MAGE: All-[MASK] Block Already Knows Where to Look in Block Diffusion LLM

Omin Kwon
Seoul National University
om0127@snu.ac.kr

Yeonjae Kim
Seoul National University
duswo1120@snu.ac.kr

Doyeon Kim
Seoul National University
pega501@snu.ac.kr

Minseo Kim
UC Berkeley
minseo.kim@berkeley.edu

Yeonhong Park
Meta
parkyh96@gmail.com

Jae W. Lee*
Seoul National University
jaewlee@snu.ac.kr

Abstract

Block diffusion LLMs are an emerging paradigm for parallel language generation, but their KV caching makes memory access the dominant bottleneck in long-context inference. Sparse attention, which attends only to a small KV subset per query, can reduce this latency with minimal accuracy loss. In block diffusion, however, the B tokens of each block must share a single KV subset, and we show this per-block constraint degrades existing sparse KV estimators by up to 25% in recall. We address this challenge by exploiting a property that emerges from the block-diffusion training objective: it aligns the block-average query across denoising steps, so the All-[MASK] block at the first step already reveals the per-block KV subset for the entire trajectory. We exploit this in MAGE ([MASK]-Guided Sparse Attention), a training-free method that runs one exact attention pass at the first step and reuses its top-k index sets for all remaining steps within the block. Across three block-diffusion families on LongBench, MAGE matches Exact Attention at k=512 with near-lossless accuracy, achieves up to $6.82\times$ end-to-end speedup at 128K context, and runs up to $3.35\times$ and $2.28\times$ faster than Quest and SparseD, designed for AR LLMs and fully bidirectional diffusion LLMs, respectively. The code is available at https://github.com/SNU-ARC/MAGE.

1 Introduction

Block diffusion LLMs (Block dLLMs) [Arriola et al., 2025, Wu et al., 2026a, Cheng et al., 2026, Bie et al., 2025] have emerged as a promising paradigm for parallel language generation: by adapting pretrained autoregressive (AR) models into block-wise diffusion models with exact KV caching, they increase the throughput by multi-token generation while maintaining competitive quality. As context length grows, attention dominates the overall inference computation and becomes bottlenecked by KV cache memory access [Pope et al., 2023], motivating sparse attention as a natural remedy. Sparse attention reduces this cost by attending only to a small selected *KV subset*—the top-k KV cache entries that capture most of the attention mass—rather than the full cache. Dynamic estimators such as Quest [Tang et al., 2024] have proven highly effective for AR LLMs, and methods such as SparseD [Wang et al., 2026c] have been proposed for fully bidirectional dLLMs—but neither was designed for block diffusion, and transferring them directly reveals a structural mismatch that substantially degrades their accuracy.

The mismatch stems from a constraint unique to block diffusion: the B tokens within a block are forwarded together, so they must share a single *per-block* KV subset rather than each selecting its

*Corresponding author.

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

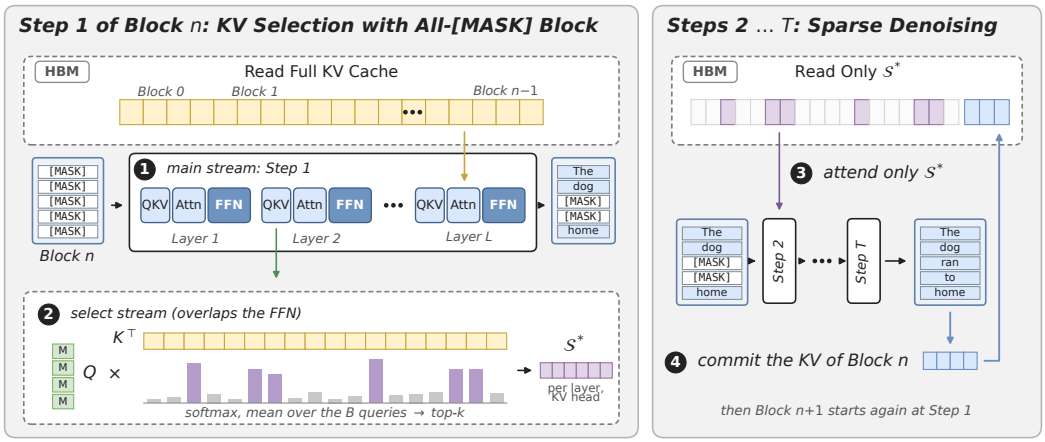

Figure 1: Overview of MAGE. ❶–❷ At Step 1 of each block, whose B positions are all `[MASK]`, the main stream runs the full forward pass while the select stream takes the queries of each layer, computes their attention over the prefix KV cache, averages it over the B queries, and keeps the top-k KV subset $\mathcal{S}^*$ per layer and KV head, overlapping the FFN. ❸ Steps 2 to T attend only to $\mathcal{S}^*$. ❹ Once the block is denoised, its KV is computed and appended to the cache, and the next block starts again at Step 1.

own per-token subset. Constructing a single shared subset that simultaneously satisfies each of the B tokens' top-k preferences is structurally harder: the union of their per-token subsets averages $4.5\times$ the budget. As a result, when existing estimators are adapted to this regime, their recall drops by up to 25%. We identify an opportunity on a *different axis*: block-diffusion training, by requiring the model to predict each masked token under all surrounding noise levels within a block, aligns the block-average query across the denoising steps. As a consequence, the All-`[MASK]` block in the first step within a block already reveals the per-block KV subset for the entire trajectory, with per-block top-k recall averaging over 84% at k=512—substantially above the AR pre-training backbone, confirming this anchoring is training-induced.

Building on this observation, we propose MAGE (`[MASK]`-Guided Sparse Attention), the first sparse attention method tailored to block-diffusion LLMs (Fig. 1). MAGE runs a single exact attention pass on the All-`[MASK]` block at the first step (Step 1), extracts the exact top-k index sets, and reuses them unchanged for all $T-1$ remaining denoising steps—requiring no re-estimation at any subsequent step. To avoid paying the selection cost on the critical path, MAGE executes the index-selection kernels asynchronously on a dedicated CUDA stream, overlapping them with the compute-intensive FFN of the main stream; the amortized non-sparse overhead is $1/T$ of one exact attention pass per step. Across three block-diffusion LLM families on LongBench, MAGE matches Exact Attention at k=512 with near-lossless accuracy and achieves end-to-end speedups of up to $6.82\times$ at 128K context, running up to $3.35\times$ faster than Quest and $2.28\times$ faster than SparseD.

Our contributions are summarized as follows:

- We identify the *per-block shared KV selection* challenge unique to block diffusion, showing that existing sparse KV estimators lose up to 25% recall under this constraint (§3.1).
- We discover the All-`[MASK]` anchoring property induced by block-diffusion training, and explain through a quantitatively verified five-link chain why Step 1 serves as an accurate anchor for the full denoising trajectory (§3.2).
- We propose MAGE, a training-free sparse attention method with an asynchronous dual-stream design that incurs only $+1.9\%$ to $+6.4\%$ latency overhead at Step 1 (§4).
- We demonstrate near-lossless accuracy at k=512 with up to $6.82\times$ end-to-end speedup at 128K context, outperforming Quest by up to $3.35\times$ and SparseD by up to $2.28\times$ (§5).

2 Related Work

2.1 Evolution of Diffusion Language Models

Diffusion LLMs generate text by iteratively denoising sequences of `[MASK]` tokens, enabling parallel generation of multiple tokens per forward pass and breaking the left-to-right dependency of autoregressive (AR) models [Nie et al., 2025, Ye et al., 2025a].

Fully Bidirectional Diffusion LLMs. Earlier diffusion LLMs [Nie et al., 2025, Ye et al., 2025a] apply bidirectional attention over the entire sequence, requiring the entire sequence to be recomputed at every denoising step. This design is incompatible with standard KV caching and leads to substantial inference latency [Kim et al., 2025], and also exhibits suboptimal task performance compared to strong AR baselines [Nie et al., 2025, Ye et al., 2025a]. While subsequent works have proposed approximate KV caching schemes to partially mitigate this inefficiency [Wu et al., 2026b, Ma et al., 2025, Lee et al., 2026], such approximations inevitably introduce errors that accumulate across denoising steps.

