LOCKS: PAGE-LOCAL COMPACT KEY SUMMARIES FOR EFFICIENT LONG-CONTEXT DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Serving large language models at long context is bottlenecked by the key-value (KV) cache, which is read at every decode step. We find that attention keys are approximately low-rank within pages. A single low-rank projection shared across pages can miss page-specific directions; fitting a basis to each page better identifies the pages receiving the most attention at comparable stored selector cost. LOCKS stores a rank-r spectral summary per page, reconstructs its within-page logits, and selects pages by log-sum-exp mass without reading candidate keys or values. It stays within about a point of FullKV on LongBench-v1, tracks the read-every-key exact-LSE oracle on RULER down to the smallest budgets, and retains quality furthest under tight budgets on AIME26 and MATH-500. At a 2048-token budget it matches FullKV aggregate quality beyond 100K context while attending about 2% of tokens. Across ranks 2–8, summaries use 4–10% of full-KV bytes. On GH200 with GPU-resident KV, LOCKS reduces complete decode-step time by $1.8\times$ at 512K context. With full KV offloaded to Grace memory, it reaches 3.82–$4.22\times$ the faster dense backend's aggregate throughput at 64K–256K by serving larger batches.

1 INTRODUCTION

Every decode step reads two things: the model's weights and the sequence's key-value (KV) cache. At short context the step is weight-bound; at long context and useful batch sizes, weights are shared across requests but each request still reads its own cache. The cache also limits the serving capacity. For Llama-3.1-8B (Grattafiori et al., 2024) with bf16 KV, a 128K context occupies roughly 16 GiB per request, more than the model's weights, and constrains both context length and concurrency.

Decode attention is sparse. A small fraction of the cache carries most of a query's attention mass, and attending only that fraction often recovers the full-cache output (Zhang et al., 2023; Tang et al., 2024; Ribar et al., 2024). The difficulty is finding it without scoring every key, which would restore the traffic sparsity removes. Selection therefore needs a small resident summary.

Sparse decoding succeeds only if its small budget is spent on the right pages. Figure 1d makes the stakes concrete on RULER-32K: at b=64, LOCKS scores 100% on NIAH-S3 and 75.2% on VT. In the separate exclusion test, removing its selected pages while retaining the rest of the cache yields 0% and 10.4%, respectively. The challenge is to identify useful pages without first reading every candidate key.

That decision is limited by the resident page summary (Fig. 1a). Quest's coordinate-wise extrema give an optimistic bound on a page's largest token logit; the extrema can come from different tokens, so the bound need not describe any token in the page. LOCKS instead fits a compact basis to each page, reconstructs its token logits, and uses their log-sum-exp to estimate the page's attention mass. Exact page-LSE selection minimizes worst-case attention-output error among deterministic value-blind rules, but computing it reads every key (§2). Under a common GQA-shared budget, Fig. 1b tests the consequence of these summaries in the attention output computed from original KV; panel (c) tests whether fitting the basis helps beyond merely storing the same number of arbitrary directions. The figure thus motivates *what the selector must preserve* and why a page-fitted spectral summary is a plausible way to preserve it.

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

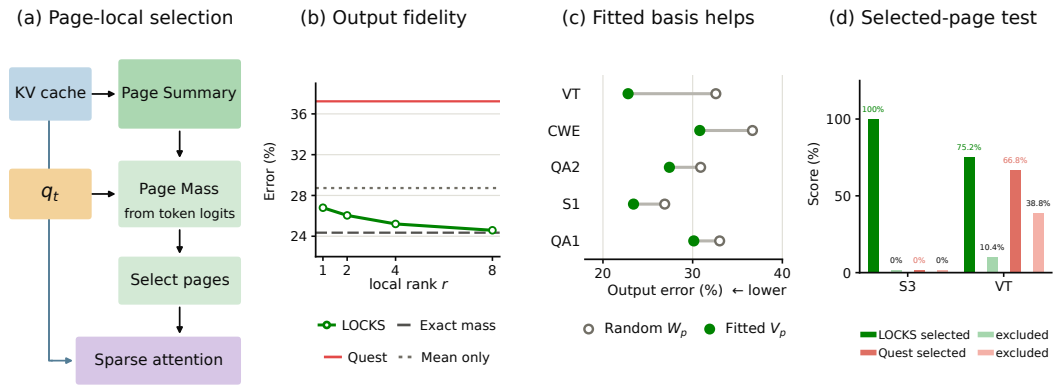

Figure 1: **(a)** LOCKS scores page mass from a resident page summary, then attends with original KV from the selected pages. **(b)** Relative ℓ_2 error against the full-cache attention output before the output projection (Llama-3.1-8B-Instruct, B=16, eight pages per KV head). **(c)** A separate controlled rank-4 ablation replaces random directions with each page's fitted basis at the same summary size and token budget. Each pair connects one sequence's output errors. **(d)** RULER-32K S3 and VT at b=64, with rank-4 LOCKS summaries. Excluded scores measure accuracy after removing each method's selected pages while retaining sink and current pages.

Fitting useful directions leaves a separate question: can the fitted representation be reused across pages, or must its scope be local?

Table 1 compares page-local summaries with shared bases that either retain or replace each page centroid at near-matched storage and b=128. The page-local fit retains more attention mass and recovers more of the exact-LSE page set. Fixed shared projections can also have blind directions (Prop. 1).

Page-local state must also be deployable. Unlike trained gates (Gao et al., 2024), LOCKS is a drop-in backend for an existing paged-attention serving engine. It builds one small factorization when each page completes, then scores and ranks pages in a fused, CUDA-graph-compatible kernel (§4, §5).

We adopt exact page-LSE selection as the deterministic value-blind minimax target (App. A.1) and make three contributions:

- **Locality finding and matched-state evidence.** We show that page-local state improves retained mass and page recall, and that these advantages survive matched summary payload (§3).
- **Shared-projection impossibility and local construction.** We prove blind directions for fixed shared low-rank projections (Prop. 1) and bound the unquantized local summary from reconstruction error through coverage and output error (Thm. 1).
- **Quality and deployability.** We show oracle-tracking quality at aggressive budgets and long-context acceleration without changing the model or the serving engine (§5).

