the attention kernel requires an MDLMPE-aware extension; the reference implementation evaluates it in query blocks. The relative per-step overhead is

$$\frac{T_{\text{MDLMPE}} - T_{\text{RoPE}}}{T_{\text{RoPE}}} = \frac{T_{\text{local}} + T_{\text{embed}} + T_{\text{pair}}}{N_\ell T_{\text{block}} + T_{\text{pos}}}. \tag{31}$$

This expression identifies the additional per-step computation attributable to MDLMPE.

D.3 Activation and Buffer Bounds

The query, key, value, output, and KV-cache backbone shapes are identical to RoPE, while pair-conditioned phase evaluation adds block-local state inside attention. With query-block size C, the additional storage obeys

$$\begin{aligned} M_{\text{persistent}} &= O(W + LF + Ld), \\ M_{\text{pair-block}} &= O(CLF), \end{aligned} \tag{32}$$

where the $O(Ld)$ projected feature can be streamed or reused after embedding injection. The binary vector $\boldsymbol{b}$ is input derived, and the fixed kernel and basis need no gradient storage. Thus

$$M_{\text{MDLMPE}} = M_{\text{RoPE}} + O(W + LF + Ld + CLF). \tag{33}$$

An unblocked implementation has $C = L$ and therefore requires $O(L^2F)$ pair working storage. Query blocking controls this term but does not remove the pair-phase computation. Any systems evaluation should specify L, W, C, dtype, selected-frequency count, attention kernel, batch size, and denoising steps.

E Qualitative Output Comparisons

Tables 8–10 present nine paired cases from ARC, MMLU, and GSM8K. Across these examples, the MDLMPE responses provide complete and internally consistent solutions for the selected instances. The comparisons contrast non-terminating or self-contradictory baseline outputs in ARC, invalid algebraic premises in MMLU, and omitted quantities or arithmetic errors in GSM8K with responses whose stated reasoning supports the final answer. These examples offer qualitative evidence concerning response completion and reasoning consistency; they do not constitute an aggregate benchmark evaluation.

F Deterministic Mechanism Analysis

The figures in this section are direct evaluations of Eqs. 16, 19, 22, and 24. They use analytical extrema or explicitly fixed positions; no recovery schedule, random token placement, checkpoint output, attention map, or benchmark result enters any panel. Figures 7 and 8 use the default $W/L = 3/16$ and $\sigma/W = 1/4$ at length 1024. The length-256 Jacobian and boundary panels use $W = 64$ and $\sigma = 16$ only to make the full matrices legible; their locality and normalization claims do not depend on choosing the default radius.

F.1 Exact Availability Envelope

For a fixed target and frequency, define the normalized contribution of source position t as

$$\alpha_{i,t,f} = \frac{w_{i,t} c_{t,f}}{\sum_u w_{i,u} c_{u,f}}, \qquad r_{i,f} = \sum_t \alpha_{i,t,f} b_t, \tag{34}$$

where $\alpha_{i,t,f} \ge 0$ and $\sum_t \alpha_{i,t,f} = 1$ in the idealized $\varepsilon \to 0$ limit. Let N_i be the number of valid positions in the local support and let $\eta = k/N_i$ denote the revealed fraction within that support. If exactly k positions are revealed, the largest attainable ratio is obtained by selecting the k largest $\alpha_{i,t,f}$ values, and the smallest by selecting the k smallest. These sorted partial sums give exact extrema over every binary mask with that cardinality; no enumeration or stochastic recovery path is required.

Figure 7 separates density from topology. Fixing η does not fix $r_{i,f}$: the vertical width of the envelope is precisely the range created by changing revealed-token placement while preserving the number of revealed positions. The envelope collapses to zero at $\eta = 0$ and to one at $\eta = 1$, while its nonzero interior width proves that intermediate availability ratios retain spatial information beyond global density.