Block Diffusion LLMs. More recent work departs from bidirectional architectures and instead trains models with block-wise generation in mind [Arriola et al., 2025]. Some approaches convert bidirectional diffusion models into block diffusion models [Wang et al., 2026a, Kim et al., 2026], while the more prevalent approach adapts pretrained AR models into block diffusion models [Gong et al., 2025], as exemplified by Fast-dLLM v2 [Wu et al., 2026a], SDAR [Cheng et al., 2026], LLaDA 2.0 [Bie et al., 2025], TraDo [Wang et al., 2026b], DreamReasoner [Wu et al., 2026c], and Nemotron-Labs-Diffusion [Fu et al., 2026], and block diffusion has also been adopted by industry-developed models [Google DeepMind, 2025, Song et al., 2025]. These models enable *exact*, block-level KV caching without approximation, shifting the primary bottleneck of attention from computation to KV cache memory access. As a result, the efficiency of block diffusion LLM inference is increasingly dominated by KV cache memory access, particularly in long-context scenarios where the cache size grows substantially. In this work, we focus on Block Diffusion LLMs and directly target this memory bottleneck.

2.2 Sparse Attention for Long-Context Inference

Sparse attention alleviates this bottleneck by accessing only the small subset of KV entries on which the attention mass concentrates—ideally the *oracle top-k*, defined as the k prefix indices that carry the bulk of the mass in the exact attention score distribution. Obtaining the oracle, however, requires the very score matrix that exact attention over the full KV cache would produce; existing methods therefore perform sparse attention through *lightweight estimators* of the oracle. We measure how well any KV subset serves a given token via *estimation accuracy*—the fraction of that token's oracle top-k entries that are covered by the selected subset:

$$\mathrm{EstAcc}_{q,h}^{(t)} \;=\; \frac{\left|\mathcal{S}_h^{(t)} \cap \mathcal{S}_{q,h}^{*(t)}\right|}{k}, \tag{1}$$

where h indexes the KV head, $\mathcal{S}_h^{(t)}$ is the KV subset being evaluated, and $\mathcal{S}_{q,h}^{*(t)}$ is the oracle KV subset for token q. No such estimator has yet been designed for block dLLMs; below, we review existing methods developed for AR LLMs and fully bidirectional dLLMs, on which our analysis later builds.

Methods for AR LLMs. In AR LLMs, the KV cache grows token by token across decoding steps, and sparse attention has been studied most extensively in this setting. Early AR sparse attention methods rely on fixed sparsity patterns—such as sliding windows or attention sinks where the attention mass concentrates [Xiao et al., 2024, Beltagy et al., 2020, Zaheer et al., 2020]—or evict less critical KV entries based on accumulated attention statistics [Zhang et al., 2023, Liu et al., 2023, Li et al., 2024, Feng et al., 2025]. Neither fixed patterns nor one-shot eviction, however, can accurately track the oracle's preferred KV entries, which vary with every newly generated token at each decoding iteration. This motivates *dynamic* estimators that select a per-token KV subset on the fly [Tang et al., 2024, Ribar et al., 2024, Singhania et al., 2024, Yang et al., 2025, 2026].

Quest Tang et al. [2024] is a representative example: it partitions the prefix KV cache into contiguous *pages* of size p, summarizes each page by its per-channel max/min keys, scores each page against the

query via a sign-aware upper bound on $q^\top k$ ($\sum_d \max(q_d K_d^{\max}, q_d K_d^{\min})$), and uses the KV caches of the top-scoring pages as the sparse subset. While page-summary estimation incurs some accuracy loss relative to the oracle, the ability to dynamically fetch a per-token KV subset substantially improves accuracy over fixed patterns. Smaller p yields finer-grained selection that tracks the oracle more closely, but the page-scoring step itself becomes more expensive—a granularity-cost trade-off. Subsequent works refine this recipe in two directions: SparQ [Ribar et al., 2024] and Loki [Singhania et al., 2024] reduce scoring cost via channel selection or low-rank projection, while PyramidKV [Cai et al., 2025], Tactic [Zhu et al., 2026], and Twilight [Lin et al., 2025] adaptively tune the per-layer/per-head selection budget.

Methods for Fully Bidirectional Diffusion LLMs. In fully bidirectional dLLMs, the entire sequence's keys and values are recomputed at every denoising step without KV cache. SparseD [Wang et al., 2026c] performs exact attention during the first 20% of the total denoising steps across all blocks, extracts each token's oracle top-k KV subset at the final exact-attention step, and reuses these per-token attention masks across the remaining 80% of the steps. MaskKV [Huang et al., 2026] instead drops less critical prompt tokens after prefill, reducing the sequence length processed in subsequent denoising steps. In both methods, the sparse pattern is captured once per inference and reused throughout without further refresh as denoising progresses, which can potentially degrade recall as the trajectory unfolds. Sparse-dLLM [Song et al., 2026] instead evicts low-saliency KV entries with a delay, but an evicted entry can no longer be attended to.

3 Challenges and Opportunities of Sparse Attention in Block Diffusion LLMs

In this section, we identify the structural constraint that sparse attention on block diffusion LLMs must operate under, and a block-diffusion-specific property that recovers estimation accuracy within this constraint along a different axis.

3.1 The Challenge: Sparse KV Subset Estimation Is Harder in Block Diffusion LLMs

Adapting sparse attention to block diffusion LLMs is structurally harder than it appears: preserving inference efficiency requires all block tokens to share a single KV subset via per-block shared memory access, and existing lightweight estimators suffer an estimation accuracy drop under this constraint. We now explain why this conflict arises and quantify its impact.

Per-Block KV Subset Forced by Shared Memory Access. Within a block, the B tokens pass through the model in a single model forward pass: their queries share the same prefix KV cache reads which include input prompts and the previously generated block, and the attention is computed as a single GEMM. Therefore, unlike AR LLMs and fully bidirectional dLLMs—where each token selects its own *per-token* sparse KV subset—a block dLLM requires the entire block to share a single *per-block* sparse KV subset. Letting each token use its own subset is possible but makes the KV accesses irregular, which slows sparse attention by 2.5–7.9× (Appendix C). More precisely, the per-token KV subset is already shared across the G query heads of a KV group under GQA [Ainslie et al., 2023], and the block structure of block dLLMs introduces a further sharing across the B token positions of the block, so the per-block top-k averages scores over both the G heads and the B tokens:

$$\underbrace{\mathcal{S}_{q,h}^{(t)} \;=\; \underset{j}{\text{Top-}k}\Big(\tfrac{1}{G}\sum_{g\in\text{group}(h)} s_{g,q,j}^{(t)}\Big)}_{\text{per-token}} \qquad \underbrace{\mathcal{S}_{b,h}^{(t)} \;=\; \underset{j}{\text{Top-}k}\Big(\tfrac{1}{GB}\sum_{g\in\text{group}(h)}\sum_{q\in b} s_{g,q,j}^{(t)}\Big)}_{\text{per-block}} \tag{2}$$

where $j \in [N]$ indexes a prefix KV position (N is the prefix length), and $s_{g,q,j}^{(t)}$ is determined by the estimation method: when s is the exact attention score matrix, the resulting set is the *oracle per-block KV subset* $\mathcal{S}_{b,h}^{*(t)}$; when a lightweight estimator like Quest substitutes a cheap approximation for s, the resulting set is the *estimated per-block KV subset* $\hat{\mathcal{S}}_{b,h}^{(t)}$.

Estimation Accuracy Drop Under Per-Block Shared Selection. A single per-block KV subset shared by B tokens cannot satisfy each token's individual oracle simultaneously. To show this empirically, we form the union of the B per-token oracle top-k subsets and measure its size as a multiple of k. Fig. 2(a) reports this ratio at k=512 across three block-diffusion models on LongBench: the union averages $4.5\times$ the per-token budget, with a heavy tail beyond $10\times$. Fig. 2(b) quantifies the

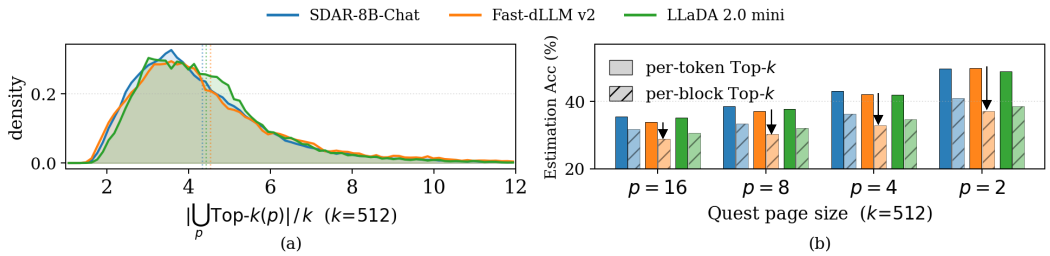

Figure 2: Sparse attention challenge on block dLLMs on LongBench, across three block-diffusion families. *(a)* Distribution of $|\bigcup_p \text{Top-}k(p)|/k$ at k=512, t=1. *(b)* Quest [Tang et al., 2024] estimation accuracy at k=512, t=1, page sizes $p \in \{16, 8, 4, 2\}$.