2 PRELIMINARIES

A decoder layer holds per-head caches $\{(\mathbf{k}_i, \mathbf{v}_i)\}_{i=1}^{T}$. A decode step forms a query $\mathbf{q}$ and outputs $\mathbf{o} = \sum_i \alpha_i \mathbf{v}_i$ with $\alpha = \mathrm{softmax}(\mathbf{q}^\top \mathbf{k}/\sqrt{d})$. Paged serving (Kwon et al., 2023) stores the cache in pages of B consecutive tokens. We write P_j for page j and select at page granularity. We use a fixed per-head token budget b, corresponding to $k = \lceil b/B \rceil$ page slots. The sink page and the most recent page always occupy two of those slots. This keeps the attended working set bounded as context length increases.

Coverage bounds sparse decoding's output error. For a page set S, let $\gamma_S(\mathbf{q})$ be the fraction of total attention mass that falls in S, and let $\mathbf{o}_S$ be attention restricted to S with the softmax renormalized. Dropping the complement shifts the output by exactly $\mathbf{o} - \mathbf{o}_S = \varepsilon\,(\bar{\mathbf{v}}_{S^c} - \bar{\mathbf{v}}_S)$, with $\varepsilon = 1 - \gamma_S(\mathbf{q})$ and $\bar{\mathbf{v}}_S, \bar{\mathbf{v}}_{S^c}$ the mass-weighted mean values of the kept and dropped sets (Tzachristas et al., 2025; Tian et al., 2026). At the model's actual values, equal coverage can still yield different output errors.

Selection precedes the value fetch, so a deterministic value-blind rule chooses pages from keys and query. For fixed keys and query, its worst-case output error over $\|\mathbf{v}_i\| \le R$ is exactly $2R(1-\gamma_S)$ (App. A.1). The minimax choice therefore maximizes coverage. Define the exact page log-sum-exp (LSE) score

$$s_j(\mathbf{q}) = \log \sum_{i \in P_j} \exp\left(\frac{\mathbf{q}^\top \mathbf{k}_i}{\sqrt{d}}\right). \tag{1}$$

Because $e^{s_j(\mathbf{q})}$ is the page's unnormalized attention mass, the highest-scoring free pages, together with the always-kept pages, achieve that minimax choice. Computing the exact scores requires reading every cached key, defeating the traffic saving of sparse decoding. LOCKS therefore approximates this value-blind minimax ranking from a compact resident page summary.

3 WHY PAGE-LOCAL STATE MATTERS

Page selection changes substantially with the scope of the low-rank key fit. To examine centering and basis fitting, we compare page-local fits with shared fits that either keep or replace the page centroid. For each completed page j, we compute

$$\boldsymbol{\mu}_j = \frac{1}{B}\sum_{i \in P_j} \mathbf{k}_i, \qquad \mathbf{H}_j = \sum_{i \in P_j} (\mathbf{k}_i - \boldsymbol{\mu}_j)(\mathbf{k}_i - \boldsymbol{\mu}_j)^\top.$$

LOCKS takes the top r eigenvectors of $\mathbf{H}_j$. The hybrid controls retain $\boldsymbol{\mu}_j$ but fit one basis to pooled $\mathbf{H}_j$ across a sequence or calibration set. Strict shared fits use one centroid and basis at either scope. Table 1 compares these choices; App. B.1 gives the fitting and budget details.

Table 1: Centroid and basis scope on RULER-32K (Llama-3.1-8B-Instruct, B=16, b=128 attended tokens per KV head). All rows use 488–490 B of selector state per page and KV head. Mass and recall use common FullKV states and the same page-LSE scorer.

Centroid	Basis	Rank	Mass (%)	Recall (%)	Score (%)
Page	Page (LOCKS r4i4)	4	**85.8**	**90.5**	**80.1**
Page	Sequence	20	84.4	81.3	77.0
Page	Global 4K	20	84.6	81.9	76.3
Page	Global 32K	20	84.5	81.9	75.2
Sequence	Sequence	28	81.2	60.2	47.6
Global 4K	Global 4K	28	81.3	58.0	63.1
Global 32K	Global 32K	28	82.2	63.3	56.9

With page centroids retained, shared bases recall 81.3–81.9% of exact-LSE pages; wholly shared fits recall 58.0–63.3%. The page-local basis reaches 90.5% at similar storage. This observation motivates fitting both parts of the summary per page. Proposition 1 shows a possible failure of a fixed shared basis even when page centroids are known.

Proposition 1 (A blind direction for a fixed shared basis)**.** *Fix any $\mathbf{W} \in \mathbb{R}^{d\times\rho}$ with $1 \le \rho < d$ and page size $B \ge 2$. There are pages P and P_c whose centered keys each have rank at most one and whose centroids and projected keys $\{\mathbf{W}^\top \mathbf{k}_i\}$ agree, but $s_{P_c}(\mathbf{q}) - s_P(\mathbf{q}) \to \infty$ for some query $\mathbf{q}$ as $c \to \infty$. Any scorer using only these shared-projection summaries assigns the pages the same score. An unquantized rank-one basis fitted to each page represents both pages' keys, and hence their LSE, exactly (proof in App. A.2).*

4 LOCKS: LOCAL COMPACT KEY SUMMARIES

Exact-LSE selection requires reading every cached key (§2), while a fixed shared low-rank projection can miss within-page variation (Prop. 1). LOCKS therefore fits a separate basis to each page and uses its resident summary to select pages at each decode step under the fixed per-head token budget b of §2. It is training-free and applies to both standard GQA decoders and recent models with trained sparse attention (App. C.6).