F.2 Frequency–Distance Transfer

Starting from an all-masked state and revealing only source t changes the ratio at target i by $\alpha_{i,t,f}$. Evaluating this quantity for every relative offset and every rotary pair yields Figure 8. The Gaussian factor confines influence to $|t-i| \le W$,

Question and reference	RoPE response	MDLMPE response
Example 1, ARC-148. How many valence electrons does selenium have? (A) 3; (B) 5; (C) 6; (D) 8. **Ref.: C.**	*Selenium is a nonmetal with atomic number 34 and is in group 66. 66. 66. 66. . . .* The response enters an unbounded numeric repetition, does not recover the intended group number, and never emits the requested final-answer sentence.	*Selenium has electron configuration* $[\mathrm{Ar}]\,3d^{10}4s^24p^4$*. The outer shell contains two* $4s$ *electrons and four* $4p$ *electrons, for six valence electrons. Equivalently, selenium is a group-16 element. The best answer is C.*
Example 2, ARC-511. Which object is orbited by a belt of asteroids? (A) Pluto; (B) Saturn; (C) the Sun; (D) the Moon. **Ref.: C.**	*The best answer is B, Saturn. The asteroid belt is between Mars and Jupiter and is not directly orbiting Saturn, so the best answer is B.* The continuation repeats the prompt and this internally inconsistent answer several times: its explanation rules out Saturn while its final letter remains B.	*The main asteroid belt occupies a region between the orbits of Mars and Jupiter, but its members follow heliocentric orbits. It is therefore a belt of objects orbiting the Sun, not a structure orbiting Saturn, Pluto, or the Moon. The best answer is C.*
Example 3, ARC-716. Which statement describes a physical characteristic of the Moon? (A) made of hot gases; (B) covered with many craters; (C) many bodies of liquid water; (D) gives off its own light. **Ref.: B.**	*The Moon is a rocky satellite with many impact craters. It lacks a substantial atmosphere and liquid surface water, so weathering and erosion do not rapidly erase those craters. It is neither made of hot gas nor a source of its own visible light. Therefore, the correct answer is B.*	*The Moon's surface preserves numerous impact craters formed over billions of years. Its extremely thin exosphere, absence of flowing surface water, and limited geological resurfacing allow these structures to remain visible. The Moon reflects sunlight rather than producing its own light. The best answer is B.*

Table 8: ARC examples 1–3 with their questions, references, and paired responses.

Question and reference	RoPE response	MDLMPE response
Example 1, MMLU-46. S1: A homomorphism may have an empty kernel. S2: No nontrivial homomorphism can map some finite group into some infinite group. (A) T,T; (B) F,F; (C) T,F; (D) F,T. **Ref.: B.**	*Statement 1 is true because an injective homomorphism can have an empty kernel. Statement 2 is false because a finite group can have a nontrivial homomorphism into an infinite group. The best answer is C.* The first step confuses a trivial kernel with an empty kernel.	*For every homomorphism* $\phi : G \to H$*, the identity satisfies* $\phi(e_G) = e_H$*, hence* $e_G \in \ker\phi$ *and the kernel cannot be empty; S1 is false. S2 is also false because an infinite group may contain a finite subgroup, allowing a nontrivial inclusion, for example* $C_2 \hookrightarrow C_2 \times \mathbb{Z}$*. The best answer is B.*
Example 2, MMLU-61. S1: Integer 2×2 matrices with nonzero determinant form a group. S2: Integer 2×2 matrices with determinant one form a group. (A) T,T; (B) F,F; (C) T,F; (D) F,T. **Ref.: D.**	*Both sets satisfy closure, associativity, identity, and inverse axioms, because every matrix with nonzero determinant is invertible. Therefore both statements are true and the best answer is A.* This overlooks that invertibility over $\mathbb{Q}$ does not guarantee an inverse with integer entries.	*S1 is false:* $\mathrm{diag}(2, 1)$ *has nonzero determinant but inverse* $\mathrm{diag}(1/2, 1)$*, which is not an integer matrix. S2 is true because a determinant-one integer matrix has inverse equal to its integer adjugate. Thus the determinant-one matrices form* $\mathrm{SL}(2, \mathbb{Z})$*. The best answer is D.*
Example 3, MMLU-91. Find all cosets of $4\mathbb{Z}$ in $2\mathbb{Z}$: (A) $4\mathbb{Z}$; (B) $4\mathbb{Z}$, $2 + 4\mathbb{Z}$; (C) $2\mathbb{Z}$; (D) $\mathbb{Z}$. **Ref.: B.**	*Elements of* $2\mathbb{Z}$ *not in* $4\mathbb{Z}$ *are represented by 2 or* -2*. Adding either representative to* $4\mathbb{Z}$ *gives the same second coset. Thus the cosets are* $4\mathbb{Z}$ *and* $2 + 4\mathbb{Z}$*, so the best answer is B.*	*Every even integer is congruent either to 0 or to 2 modulo 4. The first class is* $4\mathbb{Z}$*; the second is* $2 + 4\mathbb{Z}$*, and* $-2 + 4\mathbb{Z}$ *is identical to that second class. Hence* $[2\mathbb{Z} : 4\mathbb{Z}] = 2$ *and there are exactly two cosets. The best answer is B.*