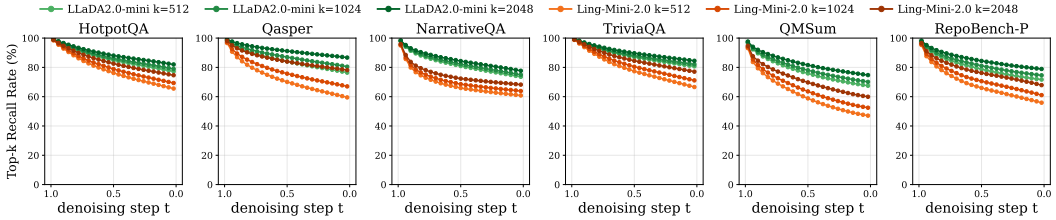

Figure 3: Per-block top-k recall along the denoising trajectory on LongBench. *Trained*: LLaDA2.0-mini; *Backbone*: its pre-training checkpoint Ling-Mini-base-2.0. Recall is substantially higher after block-diffusion training across all tasks and k.

resulting estimation accuracy drop using Quest [Tang et al., 2024]: across all three models and page sizes $p \in \{16, 8, 4, 2\}$, the per-block estimation accuracy falls below the per-token baseline by up to 25%.

3.2 The Opportunity: Accurate Estimation via All-`[MASK]` Block Anchoring

While a single k-sized subset cannot satisfy every token simultaneously, the fact that $4.5\times$ the per-token budget suffices to cover all B=32 tokens indicates that a substantial intersection of KV positions required across many tokens clearly exists. Accurately identifying this intersection within the k-budget is therefore the central question for per-block sparse estimation under the constraint of §3.1. We empirically discover that this intersection, the oracle per-block KV subset, barely drifts across the denoising trajectory, a phenomenon rooted in the block-diffusion training mechanism. This stability allows us to invest one exact-attention pass at the first step to estimate the oracle accurately and reuse it across the remaining steps, amortizing the first-step cost across the trajectory and boosting estimation accuracy without sacrificing inference efficiency.

Observation: Oracle Per-Block KV Subset Barely Drifts Across Denoising Steps. We quantify this stability as the fraction of the all-`[MASK]` oracle (t=1) recovered at an arbitrary noise level $\tau \in [0, 1)$:

$$r_{b,h}^{(\tau)} = \frac{\left|\mathcal{S}_{b,h}^{*(1)} \cap \mathcal{S}_{b,h}^{*(\tau)}\right|}{k}, \tag{3}$$

LLaDA2.0-mini averages 85% / 86% / 89% recall at k=512 / 1024 / 2048, uniform across all 6 LongBench tasks (Fig. 3). To check whether this anchoring is induced by block-diffusion training, we run the same measurement on its AR pre-training checkpoint (Ling-Mini-base-2.0): only 76% / 79% / 83%, pointing to training as the source.

Mechanism: Anchoring as a Direct Consequence of the Block-Diffusion Objective. We trace this stability to the block-diffusion training procedure itself: the prefix is kept as clean tokens while the current block is corrupted at a random noise level, yielding a mask pattern $\mathbf{m}$, and the model is trained for every $\mathbf{m}$ to predict the ground-truth token at each masked position. Since the prefix is

Table 1: Empirical verification of the five-stage alignment on LongBench, comparing block-diffusion-trained models against their AR pre-training backbones: cross-noise-level KL divergence of predictions at masked positions (Stage 1), normalized L2 gap of hidden states (Stage 2), per-position and block-average query L2 gaps (Stages 3–4), and top-k recall (Stage 5). All L2 gaps use unit-normalized vectors. Percentages are relative change from backbone to trained.

		Logit (Stage 1)	Hidden state (Stage 2)	Query $\|\tilde{q}^{(\mathbf{m}^\star)} - \tilde{q}^{(\mathbf{m})}\|$ (Stages 3–4)		Top-k Recall (Stage 5)
Family	Model	$D_{\mathrm{KL}}(p^{(\mathbf{m}^\star)} \,\|\, p^{(\mathbf{m})}) \downarrow$	$\|\tilde{h}^{(\mathbf{m}^\star)} - \tilde{h}^{(\mathbf{m})}\| \downarrow$	per-position $\downarrow$	block-avg $\downarrow$	k=512 $\uparrow$
SDAR	Qwen3-8B *(backbone)*	2.40	0.57	0.38	0.28	0.808
	SDAR-8B-Chat *(trained)*	1.38 (–43%)	0.53 (–8%)	0.34 (–13%)	0.22 (–23%)	0.852 (+5%)
Fast-dLLM v2	Qwen2.5-7B *(backbone)*	5.66	0.94	0.38	0.27	0.753
	Fast-dLLM v2 7B *(trained)*	2.08 (–63%)	0.78 (–17%)	0.28 (–27%)	0.18 (–35%)	0.826 (+10%)
LLaDA2	Ling-mini-base-2.0 *(backbone)*	2.61	0.70	0.45	0.30	0.769
	LLaDA2.0-mini *(trained)*	0.99 (–62%)	0.57 (–19%)	0.36 (–20%)	0.24 (–20%)	0.851 (+11%)

identical, the only difference seen by a [MASK] token under two mask patterns $\mathbf{m}^*$ and $\mathbf{m}$ is whether each surrounding position is itself a [MASK] or a revealed token. Yet training pushes the predictions under both patterns toward the same ground-truth target, forcing the activations of [MASK] and revealed positions to align across noise levels. This pressure propagates from hidden states to queries and ultimately to the per-block top-k subset; we verify the stages (**S1**: prediction $\rightarrow$ **S2**: hidden state $\rightarrow$ **S3**– **S4**: query $\rightarrow$ **S5**: top-k) against AR pre-training backbones in Table 1.

(Stage 1) Cross-noise-level prediction agreement. The block-diffusion loss averages cross-entropy over the Bernoulli mask schedule:

$$\mathcal{L}_{\text{block}}(\theta) \;=\; \mathbb{E}_{t\sim\mathcal{U}(0,1),\,\mathbf{m}\sim\mathrm{Bern}(t)^B}\Big[\; -\sum_{j:\,m_j=1} \log p_\theta\big(x_j \,\big|\, \mathbf{m},\,\text{prefix}\big)\Big]. \tag{4}$$

At each [MASK] token j, this is $D_{\mathrm{KL}}(\delta_{x_j} \,\|\, p_\theta(\cdot|\mathbf{m}, \text{prefix}))$, minimized for every $\mathbf{m}$ independently — so predictions at different noise levels collapse onto δ_{x_j} and thus onto each other ($D_{\mathrm{KL}}(p^{(\mathbf{m}^*)}\|p^{(\mathbf{m})})$ drops 43–63%, col. 1).

(Stage 2) Hidden state alignment. With the prefix KV fixed, this pressure aligns the block's hidden states across noise levels ($\|\tilde{h}^{(\mathbf{m}^*)} - \tilde{h}^{(\mathbf{m})}\|$ drops 8–19%, col. 2).

(Stage 3) Per-position query gap. Hidden-state alignment propagates linearly to queries via $q = W_Q h$. The per-position query gap $A := \frac{1}{B}\sum_i \|\bar{\delta}_i\|$ with $\bar{\delta}_i := \mathbb{E}_{\mathbf{m}}[q_i^{(\mathbf{m}^*)} - q_i^{(\mathbf{m})}]$ shrinks 13–27% (col. 3).

(Stage 4) Block-average query gap. Block averaging absorbs token-specific variance, keeping the gap on $\bar{q}^{(t)} := \frac{1}{B}\sum_i q_i^{(t)}$ strictly below the per-position gap (col. 4 < col. 3 in every family). By Jensen's inequality,

$$\big\|\mathbb{E}_{\mathbf{m}}[\bar{q}^{(\mathbf{m}^*)} - \bar{q}^{(\mathbf{m})}]\big\| \;=\; \Big\|\tfrac{1}{B}\sum_{i=1}^{B}\bar{\delta}_i\Big\| \;\le\; \tfrac{1}{B}\sum_{i=1}^{B}\|\bar{\delta}_i\| \;=\; A.$$

(Stage 5) Per-block top-k stability. The stabilized block-average query, against the fixed prefix keys, stabilizes the per-block top-k subset (recall at k=512 rises 5–11 points, col. 5) — establishing $\mathbf{m}^*$ as a valid anchor for the entire trajectory.