Building the summary We fit the basis to cached keys. For RoPE models, these are post-RoPE keys, paired with the post-RoPE query when scoring; Table C.2 compares pre- and post-RoPE page-local fits. For page j, compute:

$$\boldsymbol{\mu}_j = \frac{1}{B}\sum_{i\in P_j}\mathbf{k}_i, \quad \mathbf{D}_j = \left[(\mathbf{k}_i - \boldsymbol{\mu}_j)^\top\right]_{i\in P_j} \in \mathbb{R}^{B\times d}, \quad \mathbf{C}_j = \mathbf{D}_j\mathbf{V}_j \in \mathbb{R}^{B\times r}.$$

Here $\mathbf{V}_j \in \mathbb{R}^{d\times r}$ contains the top r right singular vectors of $\mathbf{D}_j$, and row $\mathbf{c}_i^\top$ of $\mathbf{C}_j$ gives $\tilde{\mathbf{k}}_i = \boldsymbol{\mu}_j + \mathbf{V}_j\mathbf{c}_i$. The squared page residual is $\tau_{j,r} := \|\mathbf{D}_j - \mathbf{C}_j\mathbf{V}_j^\top\|_F^2 = \sum_{\ell>r}\sigma_{j,\ell}^2$. Before quantization, reconstruction is exact if $r \geq \operatorname{rank}(\mathbf{D}_j)$.

In labels such as r8i4, the first number is the rank and the second is the basis bit width. The i4 variants store $\mathbf{V}_j$ in int4 and $\mathbf{C}_j, \boldsymbol{\mu}_j$ in int8, requiring $dr/2 + Br + d$ bytes per page before scales (App. C.7). Completed pages are summarized once, off the decode critical path.

Scoring from the summary Substitute the reconstructed keys into the page LSE. The common centroid logit then factors out:

$$\begin{aligned}\hat{s}_j(\mathbf{q}) &= \log\sum_{i\in P_j}\exp\left(\frac{\mathbf{q}^\top(\boldsymbol{\mu}_j + \mathbf{V}_j\mathbf{c}_i)}{\sqrt{d}}\right)\\ &= \frac{\mathbf{q}^\top\boldsymbol{\mu}_j}{\sqrt{d}} + \log\sum_{i\in P_j}\exp\left(\frac{(\mathbf{V}_j^\top\mathbf{q})^\top\mathbf{c}_i}{\sqrt{d}}\right),\end{aligned} \tag{2}$$

where $\mathbf{V}_j^\top\mathbf{q}$ is computed once per page for each decode query. The fused ranking kernel reads summaries, not candidate keys or values.

For each query $\mathbf{q}_g$ in a GQA group G, normalize its page scores to obtain $\hat{m}_g(j) = e^{\hat{s}_j(\mathbf{q}_g)}/\sum_{j'}e^{\hat{s}_{j'}(\mathbf{q}_g)}$. The share-average combine scores page j by $\bar{\hat{m}}_j = \frac{1}{G}\sum_{g=1}^G\hat{m}_g(j)$. We always keep the sink and most recent pages, then fill the other $k-2$ slots with the highest-scoring remaining pages. All G query heads use this same k-page set for their shared KV head.

This averaging follows TokenSelect's head soft vote (Wu et al., 2025). With exact shares, it maximizes group-mean retained mass among shared k-page sets containing the two always-kept pages (App. A.4). Thm. 1(ii) bounds the loss from estimated shares, and App. C.1 reports worst-head measurements.

Theorem 1 (Retention of the group-shared selection)**.** *Fix a decode step with J pages in one KV head, $J \geq k \geq 2$, queries $\{\mathbf{q}_g\}_{g=1}^G$, and values bounded by $\|\mathbf{v}_i\| \leq R$. Let S contain the sink and most recent pages and the $k-2$ other pages selected by $\bar{\hat{m}}_j$. For fully unquantized rank-r summaries, let $\eta := \max_{g,j}|\hat{s}_j(\mathbf{q}_g) - s_j(\mathbf{q}_g)|$, with $\hat{s}_j$ defined by Eq. (2).*

(i) The score error satisfies $\eta \leq \max_{g,j}\|\mathbf{q}_{g,j}^\perp\|\,\sigma_{j,r+1}/\sqrt{d}$, where $\mathbf{q}_{g,j}^\perp := (\mathbf{I} - \mathbf{V}_j\mathbf{V}_j^\top)\mathbf{q}_g$ and $\sigma_{j,r+1}$ is page j's first discarded singular value (zero if none remains). In particular, $\eta = 0$ if $r \geq \operatorname{rank}(\mathbf{D}_j)$ for every page.

(ii) The shared set satisfies $\bar{\gamma}(S) := G^{-1}\sum_{g=1}^G\gamma_S(\mathbf{q}_g) \geq e^{-4\eta}\bar{\gamma}^\star$, where $\bar{\gamma}^\star$ is the maximum group-average coverage over shared k-page sets containing the sink and most recent pages.

(iii) Consequently, the group-average output error obeys

$$\frac{1}{G}\sum_{g\leq G}\left\|\mathbf{o}_S(\mathbf{q}_g) - \mathbf{o}(\mathbf{q}_g)\right\| \;\leq\; 2R\big(1 - e^{-4\eta}\,\bar{\gamma}^\star\big).$$

Proof. *App. A.3.*

Part (i) uses each page's top-r SVD basis to bound score error from discarded singular modes. That basis minimizes squared key reconstruction error among rank-r orthonormal bases (Cor. 1). Fig. 5 reports page recall for quantized summaries against exact-LSE selection.

Implementation. LOCKS integrates as an external attention backend in vLLM without modifying the model or the engine's core. Because r, B, and b (hence k) are fixed, the ranking kernel of §4 has fixed shapes too, so the entire batched decode step, selection included, replays inside full CUDA graphs. We analyze its serving cost, including the full-summary scan, in §5.3.

5 EXPERIMENTAL EVALUATION

Existing KV-cache optimization methods often focus on either retrieving evidence from long inputs (Tang et al., 2024; Sun et al., 2025) or sustaining long reasoning traces (Cai et al., 2025; Zhang et al., 2026). We evaluate whether the same LOCKS page-local construction and selection rule serves both: long-input retrieval and QA (§5.1), and long-form reasoning (§5.2). Every quality figure sweeps the fixed per-(layer, KV-head) attended-token budget b of §2 over $b \in \{64, \ldots, 2048\}$ against FullKV, the exact-LSE oracle (§2), the LOCKS family (r8i4, r4i4, r2i4), and a suite of state-of-the-art selectors, evictors, and reasoning compressors named per benchmark. Quality comparisons use identical records and a common FullKV reference; the protocol, model roster, budget accounting, and per-arm sampling settings are in App. B. Separately, we measure LOCKS as a vLLM decode backend alongside FA3, FlashInfer, and a GPU port of Quest (§5.3).