Table 9: MMLU examples 1–3 with the decisive algebraic condition in each response.

whereas the aligned coordinate $c_{t,f}$ redistributes that local mass across frequencies. The result is a complete deterministic transfer surface rather than a selection of illustrative token positions.

The surface clarifies the division of labor between the kernel and the spectral basis. Distance controls whether a source can contribute and its broad attenuation, while frequency controls how the same source is represented across channels. Thus the fixed construction supplies multiple spatial scales without assigning a learned semantic score to any position.

Question and reference	RoPE response	MDLMPE response
Example 1, GSM8K-798. When Billy was first hired, he earned \$10 per hour. After two months he received a \$0.50 raise, and on his first anniversary he received a \$1.00 raise. Sally's starting wage is \$0.50 above Billy's original starting wage. If both work 20 hours, how much more does Billy earn? **Ref.: \$20.**	*To solve this problem, calculate Billy's and Sally's hourly wages and then determine their total earnings for 20 hours. Let's break it down step by step. First, calculate Billy's* ... The output then restarts the complete problem, begins the same first step again, restarts a second time, and truncates. It never combines the two raises, computes either 20-hour total, or emits the required final-answer line.	*Billy's current hourly wage is* $\$10.00 + \$0.50 + \$1.00 = \$11.50$. *Sally earns* $\$10.00 + \$0.50 = \$10.50$ *per hour because her comparison is with Billy's original wage. For 20 hours, Billy earns* $20 \times \$11.50 = \$230$ *and Sally earns* $20 \times \$10.50 = \$210$. *The difference is* $\$230 - \$210 = \$20$. *The final answer is \$20.*
Example 2, GSM8K-997. Walter collects \$500 from his neighbors and \$1,500 from an online fund. His lawyer offers to donate three times as much as everyone else donates. Before Walter tells the lawyer the total, a friend gives another \$200. How much will the lawyer contribute? **Ref.: \$6,600.**	*Walter first collects* $\$500 + \$1{,}500 = \$2{,}000$. *His lawyer offers three times everyone else's donation, so* $3 \times \$2{,}000 = \$6{,}000$. *Therefore Walter's lawyer is going to contribute \$6000. The final answer is \$6000.* This calculation is internally consistent for the first two sources but drops the friend's \$200 even though it arrives before Walter reports the total.	*The multiplier applies to the donations made by everyone other than the lawyer. Those sources contribute* \$500 *from neighbors,* \$1,500 *online, and* \$200 *from the friend, for* $\$500 + \$1{,}500 + \$200 = \$2{,}200$. *Three times that complete total is* $3 \times \$2{,}200 = \$6{,}600$. *The final answer is \$6600.*
Example 3, GSM8K-1014. Jamaar goes to the store with \$10 earned from mowing lawns. Including tax, peaches and pears cost \$0.50 each, apples cost \$0.75, kiwis cost \$1, and plums cost \$0.25. After buying 3 peaches, 4 pears, 2 kiwis, and 5 apples, how many plums can he buy? **Ref.: 3.**	*The cost of peaches is listed as \$1, pears as \$3, kiwis as \$1, and apples as \$3. These amounts total \$8, leaving* $\$10 - \$8 = \$2$. *At \$0.25 per plum,* $\$2/\$0.25 = 8$, *so Jamaar can buy 8 plums. The final answer is 8.* The subtraction and division are valid given the subtotal, but four preceding multiplication terms are copied or computed incorrectly.	*Compute each category independently:* $3 \times \$0.50 = \$1.50$ *for peaches,* $4 \times \$0.50 = \$2.00$ *for pears,* $2 \times \$1.00 = \$2.00$ *for kiwis, and* $5 \times \$0.75 = \$3.75$ *for apples. The subtotal is* $\$1.50 + \$2.00 + \$2.00 + \$3.75 = \$9.25$, *leaving* $\$10 - \$9.25 = \$0.75$. *Therefore* $\$0.75/\$0.25 = 3$ *plums. The final answer is 3.*