Our Proposal: One Exact Attention Pass at Step 1 Closes the Estimation Gap. Concretely, we run one exact-attention pass on the all-`[MASK]` block at t=1, extract the exact per-block KV subset $\mathcal{S}_{b,h}^{*(1)}$, i.e., the oracle at the first step, and reuse it for all remaining steps (i.e., $\hat{\mathcal{S}}_{b,h}^{(t)} = \mathcal{S}_{b,h}^{*(1)}$ for all $t < 1$). Because the per-block KV subset is computed exactly rather than approximated, our method removes the estimator-side error that lightweight methods like Quest accumulate on top of the per-block constraint. As a result (Fig. 4), our estimation accuracy surpasses Quest per-block at every page size, and matches or exceeds even Quest per-token at the finest setting p=2.

4 MAGE: [MASK]-Guided Sparse Attention

We present MAGE ([MASK]-Guided Sparse Attention), the first sparse attention method tailored to block-diffusion LLMs (Fig. 1 and Algorithm 1). Building on the one-shot All-[MASK] selection of

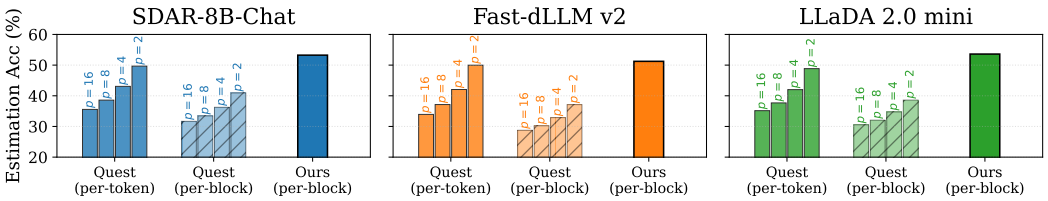

Figure 4: Estimation accuracy of different KV subset estimators at k=512 on LongBench, across three block-diffusion families.

Algorithm 1 MAGE block inference

Require: Prefix KV cache $\mathcal{C}$, block size B, budget k, denoising steps T
Ensure: Generated block $\mathbf{x}$
Initialize block: B `[MASK]` tokens
// Phase 1: exact selection at Step 1 (All-`[MASK]`); two CUDA streams run concurrently

Main Stream
for layer $\ell = 1, \ldots, L$ **do**
  $(\mathbf{Q}_\ell^{(1)}, \mathbf{K}_\ell^{(1)}, \mathbf{V}_\ell^{(1)}) \leftarrow \mathrm{QKVProj}_\ell(\mathbf{h}_{\ell-1}^{(1)})$
  enqueue $\mathbf{Q}_\ell^{(1)} \rightarrow$ SELECT ▷ non-blocking
  $\mathbf{h}_\ell^{(1)} \leftarrow \mathrm{Attn}_\ell(\mathbf{Q}_\ell^{(1)}, \mathbf{K}_\ell^{(1)}, \mathbf{V}_\ell^{(1)}, \mathcal{C})$
  $\mathbf{h}_\ell^{(1)} \leftarrow \mathrm{FFN}_\ell(\mathbf{h}_\ell^{(1)})$
end for

Select Stream
for layer $\ell = 1, \ldots, L$ **do**
  wait for $\mathbf{Q}_\ell^{(1)}$
  $\mathbf{A}_\ell^{(1)} \leftarrow \mathrm{softmax}(\mathbf{Q}_\ell^{(1)} \mathbf{K}^\top / \sqrt{d})$
  for each KV head h **do**
    $\bar{\alpha}_{\ell,h} \leftarrow \frac{1}{GB} \sum_{g,q} \mathbf{A}_\ell^{(1)}[g, q, :]$
    $\mathcal{S}_{b,h}^{*\ell} \leftarrow \text{Top-}k(\bar{\alpha}_{\ell,h})$
  end for
end for

synchronize(MAIN, SELECT) *// Phase 1 complete; all $\mathcal{S}_{b,h}^{*\ell}$ ready*
// Phase 2: sparse denoising for Step $s = 2, \ldots, T$
for $s = 2$ to T **do**
  for each layer ℓ **do**
    $(\mathbf{Q}_\ell, \mathbf{K}_\ell, \mathbf{V}_\ell) \leftarrow \mathrm{QKVProj}_\ell(\mathbf{h}_{\ell-1})$
    for each KV head h **do**
      $\mathbf{o}_{\ell,h} \leftarrow \mathrm{SparseAttn}(\mathbf{Q}_{\ell,h}, \mathbf{K}_{\ell,h}, \mathbf{V}_{\ell,h}, \mathcal{C}[\mathcal{S}_{b,h}^{*\ell}])$
    end for
    $\mathbf{h}_\ell \leftarrow \mathrm{FFN}_\ell(\mathbf{o}_\ell)$
  end for
  Unmask high-confidence positions in the block
end for
$\mathcal{C} \leftarrow \mathcal{C} \,\|\, \mathbf{KV}(\mathbf{x})$
return $\mathbf{x}$

§3.2, MAGE introduces an asynchronous dual-stream execution that hides the selection cost behind the layer's FFN, leaving the main forward pass as MAGE's sole non-sparse work. The procedure is training-free and applies to any pretrained block-diffusion LLM.

Phase 1: Exact Selection at Step 1. At the start of each block's trajectory the B block positions are all `[MASK]`, and Phase 1 splits work across two CUDA streams (Algorithm 1): a MAIN stream runs a full forward pass (FlashAttention [Dao et al., 2022] + FFN) without materializing the score matrix, while a SELECT stream independently re-derives the prefix attention scores. At each layer ℓ, the MAIN stream hands off $\mathbf{Q}_\ell^{(1)}$ right after the QKV projection and proceeds to $\mathrm{FlashAttn}_\ell$ and FFN_ℓ; the SELECT stream computes $\mathbf{A}_\ell^{(1)} = \mathrm{softmax}(\mathbf{Q}_\ell^{(1)} \mathbf{K}^\top / \sqrt{d}) \in \mathbb{R}^{G \times B \times N}$ (N the prefix length, G the query heads per KV group) and, for each KV head h, averages over the GB queries that share the head to select the top-k prefix positions:

$$\mathcal{S}_{b,h}^{*\ell} \;=\; \text{Top-}k\Big(\tfrac{1}{GB} \sum_{g \in \mathrm{group}(h)} \sum_{q \in b} \mathbf{A}_\ell^{(1)}[g, q, :] \Big). \tag{5}$$

This is the per-block KV subset of Eq. 2 computed from the exact attention scores at Step 1, so no estimator approximation is involved. The dual-stream design hides the SELECT stream's cost behind FFN_ℓ: FFN_ℓ is compute-intensive while $\mathbf{QK}^\top$ is bound by HBM bandwidth, so the two kernels

Table 2: LongBench per-task and average accuracy for all (model, method, budget) configurations.

			k=256			k=512			k=1024		
Model	Task (↑)	Exact	MAGE	Quest	SparseD	MAGE	Quest	SparseD	MAGE	Quest	SparseD
Fast-dLLM v2 7B	HotpotQA	42.38	37.98	32.13	**38.72**	39.33	34.74	**40.93**	41.29	38.06	**42.32**
	NrtvQA	23.63	**22.56**	14.98	21.83	**24.23**	15.31	22.93	22.42	19.06	**23.42**
	Qasper	36.66	**31.43**	25.82	25.05	**33.41**	31.60	31.65	34.05	**34.54**	33.50
	QMSum	24.16	**23.92**	21.68	20.95	**23.94**	22.85	22.90	**24.79**	23.10	23.34
	RepoB-P	51.84	49.81	41.20	**51.05**	51.71	46.69	**51.96**	**52.36**	49.57	51.88
	TriviaQA	75.49	**75.17**	57.72	69.66	**75.65**	61.86	70.20	**74.17**	66.24	72.51
	Avg	42.36	**40.15**	32.26	37.88	**41.38**	35.51	40.09	**41.51**	38.43	41.16
SDAR-8B-Chat	HotpotQA	59.59	**55.87**	45.81	50.74	**57.17**	49.93	56.54	57.00	55.66	**60.03**
	NrtvQA	28.94	**28.86**	20.11	23.01	**28.74**	24.19	26.22	**29.31**	26.80	28.80
	Qasper	31.02	**30.89**	22.70	17.98	**31.11**	25.68	26.47	**31.51**	28.19	29.47
	QMSum	24.41	**23.57**	18.90	20.30	**23.97**	21.48	21.75	**23.59**	23.27	23.00
	RepoB-P	50.04	46.56	33.17	**46.66**	**48.41**	41.29	48.38	48.66	46.78	**48.95**
	TriviaQA	88.83	**89.63**	78.32	78.75	**88.21**	86.23	86.70	88.16	**89.36**	87.39
	Avg	47.14	**45.90**	36.50	39.57	**46.27**	41.47	44.34	**46.37**	45.01	46.27
LLaDA 2.0 mini	HotpotQA	60.29	**56.75**	48.56	54.20	58.48	55.37	**59.65**	59.61	57.39	**60.15**
	NrtvQA	27.91	**26.19**	13.74	21.95	**28.48**	18.40	24.90	**28.16**	24.30	26.22
	Qasper	46.19	**45.23**	32.89	32.87	**46.41**	40.50	44.25	**46.38**	45.65	44.72
	QMSum	22.86	**22.82**	19.42	20.21	**22.86**	21.43	22.05	**23.06**	22.02	22.32
	RepoB-P	57.11	55.52	46.86	**55.88**	56.18	53.83	**56.74**	56.71	**57.59**	57.33
	TriviaQA	86.43	**87.49**	80.71	85.15	**86.70**	84.26	86.43	86.42	**86.93**	86.13
	Avg	50.13	**49.00**	40.36	45.04	**49.85**	45.63	49.00	**50.06**	48.98	49.48