5.1 RETRIEVAL AND LONG-DOCUMENT QA

In retrieval and long-document QA, selection must find a few specific pages from a long prefill. LOCKS tracks exact-LSE selection (§2) down to the smallest budgets and stays within about a point of FullKV on LongBench-v1 (Bai et al., 2024) while attending to only 0.6%–18.6% of its full KV cache (mean $\sim$11K tokens), whereas four state-of-the-art selectors and evictors fall behind as the budget shrinks (Fig. 2), namely Quest (Tang et al., 2024), KVzip (Kim et al., 2025), ShadowKV (Sun et al., 2025), and RocketKV (Behnam et al., 2025). The gap is widest on RULER (Hsieh et al., 2024) at the smallest budgets. Across the plotted sweep, this corresponds to attending to only 0.4%–12.5% of the full RULER-16K cache and 0.2%–6.25% of the full RULER-32K cache. Fig. 1d shows that excluding LOCKS's selected pages sharply lowers S3 and VT accuracy at 32K, even with the rest of the cache attended. Exact-LSE also has limits as a task-accuracy reference: RocketKV can surpass the oracle on NIAH-MK3, while LOCKS and the oracle fail on word-frequency aggregation at b=64 (App. B).

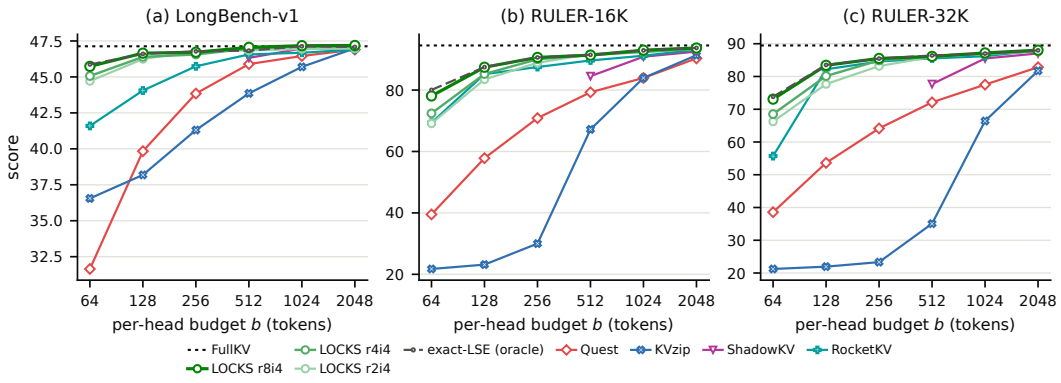

Figure 2: Accuracy as a function of the attended-token budget (Llama-3.1-8B, greedy decoding). Panels show (a) LongBench-v1 (14-subset mean, average context $\sim$11K tokens), (b) RULER-16K, and (c) RULER-32K (13-task means). The x-axis shows a common nominal budget. Actual KV reads and cache residency differ by method (App. B.2.1). ShadowKV begins at b=512 because its 416 always-attended tokens exceed the smaller budgets.

We extend the comparison to 100K+ contexts on InfiniteBench (Zhang et al., 2024). With GLM-4-9B-Chat-1M (Team GLM, 2024), the same ordering holds: LOCKS tracks the exact-LSE oracle and has the highest measured mean among deployable selectors at the tight b=512 and generous b=2048 budgets (Table 2). On a 100K cache, these correspond to attending to about 0.5% and 2% of tokens, respectively, and less on longer records.

Table 2: InfiniteBench scores at 100K+ context (GLM-4-9B-Chat-1M). Retr and LongQA average retrieval and long-document QA tasks; Avg covers all 11 tasks. Entries are means ± 95% bootstrap CI half-widths. Bold marks the leading deployable point estimate; the leading intervals overlap. Per-task results are in Table B.5.

	budget b=512			budget b=2048		
Method	Retr	LongQA	Avg	Retr	LongQA	Avg
FullKV	76.0 ±4.1	35.3 ±3.4	43.0 ±2.1	76.0 ±4.1	35.3 ±3.4	43.0 ±2.1
Oracle (exact-LSE)	74.7 ±3.9	35.1 ±3.7	42.3 ±3.0	76.7 ±4.2	36.2 ±3.7	43.9 ±2.8
LOCKS (r8i4)	**71.3** ±3.2	**33.6** ±3.7	**41.1** ±2.4	**76.0** ±4.1	**36.6** ±3.8	**43.6** ±2.6
RocketKV	64.0 ±5.2	33.6 ±4.1	38.5 ±2.7	75.3 ±4.8	35.0 ±4.2	42.3 ±2.4
ShadowKV	66.0 ±1.3	29.1 ±3.4	36.0 ±2.1	71.3 ±3.2	33.1 ±3.8	40.3 ±2.3
Quest	62.7 ±3.0	26.7 ±3.5	34.7 ±2.7	58.0 ±4.0	31.0 ±3.8	35.7 ±2.6

5.2 LONG-FORM REASONING

Long-form reasoning grows the KV cache over thousands of generated tokens while the relevant past tokens can change. We evaluate this regime on AIME26 (2026 I+II, 30 problems) and MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024) with Qwen3-4B (Yang et al., 2025a) in thinking mode. At the same attended-token budgets, we compare LOCKS with FullKV, the exact-LSE oracle, and four baselines: Quest (Tang et al., 2024), R-KV (Cai et al., 2025), TriAttention (Mao et al., 2026), and LazyEviction (Zhang et al., 2026).