Table 10: GSM8K examples 1–3 with their questions, references, and paired responses.

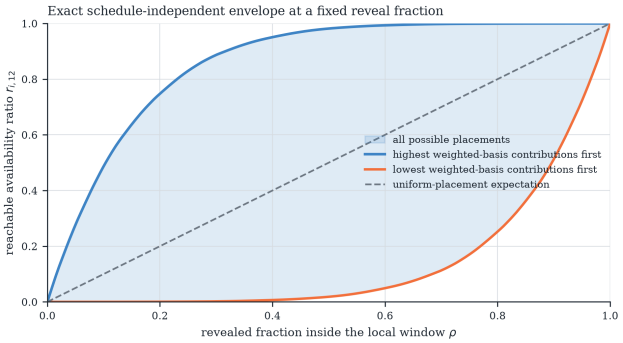

Figure 7: Exact schedule-independent availability envelope at rotary pair $f = 12$, with fixed radius $W = 192$ and $\sigma = 48$. At each local revealed fraction η, the shaded band contains every attainable $r_{i,12}$. Its upper and lower boundaries reveal the positions with the largest and smallest weighted-basis contributions first; the diagonal is the uniform-placement expectation, not a sampled trajectory.

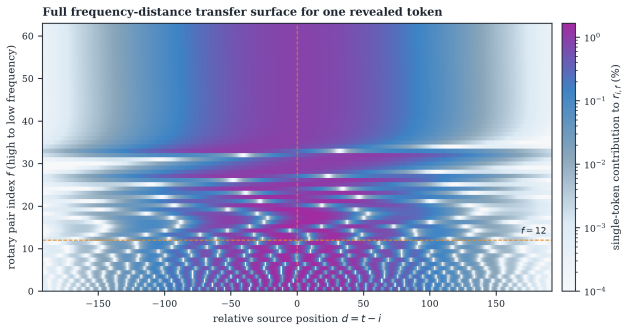

Figure 8: Frequency–distance transfer for one revealed source token in a fixed length-1024 sequence, with center target, $W = 192$, and $\sigma = 48$. Each cell is $100\alpha_{i,t,f}$, the percentage contribution to $r_{i,f}$. The vertical dashed line marks the target and the horizontal dashed line marks $f = 12$. Gaussian locality creates the bounded horizontal support; the RoPE-aligned basis produces the frequency-dependent internal structure.

F.3 Local Influence and Its Analytical Bound

Equation 19 gives the exact Jacobian of the unprojected multiscale feature with respect to one binary state. Because each component of $\boldsymbol{c}_t$ lies in $[0, 1]$, $\|\boldsymbol{c}_t\|_2 \le \sqrt{F}$ and therefore

$$\left\| \frac{\partial \boldsymbol{u}_i}{\partial b_t} \right\|_2 = w_{i,t} \|\boldsymbol{c}_t\|_2 \le w_{i,t} \sqrt{F}. \tag{35}$$

Figure 9 evaluates both sides for all target–source pairs under one fixed configuration.