target different hardware resources and run without contention. The streams synchronize once at the end of Phase 1, leaving the main forward pass as MAGE's sole non-sparse cost—amortized to $1/T$ per step over the T-step trajectory.

Phase 2: Sparse Denoising for Step $s = 2, ..., T$. At every subsequent denoising step, attention at each layer/head retrieves only the k KV pairs at indices $\mathcal{S}^{*\ell}_{b,h}$ and ignores the remaining prefix entries. The model then unmasks high-confidence positions in the block as in standard block-diffusion decoding. Because the selection is fixed once at Step 1 and reused unchanged across all heads, layers, and steps within the block, no re-estimation is performed at any $s \geq 2$.

5 Evaluation

Setup. We evaluate MAGE on three open block-diffusion LLM families: Fast-dLLM v2 7B [Wu et al., 2026a], SDAR-8B-Chat [Cheng et al., 2026], and LLaDA 2.0 mini [Bie et al., 2025]. We compare against the dense baseline—*Exact Attention* (FlashInfer [Ye et al., 2025b])—and two sparse-attention baselines, Quest [Tang et al., 2024] and SparseD [Wang et al., 2026c], both adapted to block-diffusion decoding via per-block KV selection. Following their original designs, Quest re-estimates the per-block top-k at every denoising step with page size p=16, and SparseD performs exact attention for the first 20% of the total denoising steps across all blocks before reusing the captured sparse pattern for the remaining 80%. All sparse methods share the same KV budget $k \in \{256, 512, 1024\}$ and retain exact attention for the first two layers (layers 1–2) across all methods for a fair comparison. We use block size B=32 and unmasking confidence threshold p_{conf}=0.95 [Bie et al., 2025]. All sparse-attention kernels are implemented with FlashInfer, and experiments run on a NVIDIA H100 GPU.

5.1 Accuracy

LongBench. We evaluate on six LongBench [Bai et al., 2024] long-context tasks spanning single-document QA (NarrativeQA, Qasper), multi-document QA (HotpotQA), few-shot learning (TriviaQA), summarization (QMSum), and code completion (RepoBench-P). Table 2 reports per-task scores (F1 / ROUGE-L / EditSim depending on task) and averages across the six tasks for each (model, method, budget) configuration.

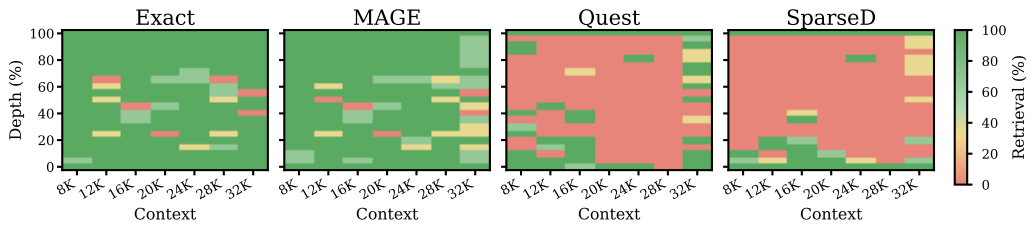

Figure 5: NIAH retrieval on LLaDA 2.0 mini at k=256. x: context length (8K–32K); y: needle depth (0–100%); color: retrieval score. MAGE ≈ Exact, SparseD collapses on most cells, Quest fails across most cells beyond 8K. B=32

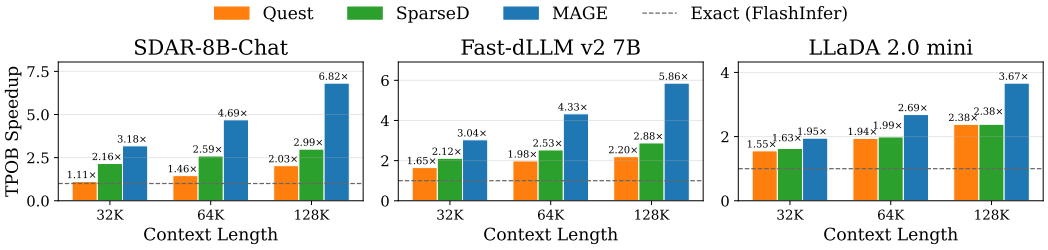

Figure 6: End-to-end TPOB (time-per-output-block) speedup over Exact Attention (FlashInfer) on three block-diffusion LLM families. Budget k=1024, 32 steps/block.

At k=512, MAGE reaches near-lossless accuracy with average drops of only 0.98/0.87/0.28 on Fast-dLLM v2 7B/SDAR-8B-Chat/LLaDA 2.0 mini, far below Quest (6.85/5.67/4.50) and SparseD (2.27/2.80/1.13); even at k=256, MAGE's drops (2.21/1.24/1.13) stay well under Quest's (10.10/10.64/9.77) and SparseD's (4.48/7.57/5.09). MAGE's margin is largest on retrieval-heavy tasks (NarrativeQA, HotpotQA), because MAGE accurately extracts the KV subset each block needs whereas Quest's page-anchoring incurs approximation error and SparseD never refreshes its captured pattern, propagating errors across later blocks.

Needle-in-a-Haystack. Fig. 5 shows NIAH heatmaps for LLaDA 2.0 mini at k=256 over context lengths 8K–32K. MAGE closely tracks Exact with only a few scattered failures, whereas SparseD shows widespread red/yellow cells across most (context, depth) positions, and Quest collapses into near-total failure beyond 12K. At 8K, retrieval scores are MAGE 97, SparseD 17, Quest 41 (Exact 98); per-context scores for all three models at $k \in \{256, 512, 1024\}$ are in Table 3 (Appendix A).

5.2 Efficiency

End-to-End Speedup. Fig. 6 reports MAGE's end-to-end TPOB (time-per-output-block) speedup over Exact Attention (FlashInfer) on all three block-diffusion families at three context lengths (32K, 64K, 128K) with a fixed sparse budget k=1024 and n=32 block steps; the 64K and 128K settings use NTK-based RoPE extrapolation [Liu et al., 2026] to extend each model's context window. Across every (model, context) pair, the speedup grows with context length as the savings of sparse attention scale with N. At 128K context, MAGE's speedup over Exact reaches $\mathbf{6.82}\times$ on SDAR-8B-Chat, $\mathbf{5.86}\times$ on Fast-dLLM v2 7B, and $\mathbf{3.67}\times$ on LLaDA 2.0 mini — consistently ahead of both Quest and SparseD on every (model, context) cell, running up to $\mathbf{3.35}\times$ faster than Quest and $\mathbf{2.28}\times$ faster than SparseD. That MAGE dominates across three independent block-diffusion implementations indicates the gain is robust to the model rather than a property of any single one. The same ranking holds under shorter block-step budgets $n \in \{24, 16, 8\}$; see Appendix B.

Latency Breakdown. Fig. 7 decomposes latency of one denoising step at 64K, k=1024 into Attention, Estimation overhead, and MISC, with Exact as a dashed reference. MAGE's first step

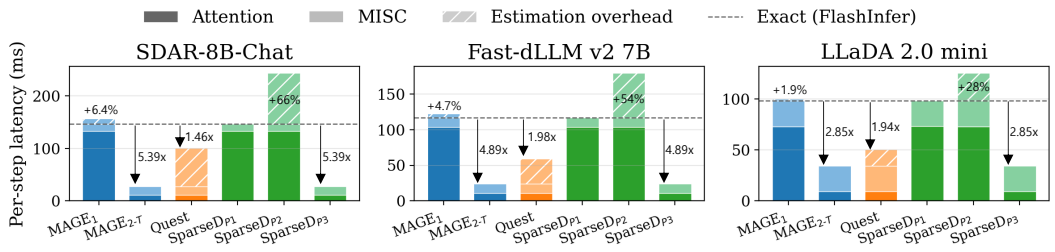

Figure 7: Per-step latency breakdown at 64K, k=1024. Bars decompose one denoising step into Attention, Estimation overhead (hatched), and MISC; the dashed line marks Exact Attention (FlashInfer).