Figure 3 shows a task-dependent budget floor. On AIME26, LOCKS tracks the exact-LSE oracle and reaches FullKV-level accuracy by b=512, while all four baselines remain far below it. Quest still scores 0% there despite being able to reselect past KV at each step, so irreversible eviction alone does not explain the gap. On MATH-500, r8i4 matches FullKV at b=1024, and the baselines trail across the sweep. Even at b=64, all three LOCKS ranks score above 50% on MATH-500, while no baseline reaches 12%. At this budget, even the exact-LSE oracle scores 0% on AIME26 but 56% on MATH-500: AIME26's collapse cannot be attributed to summary approximation alone. At b=2048, the strongest AIME26 baseline still trails FullKV by over 14 points, whereas the MATH-500 baselines have largely recovered. The r4i4 and r2i4 variants often match or exceed r8i4 on both tasks. Full results are in Tables B.6 and B.7.

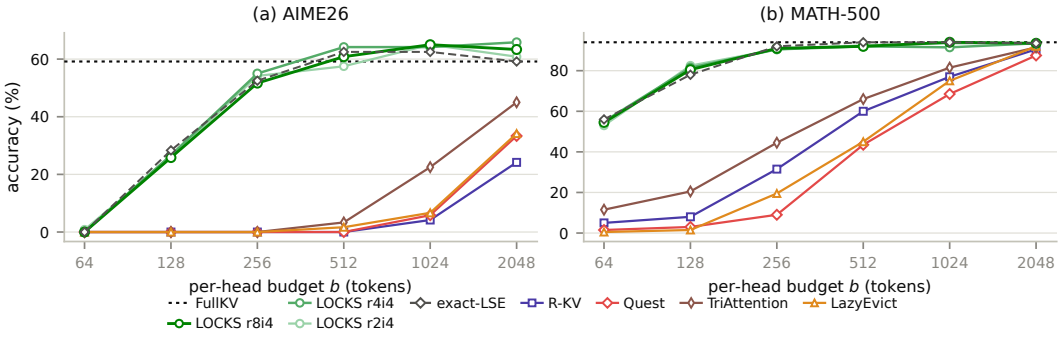

Figure 3: Reasoning accuracy across attended-token budgets with Qwen3-4B (Yang et al., 2025a) (thinking on, temperature 0.6, top-p 0.95, top-k 20, avg@4). Panels show (a) AIME26 (38,912-token generation cap) and (b) MATH-500 (16,384-token generation cap). Further details are in App. B.

5.3 EFFICIENCY

We evaluate LOCKS on GH200 hardware after long prefills, during long generation, and with batched requests. We time complete vLLM decode steps, including page selection, against dense FA3 and FlashInfer and an optimized Quest port (Tang et al., 2024). Table B.8 also reports latencies for Quest-U, a port adapted from the upstream implementation. The other sparse quality baselines were not integrated into this timing setup.

With GPU-resident KV, LOCKS and Quest have comparable first-token and decode latencies on GLM-4-9B-Chat-1M, as shown in Fig. 4(a–c). LOCKS's TTFT includes summary construction yet exceeds FA3's by no more than 6.4% across 16K–512K prefills. At 512K, it decodes in 10.1 ms per step versus FlashInfer's 18.0 ms. Panel (d) compares quality and decode latency on Qwen3-4B: at b=2048, LOCKS scores 65.8% on AIME26 versus Quest's 33.3%, with mean step times of 4.49 and 4.58 ms, respectively.

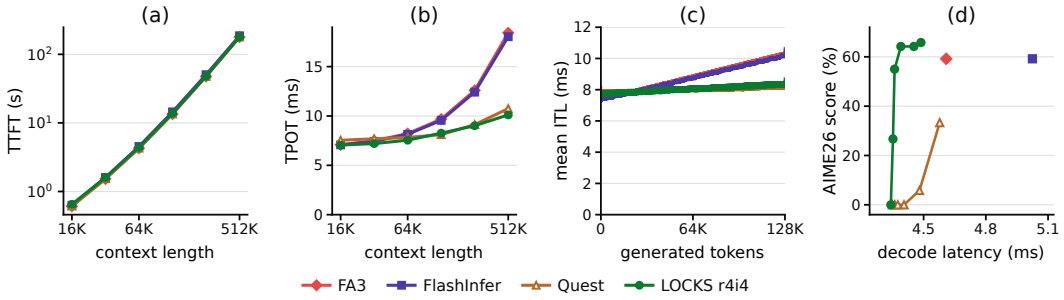

Figure 4: GH200 decode efficiency (batch 1, resident bf16 KV). For GLM-4-9B-Chat-1M (Team GLM, 2024), (a,b) show mean TTFT over three prefills and mean decode time over 99 steps at b=2048; (c) shows 10-token mean decode time during a 128K continuation after a 32K prefill. (d) pairs Qwen3-4B (Yang et al., 2025a) AIME26 avg@4 scores (FullKV for FA3 and FlashInfer) with mean decode time after a 100-token input and 38,912 generated tokens. Markers connect budgets 64–2048 in ascending order.

Serving systems batch requests to increase throughput (Yu et al., 2022; Kwon et al., 2023). With its original KV offloaded to Grace memory, LOCKS uses GPU memory for a hot cache and larger batches. At 64K, 128K, and 256K, Table 3 reports 1,887.7, 1,067.5, and 530.7 tokens/s, or 3.82–4.22× the faster dense backend. Its measured batches (58, 29, 14 versus 12, 6, 3) yield this aggregate gain despite longer mean decode steps.

Table 3: GH200 batched decode (GLM-4-9B-Chat-1M, TP= 1, bf16). Dense KV stays in GPU HBM; LOCKS r4i4 (b=2048) uses Grace memory over coherent NVLink-C2C (Syed et al., 2025) with a GPU hot cache. Hot KV is allocated GPU cache capacity. The "Hot pages" column gives the share of selected page accesses served from that cache over 192 decode steps. Throughput is aggregate generated tokens per second at the shown batch; Speedup is its ratio to the faster dense row at the same context.