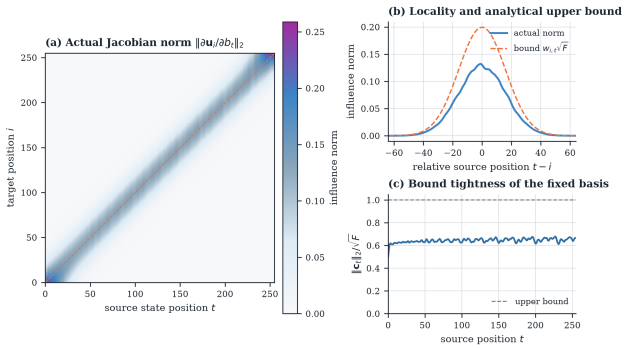

Figure 9: Exact single-state influence and analytical localization bound for length 256, $F = 64$, $W = 64$, and $\sigma = 16$. (a) The full Jacobian-norm matrix is banded around $t = i$. (b) A center-target cross-section compares the exact norm with $w_{i,t}\sqrt{F}$. (c) The basis factor $\|\boldsymbol{c}_t\|_2/\sqrt{F}$ never exceeds one, verifying the bound independently of a token-recovery pattern.

The banded matrix establishes strict locality: outside the truncated Gaussian support, changing b_t has zero direct effect on $\boldsymbol{u}_i$. Inside the support, the dashed bound isolates the worst-case magnitude contributed by the fixed basis. The plot therefore makes both the receptive field and the maximum per-token perturbation auditable before learning $\boldsymbol{W}_p$.

F.4 Intermediate Angular Coordinate and Endpoint Sensitivity

For scalar analysis, let $q(r) = \arccos(2r - 1)$ denote the uncentered angularization and $s(r) = q(r) - \pi/2$ the centered coordinate used in Eq. 24. The first map is monotone from $[0, 1]$ to $[0, \pi]$, while centering shifts its range to $[-\pi/2, \pi/2]$ without changing its derivative:

$$\frac{\partial q}{\partial r} = \frac{\partial s}{\partial r} = -\frac{1}{\sqrt{r(1-r)}}. \tag{36}$$

The magnitude is minimized at $r = 1/2$, where it equals two, and diverges as r approaches either endpoint. Figure 10 shows the deterministic angularization and its complete sensitivity curve, rather than selected availability states.

The two panels distinguish output range from numerical sensitivity. Bounded s prevents unbounded inputs to the phase MLP, while the subsequent $\tau_{\max} \tanh(\cdot)$ independently bounds the actual pair-conditioned rotary residual. Bounded values alone do not bound the derivative of

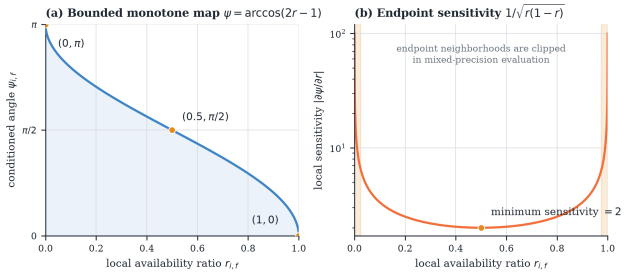

Figure 10: Deterministic ratio-to-intermediate-coordinate conversion. (a) The uncentered map $q = \arccos(2r - 1)$ maps $r = 0, 1/2, 1$ to $\pi, \pi/2, 0$; the implemented pair coordinate is $s = q - \pi/2$, which is passed to the shared MLP and is not substituted directly for RoPE. (b) The exact local sensitivity $|\partial q/\partial r| = |\partial s/\partial r| = 1/\sqrt{r(1-r)}$ is smallest at $r = 1/2$ and grows near both endpoints; the shaded endpoint neighborhoods indicate where mixed-precision clipping is applied.