(MAGE $_1$) adds top-k selection on top of an exact pass, but the selection runs on the asynchronous SELECT stream in parallel with FFN (Algorithm 1), keeping critical-path overhead at $+\mathbf{1.9}\%$–$+\mathbf{6.4}\%$ over Exact. The remaining $n-1$ steps (MAGE $_{2\text{-}T}$) run pure sparse attention at $\mathbf{2.85}\times$–$\mathbf{5.39}\times$ over Exact, amortizing the single exact pass per block. Quest recomputes its page-anchor estimate on the critical path at every step, capping per-step speedup at $\mathbf{1.46}\times$–$\mathbf{1.98}\times$. SparseD runs exact attention for the first 20% of steps across the trajectory (P1), takes a capture step at the boundary (P2, $+\mathbf{28}\%$–$+\mathbf{66}\%$), and runs sparse attention for the remaining 80% (P3, matching MAGE $_{2\text{-}T}$); MAGE keeps only $1/n$ of steps exact while SparseD keeps 20%, so the gap accumulates end-to-end.

6 Limitations

MAGE targets the attention bottleneck that dominates long-context inference, so its end-to-end benefit diminishes in short-context regimes where FFN and other non-attention computations account for most of the latency. In addition, MAGE reduces KV cache access time but not its memory footprint; alleviating the memory pressure of long-context KV caches requires complementary techniques such as KV offloading [Lee et al., 2024, Sun et al., 2025, Kwon et al., 2026] or cache compression, which we leave to future work.

7 Conclusion

We present MAGE, a training-free sparse-attention method for block-diffusion LLMs. By exploiting the All-`[MASK]` anchoring property induced by block-diffusion training, MAGE runs one exact attention pass at Step 1 and reuses the resulting per-block top-k indices across the remaining $T-1$ steps, with selection cost hidden asynchronously behind the FFN. Across three block-diffusion families, MAGE matches Exact Attention at k=512 with near-lossless accuracy while achieving up to $6.82\times$ end-to-end speedup at 128K context.

8 Acknowledgments

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2026-25524173, Ultra-Long-Term Hierarchical Memory and Reasoning Architecture for Next-Generation Omnimodal Agents). Additional support was provided by Institute of Information & communications Technology Planning & Evaluation (IITP) under the Graduate School of Artificial Intelligence Semiconductor (IITP-2026-RS-2023-00256081) grant funded by the Korea government (MSIT). Samsung Electronics Co., Ltd also supported this work (IO260225-15628-01).

References

J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebron, and S. Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.298. URL https://aclanthology.org/2023.emnlp-main.298/.

M. Arriola, A. Gokaslan, J. Chiu, Z. Yang, Z. Qi, J. Han, S. Sahoo, and V. Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 50726–50753, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/7ede97c3e082c6df10a8d6103a2eebd2-Paper-Conference.pdf.

Y. Bai, X. Lv, J. Zhang, H. Lyu, J. Tang, Z. Huang, Z. Du, X. Liu, A. Zeng, L. Hou, Y. Dong, J. Tang, and J. Li. LongBench: A bilingual, multitask benchmark for long context understanding. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150.

T. Bie, M. Cao, K. Chen, L. Du, M. Gong, Z. Gong, Y. Gu, J. Hu, Z. Huang, Z. Lan, C. Li, C. Li, J. Li, Z. Li, H. Liu, L. Liu, G. Lu, X. Lu, Y. Ma, J. Tan, L. Wei, J.-R. Wen, Y. Xing, X. Zhang, J. Zhao, D. Zheng, J. Zhou, J. Zhou, Z. Zhou, L. Zhu, and Y. Zhuang. LLaDA2.0: Scaling up diffusion language models to 100b, 2025. URL https://arxiv.org/abs/2512.15745.

Z. Cai, Y. Zhang, B. Gao, Y. Liu, Y. Li, T. Liu, K. Lu, W. Xiong, Y. Dong, J. Hu, and W. Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=ayi7qezU87.

S. Cheng, Y. Bian, D. Liu, Y. Jiang, Y. Liu, L. Zhang, Q. Yao, Z. Tian, W. Wang, Q. Guo, K. Chen, B. Qi, and B. Zhou. SDAR: A synergistic diffusion-AutoRegression paradigm for scalable sequence generation. In M. Liakata, V. P. Moreira, J. Zhang, and D. Jurgens, editors, *Findings of the Association for Computational Linguistics: ACL 2026*, pages 22058–22075, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1110. URL https://aclanthology.org/2026.findings-acl.1110/.

T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré. FlashAttention: Fast and memory-efficient exact attention with io-awareness. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359. Curran Associates, Inc., 2022. doi: 10.52202/068431-1189. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/67d57c32e20fd0a7a302cb81d36e40d5-Paper-Conference.pdf.

Y. Feng, J. Lv, Y. Cao, X. Xie, and S. K. Zhou. Ada-KV: Optimizing KV cache eviction by adaptive budget allocation for efficient LLM inference. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 113152–113188. Curran Associates, Inc., 2025. doi: 10.52202/085713-3773. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a40ff56daab9f4808b1e18350c8a11ce-Paper-Conference.pdf.

Y. Fu, L. Whalen, A. Garg, C. Wu, M. Khadkevich, N. Oswald, E. Xie, D. Egert, S. Turuvekere Sreenivas, S. Diao, C. D. Yu, Y. Yu, W. Chen, S. Norouzi, J. Liu, S. Lan, L. Zhu, J. Wang, J. Jiang, M. Mardani, M. Maghoumi, S. Han, A. Jukić, N. Tajbakhsh, J. Kautz, and P. Molchanov. Nemotron-labs-diffusion: A tri-mode language model unifying autoregressive, diffusion, and self-speculation decoding. In H.-Y. Lee, S. Dev, J. Li, and D. Elliott, editors, *Proceedings of the 2026 Conference on Empirical Methods in Natural Language*

Processing, pages 54868–54887, Budapest, Hungary, Oct. 2026. Association for Computational Linguistics. ISBN 979-8-89176-459-0. doi: 10.18653/2026.emnlp-main.2416. URL https://aclanthology.org/2026.emnlp-main.2416/.

S. Gong, S. Agarwal, Y. Zhang, J. Ye, L. Zheng, M. Li, C. An, P. Zhao, W. BI, J. Han, H. Peng, and L. Kong. Scaling diffusion language models via adaptation from autoregressive models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 5046–5073, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/0fa81c3f0d57f95b8776de3a248ef0ed-Paper-Conference.pdf.

Google DeepMind. Gemini diffusion. https://deepmind.google/models/gemini-diffusion/, 2025. Accessed: 2026-04-14.

J. Huang, Y. Zhang, Y. Yang, B. Huang, and L. Zhang. Mask tokens as prophet: Fine-grained cache eviction for efficient dLLM inference. In M. Liakata, V. P. Moreira, J. Zhang, and D. Jurgens, editors, *Findings of the Association for Computational Linguistics: ACL 2026*, pages 3456–3479, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.170. URL https://aclanthology.org/2026.findings-acl.170/.

M. Kim, C. Hooper, A. Tomar, C. Xu, M. Farajtabar, M. W. Mahoney, K. Keutzer, and A. Gholami. Beyond next-token prediction: A performance characterization of diffusion versus autoregressive language models. *arXiv preprint arXiv:2510.04146*, 2025. URL https://arxiv.org/abs/2510.04146.

M. Kim, C. Xu, C. Hooper, H. Singh, B. Athiwaratkun, C. Zhang, K. Keutzer, and A. Gholami. CDLM: Consistency diffusion language models for faster sampling. In A. Chowdhery and Z. Jia, editors, *Proceedings of Machine Learning and Systems*, volume 8, pages 1148–1165. MLSys, 2026. URL https://proceedings.mlsys.org/paper_files/paper/2026/file/054de805fcceb78a201f5e9d53c85908-Paper-Conference.pdf.

O. Kwon, D. Kim, J. Park, S. Y. Lee, S. Kim, I. Stoica, and J. W. Lee. HERALD: High-throughput block diffusion LLM inference via CPU-GPU cooperative KV cache retrieval, 2026. URL https://arxiv.org/abs/2606.21633.