Context	Method	Batch	Hot KV (GiB)	Hot pages (%)	Mean ITL (ms)	Throughput (tok/s)	Speedup
64K	FA3	11	N/A	N/A	22.5	489.0	0.99×
	FlashInfer	12	N/A	N/A	24.3	493.9	1.00×
	LOCKS r4i4	58	36.3	90.3	30.7	**1,887.7**	**3.82×**
128K	FA3	6	N/A	N/A	24.2	247.5	0.98×
	FlashInfer	6	N/A	N/A	23.7	253.1	1.00×
	LOCKS r4i4	29	31.7	91.0	27.2	**1,067.5**	**4.22×**
256K	FA3	3	N/A	N/A	24.1	124.3	0.98×
	FlashInfer	3	N/A	N/A	23.6	127.0	1.00×
	LOCKS r4i4	14	21.9	89.2	26.4	**530.7**	**4.18×**

5.4 KV-CACHE QUANTIZATION UNDER EXTREME SPARSITY

Selection reads a few pages per step; KV quantization can shrink their storage but may amplify error under sparse attention. We test KIVI quantization (Liu et al., 2024) with two summary inputs: POST builds from bf16 keys, while PRE builds from quantized keys; both attend to quantized KV (Table 4; full grid in Table B.10). Four-bit KV remains compatible with rank-8 LOCKS across every tested budget, while two-bit KV consistently costs quality, especially at tight budgets.

Table 4: RULER-32K average (13 tasks, Llama-3.1-8B) under 4- and 2-bit KIVI quantization at three budgets. POST builds the summary from bf16 keys; PRE builds it from quantized keys. Both attend to quantized KV. The bf16 LOCKS control matches budget and rank. Full grid: Table B.10.

	r=8					r=4					r=2				
		4-bit		2-bit			4-bit		2-bit			4-bit		2-bit	
b	bf16	POST	PRE	POST	PRE	bf16	POST	PRE	POST	PRE	bf16	POST	PRE	POST	PRE
64	73.1	72.8	73.0	68.0	67.6	68.5	68.6	68.8	64.9	63.7	66.2	65.4	65.9	60.0	60.5
256	85.5	85.2	85.2	81.9	80.7	84.9	83.8	84.4	80.4	79.0	83.2	82.7	82.2	78.2	76.5
2048	88.0	87.6	87.4	85.0	85.3	87.7	87.2	87.6	84.5	84.9	87.6	87.5	87.8	84.8	84.3

5.5 ABLATIONS

We vary rank and basis precision across four page sizes at a fixed attended-token budget (Fig. 5). Int8 nearly tracks bf16, whereas int4 levels off at higher rank and int2 gains little from added rank. At the same rank, larger pages have lower recall, so increasing B calls for a richer basis.

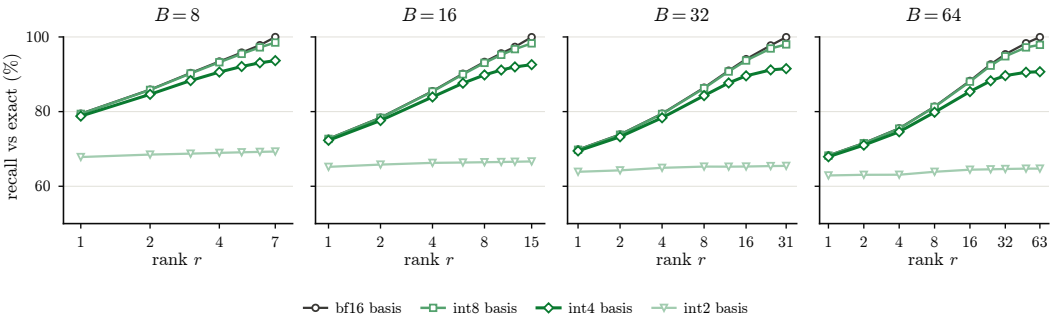

Figure 5: Page recall vs. exact-LSE by rank, basis precision, and page size (Llama-3.1-8B, b=1024).

Larger pages reduce summary bytes per KV token, but leave fewer pages within the fixed budget. Table 5 shows how this tradeoff reaches task scores: LongBench-v1 varies by at most 0.2 points across page sizes, while RULER-32K favors B=16 by 2.3–3.7 points over the larger pages at these budgets. RULER-16K is less sensitive, especially at b=2048. Further GQA ablations and extended model studies appear in App. C.

Table 5: Page-size ablation (Llama-3.1-8B, r8i4); full grid and FullKV in Table C.3.

	LongBench-v1			RULER-16K			RULER-32K		
b	B=16	B=32	B=64	B=16	B=32	B=64	B=16	B=32	B=64
256	46.7	46.8	46.8	90.6	89.7	89.0	85.5	82.7	83.2
1024	47.2	47.1	47.1	93.0	92.1	92.7	87.2	83.5	84.8
2048	47.2	47.1	47.0	93.7	93.0	93.6	88.0	84.5	85.1

6 Related Work

KV eviction. Eviction reduces cache residency by choosing what to keep before future queries arrive. Methods use attention or reconstruction signals (Zhang et al., 2023; Li et al., 2024; Kim et al., 2025; Zweiger et al., 2026), sink-and-window structure (Xiao et al., 2024b), or head-specific budgets and retention (Feng et al., 2025; Yang et al., 2026b; Xiao et al., 2025). RocketKV combines eviction with query-time selection (Behnam et al., 2025). These designs can reduce resident KV, but an evicted page is unavailable to a later query. LOCKS retains access to all pages while reading only selected KV at each decode step.

Sparse attention and retrieval. Query-aware selectors score retained context using page extrema (Tang et al., 2024), centroids or clusters (Hooper et al., 2025; Liu et al., 2025b), and representative keys (Xiao et al., 2024a). Low-rank, latent, or partial-key representations also guide retrieval (Singhania et al., 2024; Sun et al., 2025; Mu et al., 2025; Lee et al., 2024; Li et al., 2025). Other selectors use codebooks or compressed keys (Zhang et al., 2025; Yang et al., 2026a), moment summaries (Tian et al., 2026), or head votes (Wu et al., 2025). LOCKS differs in fitting each page's own basis and reconstructing token-level logits before estimating its log-sum-exp mass. Related systems train landmark-based chunk retrieval (Mohtashami & Jaggi, 2023; Hu et al., 2026), target coverage (Lin et al., 2025; Zhu et al., 2025), reuse rankings (Yang et al., 2025b; Deshmukh et al., 2025), or reduce decode scoring cost (Ribar et al., 2024; Chen et al., 2025; Liu et al., 2025a). Prism scores spectral bands for prefill (Wang et al., 2026b), while MInference targets sparse prefill (Jiang et al., 2024).