the intermediate map. Equivalently, clipping the ratio to $\widetilde{r} = \mathrm{clip}(r, \delta, 1 - \delta)$ gives the explicit finite bound

$$\left| \frac{\partial s}{\partial \widetilde{r}} \right| \le \frac{1}{\sqrt{\delta(1-\delta)}}. \tag{37}$$

Thus endpoint clipping controls mixed-precision sensitivity while leaving the interior map unchanged. It is a numerical safeguard near all-masked and all-revealed states, not the final phase rule or a mechanism for selecting a recovery schedule.

F.5 Boundary Normalization and Endpoint Invariants

Sequence boundaries truncate a symmetric Gaussian window, so the unnormalized row mass is smaller near the first and last positions. Row normalization in Eq. 22 removes this purely geometric loss of support. Figure 11 compares the raw mass, normalized kernels, and the two uniform endpoint states.

This invariance prevents a boundary position from appearing less available merely because half of its nominal neighborhood lies outside the sequence. Boundary effects can still change the distribution of relative offsets in panel (b), but they do not alter the meaning of the two availability endpoints. Together, Figures 7–11 establish range, topology sensitivity, multiscale locality, bounded influence, angular sensitivity, and edge behavior. They characterize the representation; task-quality claims continue to rely on the matched experiments in the main paper and Appendices B–D.

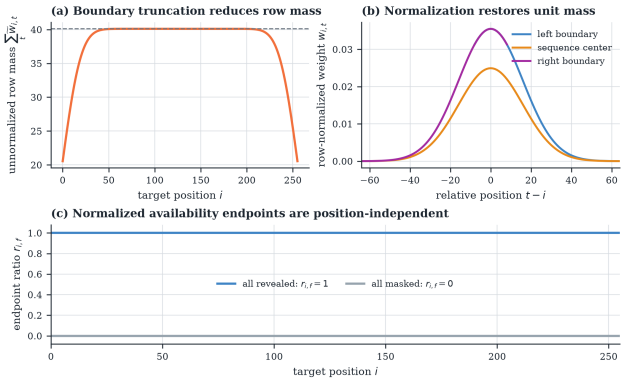

Figure 11: Boundary normalization for length 256, $W = 64$, and $\sigma = 16$. (a) Truncation reduces unnormalized Gaussian row mass near sequence ends. (b) Renormalized kernels at the left boundary, center, and right boundary each have unit mass despite asymmetric support. (c) Consequently, the ideal endpoint ratios are position independent: $r_{i,f} = 0$ for an all-masked state and $r_{i,f} = 1$ for an all-revealed state (up to the denominator safeguard ε).

References

Allal, L. B.; Lozhkov, A.; Bakouch, E.; Blázquez, G. M.; Penedo, G.; Tunstall, L.; Marafioti, A.; Kydlíček, H.; Lajarín, A. P.; Srivastav, V.; et al. 2025. SmolLM2: When Smol Goes Big – Data-Centric Training of a Small Language Model. arXiv:2502.02737.

Arriola, M.; Gokaslan, A.; Chiu, J. T.; Yang, Z.; Qi, Z.; Han, J.; Sahoo, S. S.; and Kuleshov, V. 2025. Block Diffusion: Interpolating Between Autoregressive and Diffusion Language Models. In *International Conference on Learning Representations*.

Austin, J.; Johnson, D. D.; Ho, J.; Tarlow, D.; and van den Berg, R. 2021a. Structured Denoising Diffusion Models in Discrete State-Spaces. In *Advances in Neural Information Processing Systems*, volume 34, 17981–17993.

Austin, J.; Odena, A.; Nye, M.; Bosma, M.; Michalewski, H.; Dohan, D.; Jiang, E.; Cai, C.; Terry, M.; Le, Q.; and Sutton, C. 2021b. Program Synthesis with Large Language Models. *arXiv preprint arXiv:2108.07732*.