W. Lee, J. Lee, J. Seo, and J. Sim. InfiniGen: Efficient generative inference of large language models with dynamic KV cache management. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pages 155–172, Santa Clara, CA, July 2024. USENIX Association. ISBN 978-1-939133-40-3. URL https://www.usenix.org/conference/osdi24/presentation/lee.

Y. Lee, S. Dan, J. Lee, J. Park, and J. H. Ahn. DyLLM: Efficient diffusion LLM inference via saliency-based token selection and partial attention. In T. Zhang, M. Dudik, M. Jaggi, A. Agarwal, S. Li, D. Schuurmans, J. Zhu, F. Berkenkamp, H. Dong, and A. Bietti, editors, *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 63635–63655. PMLR, 06–11 Jul 2026. URL https://proceedings.mlr.press/v306/lee26i.html.

Y. Li, Y. Huang, B. Yang, B. Venkitesh, A. Locatelli, H. Ye, T. Cai, P. Lewis, and D. Chen. SnapKV: LLM knows what you are looking for before generation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 22947–22970. Curran Associates, Inc., 2024. doi: 10.52202/079017-0722. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/28ab418242603e0f7323e54185d19bde-Paper-Conference.pdf.

C. Lin, J. Tang, S. Yang, H. Wang, T. Tang, B. Tian, I. Stoica, S. Han, and M. Gao. Twilight: Adaptive attention sparsity with hierarchical top-p pruning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 119406–119432. Curran Associates, Inc., 2025. doi: 10.52202/085713-3986. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/ad2f57116d62a9d8dcfaba6a168715e4-Paper-Conference.pdf.

X. Liu, Y. Song, Z. Liu, Z. Huang, Q. Guo, Z. He, and X. Qiu. LongLLaDA: unlocking long context capabilities in diffusion LLMs. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'26/IAAI'26/EAAI'26. AAAI Press, 2026. ISBN 978-1-57735-906-7. doi: 10.1609/aaai.v40i38.40491. URL https://doi.org/10.1609/aaai.v40i38.40491.

Z. Liu, A. Desai, F. Liao, W. Wang, V. Xie, Z. Xu, A. Kyrillidis, and A. Shrivastava. Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 52342–52364. Curran Associates, Inc., 2023. doi: 10.52202/075280-2279. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a452a7c6c463e4ae8fbdc614c6e983e6-Paper-Conference.pdf.

X. Ma, R. Yu, G. Fang, and X. Wang. dKV-Cache: The cache for diffusion language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 149009–149033. Curran Associates, Inc., 2025. doi: 10.52202/085713-4984. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/db0ee27cb50dd9087b133f6e7d28a90e-Paper-Conference.pdf.

S. Nie, F. Zhu, Z. You, X. Zhang, J. Ou, J. Hu, J. Zhou, Y. Lin, J.-R. Wen, and C. LI. Large language diffusion models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 50608–50646. Curran Associates, Inc., 2025. doi: 10.52202/085713-1689. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/48b383b24230e0e6e649d9c98dae4d8c-Paper-Conference.pdf.

R. Pope, S. Douglas, A. Chowdhery, J. Devlin, J. Bradbury, J. Heek, K. Xiao, S. Agrawal, and J. Dean. Efficiently scaling transformer inference. In D. Song, M. Carbin, and T. Chen, editors, *Proceedings of Machine Learning and Systems*, volume 5, pages 606–624. Curan, 2023. URL https://proceedings.mlsys.org/paper_files/paper/2023/file/c4be71ab8d24cdfb45e3d06dbfca2780-Paper-mlsys2023.pdf.

L. Ribar, I. Chelombiev, L. Hudlass-Galley, C. Blake, C. Luschi, and D. Orr. SparQ attention: bandwidth-efficient LLM inference. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

P. Singhania, S. Singh, S. He, S. Feizi, and A. Bhatele. Loki: Low-rank keys for efficient sparse attention. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 16692–16723. Curran Associates, Inc., 2024. doi: 10.52202/079017-0532. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/1e027da6bec9ceb2ec37951ceeccae93-Paper-Conference.pdf.

Y. Song, Z. Zhang, C. Luo, P. Gao, F. Xia, H. Luo, Z. Li, Y. Yang, H. Yu, X. Qu, Y. Fu, J. Su, G. Zhang, W. Huang, M. Wang, L. Yan, X. Jia, J. Liu, W.-Y. Ma, Y.-Q. Zhang, Y. Wu, and H. Zhou. Seed diffusion: A large-scale diffusion language model with high-speed inference, 2025. URL https://arxiv.org/abs/2508.02193.

Y. Song, X. Liu, R. Li, Z. Liu, Z. Huang, Q. Guo, Z. He, and X. Qiu. Sparse-dLLM: Accelerating diffusion LLMs with dynamic cache eviction. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'26/IAAI'26/EAAI'26. AAAI Press, 2026. ISBN 978-1-57735-906-7. doi: 10.1609/aaai.v40i39.40586. URL https://doi.org/10.1609/aaai.v40i39.40586.

H. Sun, L.-W. Chang, W. Bao, S. Zheng, N. Zheng, X. Liu, H. Dong, Y. Chi, and B. Chen. ShadowKV: KV cache in shadows for high-throughput long-context LLM inference. In A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 57355–57373. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/sun25b.html.

J. Tang, Y. Zhao, K. Zhu, G. Xiao, B. Kasikci, and S. Han. QUEST: Query-aware sparsity for efficient long-context LLM inference. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 47901–47911. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/tang24l.html.

X. Wang, C. Xu, Y. Jin, J. Jin, H. Zhang, K. Yu, and Z. Deng. Diffusion LLMs can do faster-than-AR inference via discrete diffusion forcing. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 2282–2299, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/04bb76a98d9f32226b3beba7bd26a51f-Paper-Conference.pdf.

Y. Wang, L. Yang, B. Li, Y. Tian, K. Shen, and M. Wang. Revolutionizing reinforcement learning framework for diffusion large language models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 101476–101501, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a61023ce36d21010f1423304f8ec49af-Paper-Conference.pdf.

Z. Wang, G. Fang, X. Ma, X. Yang, and X. Wang. SparseD: Sparse attention for diffusion language models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 99320–99337, 2026c. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a1598ba0aa7bd7bdf18e0ad04d993edc-Paper-Conference.pdf.

C. Wu, H. Zhang, S. Xue, S. Diao, Y. Fu, Z. Liu, P. Molchanov, P. Luo, S. Han, and E. Xie. Fast-dLLM v2: Efficient block-diffusion LLM. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 128353–128370, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/d0865cbe51d35ec322f9af9db7806fc7-Paper-Conference.pdf.

C. Wu, H. Zhang, S. Xue, Z. Liu, S. Diao, L. Zhu, P. Luo, S. Han, and E. Xie. Fast-dLLM: Training-free acceleration of diffusion LLM by enabling KV cache and parallel decoding. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 57027–57051, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/5d8d4e6061c3ba96c240b7fa1ae3471d-Paper-Conference.pdf.

Z. Wu, L. Zheng, J. Ye, S. Gong, X. Zhao, Y. Feng, W. Bi, and L. Kong. Dreamreasoner-8b: Block-size curriculum learning for diffusion reasoning models, 2026c. URL https://arxiv.org/abs/2606.19257.

G. Xiao, Y. Tian, B. Chen, S. Han, and M. Lewis. Efficient streaming language models with attention sinks. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 21875–21895, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5e5fd18f863cbe6d8ae392a93fd271c9-Paper-Conference.pdf.

L. Yang, Z. Zhang, Z. Chen, Z. Li, and Z. Jia. TidalDecode: Fast and accurate LLM decoding with position persistent sparse attention. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 5823–5844, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/11440c427f0f76f191ac06b50d7a2517-Paper-Conference.pdf.

L. Yang, Z. Zhang, A. Jain, S. Cao, B. Yuan, Y. Chen, Z. Jia, and R. Netravali. Less is more: Fast and accurate reasoning with cross-head unified sparse attention. In T. Zhang, M. Dudik, M. Jaggi, A. Agarwal, S. Li, D. Schuurmans, J. Zhu, F. Berkenkamp, H. Dong, and A. Bietti, editors, *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 147014–147031. PMLR, 06–11 Jul 2026. URL https://proceedings.mlr.press/v306/yang26dj.html.

J. Ye, Z. Xie, L. Zheng, J. Gao, Z. Wu, X. Jiang, Z. Li, and L. Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025a. URL https://arXiv.org/abs/2508.15487.