Selection objectives. The exact top-k truncation identity also appears in Tzachristas et al. (2025) and Tian et al. (2026); the latter derives a value-agnostic mass criterion. Expected Attention, value-aware scoring, verified sampling, and softmax thresholding offer other ways to decide what to attend (Devoto et al., 2025; Guo et al., 2024; Desai et al., 2026; Yuan et al., 2026). Exact top-k acceleration (Cheng et al., 2026) and certified quantized attention (Calver, 2026) address the reliability or cost of different selection mechanisms. Our target is the exact-LSE ranking implied by value-blind output-error control (§2).

KV quantization and compression. Quantizers shrink stored keys and values (Liu et al., 2024; Hooper et al., 2024; Hong et al., 2025; He et al., 2024; Yang et al., 2024). MQA and GQA share KV across query heads (Shazeer, 2019; Ainslie et al., 2023), while low-rank schemes reduce each cached entry (Chang et al., 2025; Saxena et al., 2024; Zhu et al., 2026; Lesens et al., 2025). Storage compression reduces cache footprint, while page selection reduces the KV read per query. §5.4 tests LOCKS with KIVI on accuracy.

Trained sparsity and long reasoning. Native sparse and hybrid attention change the model's attention architecture (Yuan et al., 2025; Lu et al., 2025; DeepSeek-AI, 2026; Z.ai, 2026; Qwen Team, 2026). SeerAttention distills a gate (Gao et al., 2024), and SPLA learns a selector inside sparse-plus-linear attention (Wang et al., 2026a). Long-reasoning methods instead allocate or evict generated history by redundancy, hierarchy, values, or mass regions (Cai et al., 2025; Liu et al., 2026; Su et al., 2026; Chang et al., 2026; Yang & Shen, 2026). LOCKS uses one training-free page selector across long-input and long-reasoning tasks. Sparse-attention tradeoffs across tasks are also compared by Nawrot et al. (2026).

7 Conclusion

Attention keys admit compact low-rank approximations within pages. LOCKS uses that locality to approximate exact page LSE, the value-blind selection target linked to worst-case attention-output error. At tight budgets, its training-free summaries track exact-LSE selection on retrieval and QA while preserving long-form reasoning accuracy. On one GH200 with GPU-resident KV, LOCKS r4i4 cuts complete decode-step time by $1.8\times$ at 512K context. On Qwen3-4B AIME26, it also offers a better quality–latency tradeoff than Quest at similar decode times (Fig. 4(d)). With full KV offloaded to Grace memory, its GPU hot cache admits larger batches and yields 3.82–$4.22\times$ the faster dense backend's aggregate throughput at 64K–256K context. These results identify page-specific geometry as a useful unit of representation for sparse decode. Adaptive page ranks and cheaper summary scans remain open.

Reproducibility Statement

The summary construction, page scoring, and selection rule are specified in §4, with proofs in App. A. The evaluation protocols, model and baseline settings, budget accounting, hardware, sampling seeds, and per-task results are reported in §5 and App. B. Further implementation choices and ablations are documented in App. C.

AI Use Statement

Generative AI tools assisted with literature search and citation checks, manuscript drafting and editing, LaTeX formatting, figure preparation and plotting code, feedback on experimental design and protocols, review of mathematical claims and proofs, and kernel optimizations. The authors reviewed these suggestions, including the citations, calculations, proofs, and experimental reporting, and revised them where needed. The authors take responsibility for the final text, claims, code, figures, and other artifacts.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4895–4901, 2023. arXiv:2305.13245.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.172.

Baidu ERNIE Team. ERNIE 4.5 technical report. `https://ernie.baidu.com/blog/publication/ERNIE_Technical_Report.pdf`, 2025.

Payman Behnam, Yaosheng Fu, Ritchie Zhao, Po-An Tsai, Zhiding Yu, and Alexey Tumanov. RocketKV: Accelerating long-context LLM inference via two-stage KV cache compression. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pp. 3358–3392. PMLR, 2025. arXiv:2502.14051.

Zefan Cai, Wen Xiao, Hanshi Sun, Cheng Luo, Yikai Zhang, Ke Wan, Yucheng Li, Yeyang Zhou, Li-Wen Chang, Jiuxiang Gu, Zhen Dong, Anima Anandkumar, Abedelkadir Asi, and Junjie Hu. R-KV: Redundancy-aware KV cache compression for reasoning models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2505.24133.

Dean Calver. Runtime-certified bounded-error quantized attention, 2026. arXiv:2605.20868.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, et al. Palu: KV-cache compression with low-rank projection. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2407.21118.

Ting-Yun Chang, Harvey Yiyun Fu, Deqing Fu, Chenghao Yang, Jesse Thomason, and Robin Jia. Value-aware stochastic KV cache eviction for reasoning models, 2026. arXiv:2606.03928.

Zhuoming Chen, Ranajoy Sadhukhan, Zihao Ye, et al. MagicPIG: LSH sampling for efficient LLM generation. In *International Conference on Learning Representations (ICLR)*, 2025. Spotlight; OpenReview ALzTQUgW8a.

Long Cheng, Ritchie Zhao, Timmy Liu, Mindy Li, Xianjie Qiao, Kefeng Duan, Yu-Jung Chen, Xiaoming Chen, Bita Darvish Rouhani, and June Yang. Guess-verify-refine: Data-aware top-K for sparse-attention decoding on Blackwell via temporal correlation, 2026. arXiv:2604.22312.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Aditya Desai, Kumar Agrawal, Shuo Yang, Alejandro Cuadron Lafuente, Luis Gaspar Schroeder, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. vAttention: Verified sparse attention via sampling. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.05688.

Dhruv Deshmukh, Saurabh Goyal, Nipun Kwatra, and Ramachandran Ramjee. Kascade: A practical sparse attention method for long-context LLM inference, 2025. arXiv:2512.16391.