Bie, T.; Cao, M.; Chen, K.; Du, L.; Gong, M.; et al. 2025. LLaDA2.0: Scaling Up Diffusion Language Models to 100B. *arXiv preprint arXiv:2512.15745*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33: 1877–1901.

Chang, H.; Zhang, H.; Jiang, L.; Yang, C.; and Liu, Y. 2022. MaskGIT: Masked Generative Image Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11315–11325.

Chelba, C.; Mikolov, T.; Schuster, M.; Ge, Q.; Brants, T.; Koehn, P.; and Robinson, T. 2014. One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling. *arXiv preprint arXiv:1312.3005*.

Chen, J.; Liang, Y.; and Liu, Z. 2026. DFlash: Block Diffusion for Flash Speculative Decoding. arXiv:2602.06036.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H. P. d. O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374*.

Chen, S.; Wong, S.; Chen, L.; and Tian, Y. 2023. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv preprint arXiv:1803.05457*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training Verifiers to Solve Math Word Problems. In *arXiv preprint arXiv:2110.14168*.

Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q. V.; and Salakhutdinov, R. 2019. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2978–2988.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

Ding, Y.; Zhang, L. L.; Zhang, C.; Xu, Y.; Shang, N.; Xu, J.; Yang, F.; and Yang, M. 2024. LongRoPE: Extending LLM Context Window Beyond 2 Million Tokens. *arXiv preprint arXiv:2402.13753*.

Ghazvininejad, M.; Levy, O.; Liu, Y.; and Zettlemoyer, L. 2019. Mask-Predict: Parallel Decoding of Conditional Masked Language Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 6112–6121.

Gokaslan, A.; Cohen, V.; Pavlick, E.; and Tellex, S. 2019. OpenWebText Corpus. http://Skylion007.github.io/OpenWebTextCorpus.

Gong, S.; Meng, C.; Feng, J.; Chen, Y.; and He, Y. 2023. DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models. In *International Conference on Learning Representations*.

Gu, J.; Wang, C.; and Zhao, J. 2019. Levenshtein Transformer. In *Advances in Neural Information Processing Systems*, volume 32.

He, G.; Nie, S.; Zhu, F.; Zhao, Y.; Bai, T.; Yan, R.; Fu, J.; Li, C.; and Yuan, B. 2025. UltraLLaDA: Scaling the Context Length to 128K for Diffusion Large Language Models. *arXiv preprint arXiv:2510.10481*.

He, P.; Liu, X.; Gao, J.; and Chen, W. 2021. DeBERTa: Decoding-Enhanced BERT with Disentangled Attention. In *International Conference on Learning Representations*.

He, Z.; Sun, T.; Tang, Q.; Wang, K.; Huang, X.; and Qiu, X. 2023. DiffusionBERT: Improving Generative Masked Language Models with Diffusion Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021a. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021b. Measuring Mathematical Problem Solving With the MATH Dataset. In *Advances in Neural Information Processing Systems*, volume 34, 11859–11870.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33, 6840–6851.

Hoogeboom, E.; Nielsen, D.; Jaini, P.; Forré, P.; and Welling, M. 2021. Argmax Flows and Multinomial Diffusion: Learning Categorical Distributions. In *Advances in Neural Information Processing Systems*, volume 34, 12454–12465.

Lambert, N.; Morrison, J.; Pyatkin, V.; Huang, S.; Ivison, H.; Brahman, F.; Miranda, L. J. V.; Liu, A.; Dziri, N.; Lyu, S.; et al. 2024. TÜLU 3: Pushing Frontiers in Open Language Model Post-Training. *arXiv preprint arXiv:2411.15124*.

Li, X. L.; Thickstun, J.; Gulrajani, I.; Liang, P.; and Hashimoto, T. B. 2022. Diffusion-LM Improves Controllable Text Generation. In *Advances in Neural Information Processing Systems*, volume 35, 4328–4343.