Z. Ye, L. Chen, R. Lai, W. Lin, Y. Zhang, S. Wang, T. Chen, B. Kasikci, V. Grover, A. Krishnamurthy, and L. Ceze. FlashInfer: Efficient and customizable attention engine for LLM inference serving. In M. Zaharia, G. Joshi, and Y. Lin, editors, *Proceedings of Machine Learning and Systems*, volume 7. MLSys, 2025b. URL https://proceedings.mlsys.org/paper_files/paper/2025/file/dbf02b21d77409a2db30e56866a8ab3a-Paper-Conference.pdf.

M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed. Big Bird: Transformers for longer sequences. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/c8512d142a2d849725f31a9a7a361ab9-Paper.pdf.

Z. Zhang, Y. Sheng, T. Zhou, T. Chen, L. Zheng, R. Cai, Z. Song, Y. Tian, C. Ré, C. Barrett, Z. A. Wang, and B. Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 34661–34710. Curran Associates, Inc., 2023. doi: 10.52202/075280-1506. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6ceefa7b15572587b78ecfcebb2827f8-Paper-Conference.pdf.

K. Zhu, T. Tang, Q. Xu, Z. Jin, Y. Gu, Z. Zeng, R. Kadekodi, L. Zhao, A. Li, A. Krishnamurthy, and B. Kasikci. Tactic: Adaptive sparse attention with clustering and distribution fitting for long-context LLMs. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 30626–30644, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/33f94d79acf71051d6a27f4d8889e20e-Paper-Conference.pdf.

A NIAH Scores Across Models

Table 3 reports per-context-length NIAH retrieval scores at 8K and 32K context for Exact Attention and the three sparse methods (MAGE, Quest, SparseD) at $k \in \{256, 512, 1024\}$ across all three block-diffusion model families. Each cell is the mean retrieval score over the 21 needle-depth grid at that context length. Bold marks the best sparse method (MAGE/ Quest / SparseD) per column within each model. Fig. 5 in the main text visualizes the underlying (context, depth) cells for LLaDA 2.0 mini at k=256.

Table 3: Needle-in-a-Haystack accuracy (%) at 8K and 32K context, averaged over depth positions, for sparse budgets $k \in \{256, 512, 1024\}$ across three block-diffusion models. Bold = best sparse method (MAGE / Quest / SparseD) per column within each model.

	SDAR-8B-Chat						*LLaDA 2.0 mini*						*Fast-dLLM v2 7B*					
	8K			32K			8K			32K			8K			32K		
Method	256	512	1024	256	512	1024	256	512	1024	256	512	1024	256	512	1024	256	512	1024
Exact	100	100	100	57	57	57	98	98	98	90	90	90	100	100	100	46	46	46
MAGE	**100**	**100**	**100**	**76**	**71**	**67**	**97**	**98**	98	**62**	68	**76**	89	94	**100**	**33**	**37**	46
Quest	90	100	100	16	16	16	41	48	60	57	**70**	75	**97**	98	100	22	35	**52**
SparseD	68	100	100	10	38	49	17	68	**100**	25	65	70	56	**100**	100	13	27	48

B End-to-end Speedup at Shorter Block-step Budgets

The headline speedups in Fig. 6 use n=32 steps/block; Fig. 8 extends this to shorter budgets (n=24, 16, 8) on the same three model families. MAGE remains the fastest method on every (model, context) cell, with the absolute speedup tightening slightly at n=8 as its first-step dense pass amortizes over fewer sparse steps.

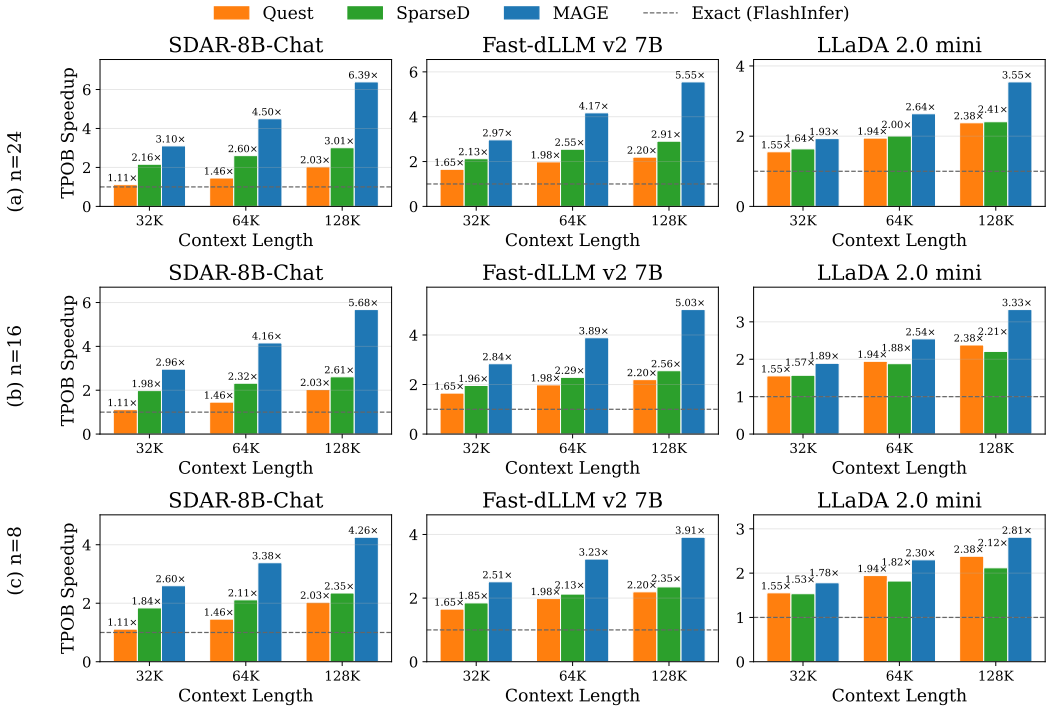

Figure 8: End-to-end TPOB speedup over Exact Attention (FlashInfer) at (a) n=24 steps/block, (b) n=16 steps/block, (c) n=8 steps/block. budget k=1024.

C Per-Token vs. Per-Block Sparse Attention Kernels

A block could in principle let each of its B tokens attend to its own per-token KV subset. We implement this alternative by treating each token as an independent decode request with its own KV-cache accesses, executed with FlashInfer's batched paged-attention kernel, and compare it with the per-block kernel of MAGE, in which all B queries attend to a single shared KV subset (Table 4). The per-block kernel is 2.5–$7.9\times$ faster, because the irregular accesses of the per-token kernel offset most of the benefit of reading fewer KV entries. A shared per-block KV subset is therefore necessary for efficient sparse attention in block dLLMs.

Table 4: Sparse attention kernel latency with per-block and per-token KV subsets (B=32, k=1024, H100).

Request batch	Per-block (ms)	Per-token (ms)	Per-block speedup
1	0.026	0.066	$2.5\times$
4	0.047	0.242	$5.2\times$
16	0.110	0.874	$7.9\times$
64	0.469	3.711	$7.9\times$

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately state our identification of the per-block selection challenge, the All-[MASK] anchoring property, and MAGE's training-free design.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss limitations in a dedicated Limitations section, noting that MAGE's benefit is concentrated in long-context regimes and that it reduces KV cache access time but not its memory footprint.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The derivation of the block-average query gap bound (Eqs. 4-7) states its assumptions and is presented in full in §3.2.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: MAGE is a training-free algorithm fully specified in Algorithm 1, with all experimental settings (models, baselines, budgets, hyperparameters, hardware) detailed in §5.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: We provide an anonymized code package as supplemental material, including scripts to reproduce the main experiments on the publicly available models and benchmarks used in §5.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: §5 specifies all evaluation settings including models, baselines, sparse budgets, block size, decoding threshold, and hardware.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Speed measurements are reported as the median over 30 runs to mitigate timing noise, though error bars are not included; accuracy results follow the standard deterministic evaluation protocol of LongBench and NIAH.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: §5 specifies that all experiments run on a single NVIDIA H100 GPU.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and our work conforms to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: MAGE is a general-purpose inference acceleration technique with no direct path to specific societal applications beyond those of the underlying LLMs it accelerates.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: We release no new models or datasets; MAGE is an inference-time technique applied to existing publicly available models.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All models, benchmarks, kernels, and baseline methods used are publicly available and properly cited in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The supplemental code package includes documentation describing how to run MAGE and reproduce the experiments in §5.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: This work does not involve human subjects research.

    Guidelines:

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

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are the target of our acceleration method but are not used as a component of the core methodology.

    Guidelines:

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