Liu, X.; Song, Y.; Liu, Z.; Huang, Z.; Guo, Q.; He, Z.; and Qiu, X. 2025. LongLLaDA: Unlocking Long Context Capabilities in Diffusion LLMs. *arXiv preprint arXiv:2506.14429*.

Lou, A.; Meng, C.; and Ermon, S. 2024. Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution. In *International Conference on Machine Learning*.

Mahabadi, R. K.; Ivison, H.; Tae, J.; Henderson, J.; Beltagy, I.; Peters, M. E.; and Cohan, A. 2024. TESS: Text-to-Text Self-Conditioned Simplex Diffusion. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, 2347–2361.

Mahoney, M. 2011. Text8 Dataset. http://mattmahoney.net/dc/textdata.

Marcus, M. P.; Santorini, B.; and Marcinkiewicz, M. A. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2): 313–330.

Merity, S.; Xiong, C.; Bradbury, J.; and Socher, R. 2017. Pointer Sentinel Mixture Models. *arXiv preprint arXiv:1609.07843*.

Nichol, A. Q.; and Dhariwal, P. 2021. Improved Denoising Diffusion Probabilistic Models. In *International Conference on Machine Learning*, 8162–8171.

Nie, S.; Zhu, F.; You, Z.; Zhang, X.; Ou, J.; Hu, J.; Zhou, J.; Lin, Y.; Wen, J.-R.; and Li, C. 2025. Large Language Diffusion Models. *Advances in Neural Information Processing Systems*.

Ou, J.; Nie, S.; Xue, K.; Zhu, F.; Sun, J.; Li, Z.; and Li, C. 2025. Your Absorbing Discrete Diffusion Secretly Models the Conditional Distributions of Clean Data. In *International Conference on Learning Representations*.

Peng, B.; Quesnelle, J.; Fan, H.; and Shippole, E. 2024. YaRN: Efficient Context Window Extension of Large Language Models. In *International Conference on Learning Representations*.

Press, O.; Smith, N. A.; and Lewis, M. 2022. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. In *International Conference on Learning Representations*.

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

Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-Attention with Relative Position Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 464–468.

Shi, J.; Han, K.; Wang, Z.; Doucet, A.; and Titsias, M. K. 2024. Simplified and Generalized Masked Diffusion for Discrete Data. In *Advances in Neural Information Processing Systems*, volume 37.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.

Song, Y.; Zhang, Z.; Luo, C.; Gao, P.; Xia, F.; Luo, H.; et al. 2025. Seed Diffusion: A Large-Scale Diffusion Language Model with High-Speed Inference. *arXiv preprint arXiv:2508.02193*.

Stern, M.; Chan, W.; Kiros, J.; and Uszkoreit, J. 2019. Insertion Transformer: Flexible Sequence Generation via Insertion Operations. In *International Conference on Machine Learning*, 5976–5985.

Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Wen, B.; and Liu, Y. 2024. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063.

Sun, Y.; Dong, L.; Patra, B.; Ma, S.; Huang, S.; Benhaim, A.; Chaudhary, V.; Song, X.; and Wei, F. 2023. A Length-Extrapolatable Transformer. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Wang, Y.; Ma, X.; Zhang, G.; Ni, Y.; Chandra, A.; Guo, S.; Ren, W.; Arulraj, A.; He, X.; Jiang, Z.; et al. 2024. MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark. *arXiv preprint arXiv:2406.01574*.

Ye, J.; Xie, Z.; Zheng, L.; Gao, J.; Wu, Z.; Jiang, X.; Li, Z.; and Kong, L. 2025. Dream 7B: Diffusion Large Language Models. *arXiv preprint arXiv:2508.15487*.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4791–4800.

Zhu, D.; Yang, N.; Wang, L.; Song, Y.; Wu, W.; Wei, F.; and Li, S. 2024. PoSE: Efficient Context Window Extension of LLMs via Positional Skip-wise Training. In *International Conference on Learning Representations*.