Alessio Devoto, Maximilian Jeblick, and Simon Jégou. Expected attention: KV cache compression by estimating attention from future queries distribution, 2025. arXiv:2510.00636.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S. Kevin Zhou. Ada-KV: Optimizing KV cache eviction by adaptive budget allocation for efficient LLM inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. OpenReview tcisuhGsQZ; arXiv:2407.11550.

Yizhao Gao, Zhichen Zeng, Dayou Du, et al. SeerAttention: Learning intrinsic sparse attention in your LLMs. *arXiv preprint arXiv:2410.13276*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Zhiyu Guo, Hidetaka Kamigaito, and Taro Watanabe. Attention score is not all you need for token importance indicator in KV cache reduction: Value also matters. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. arXiv:2406.12335.

Yefei He, Luoming Zhang, Weijia Wu, Jing Liu, Hong Zhou, and Bohan Zhuang. ZipCache: Accurate and efficient KV cache quantization with salient token identification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2405.14256.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Yankun Hong, Xing Li, Hui-Ling Zhen, Xianzhi Yu, Wulong Liu, and Mingxuan Yuan. SVDq: 1.25-bit and 410x key cache compression for LLM attention. *arXiv preprint arXiv:2502.15304*, 2025.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. In *NeurIPS*, 2024.

Coleman Hooper, Sebastian Zhao, Luca Manolache, et al. Multipole attention for efficient long context reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2506.13059.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *Conference on Language Modeling (COLM)*, 2024. arXiv:2404.06654.

Xiang Hu, Xinyu Wei, Hao Gu, et al. Hierarchical sparse attention done right: Toward infinite context modeling. *arXiv preprint arXiv:2607.02980*, 2026.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, et al. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *NeurIPS*, 2024.

Jang-Hyun Kim, Jinuk Kim, Sangwoo Kwon, Jae W. Lee, Sangdoo Yun, and Hyun Oh Song. KVzip: Query-agnostic KV cache compression with context reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. Oral; OpenReview JFygzwx8SJ; arXiv:2505.23416.

Woosuk Kwon et al. Efficient memory management for large language model serving with pagedattention. In *SOSP*, 2023.

Wonbeom Lee, Jungi Lee, Junghwan Seo, and Jaewoong Sim. InfiniGen: Efficient generative inference of large language models with dynamic KV cache management. In *OSDI*, 2024.

Damien Lesens, Beheshteh T. Rakhshan, and Guillaume Rabusseau. KQ-SVD: Compressing the KV cache with provable guarantees on attention fidelity, 2025. arXiv:2512.05916.

Tenghui Li, Guoxu Zhou, Xuyang Zhao, Yuning Qiu, and Qibin Zhao. Efficient low rank attention for long-context inference in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2510.23649.

Yuhong Li et al. Snapkv: LLM knows what you are looking for before generation. In *NeurIPS*, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

Chaofan Lin, Jiaming Tang, Shuo Yang, Hanshuo Wang, Tian Tang, Boyu Tian, Ion Stoica, Song Han, and Mingyu Gao. Twilight: Adaptive attention sparsity with hierarchical top-p pruning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. Spotlight; OpenReview Ve693NkzcU; arXiv:2502.02770.

Di Liu, Meng Chen, Baotong Lu, et al. RetrievalAttention: Accelerating long-context LLM inference via vector retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a.

Guangda Liu, Chengwei Li, Jieru Zhao, Chenqi Zhang, and Minyi Guo. ClusterKV: Manipulating LLM KV cache in semantic space for recallable compression. In *Proceedings of the 62nd ACM/IEEE Design Automation Conference (DAC)*, 2025b. arXiv:2412.03213.

Wenhao Liu, Hao Shi, Yunhe Li, Weizhi Fei, Xiangyuan Wang, Mengzhe Ruan, Hanxu Hou, Peisong Wang, Linqi Song, and Shuang Qiu. ReasonAlloc: Hierarchical decoding-time KV cache budget allocation for reasoning models, 2026. arXiv:2606.11164.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *ICML*, 2024.

Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, Zhiqi Huang, Huan Yuan, Suting Xu, Xinran Xu, Guokun Lai, Yanru Chen, Huabin Zheng, Junjie Yan, Jianlin Su, Yuxin Wu, Neo Y. Zhang, Zhilin Yang, Xinyu Zhou, Mingxing Zhang, and Jiezhong Qiu. MoBA: Mixture of block attention for long-context LLMs, 2025. arXiv:2502.13189.

Weian Mao, Xi Lin, Wei Huang, Yuxin Xie, Tianfu Fu, Bohan Zhuang, Song Han, and Yukang Chen. TriAttention: Efficient long reasoning with trigonometric KV compression. *arXiv preprint arXiv:2604.04921*, 2026.

Mathematical Association of America. American invitational mathematics examination (AIME) 2026. `https://maa.org/news/2025-26-aime-thresholds-are-now-available/`, 2026.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Amirkeivan Mohtashami and Martin Jaggi. Random-access infinite context length for transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. Landmark Attention; arXiv:2305.16300.

Junlin Mu, Hantao Huang, Jihang Zhang, Minghui Yu, Tao Wang, and Yidong Li. SALS: Sparse attention in latent space for KV cache compression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2510.24273.

Piotr Nawrot et al. The sparse frontier: Sparse attention trade-offs in transformer LLMs. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 38667–38701, 2026. arXiv:2504.17768.

Zihan Qiu, Zekun Wang, Xiao Li, et al. On the design of Qwen3.8-Next architecture: Evaluation, efficiency, and training stability. *arXiv preprint arXiv:2608.30320*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents. `https://qwen.ai/blog?id=qwen3.5`, 2026.

Luka Ribar, Ivan Chelombiev, Luke Hudlass-Galley, Charlie Blake, Carlo Luschi, and Douglas Orr. SparQ attention: Bandwidth-efficient LLM inference. In *ICML*, 2024.

Utkarsh Saxena, Gobinda Saha, Sakshi Choudhary, and Kaushik Roy. Eigen attention: Attention in low-rank space for KV cache compression. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15332–15344, 2024.

Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

Prajwal Singhania, Siddharth Singh, Shwai He, Soheil Feizi, and Abhinav Bhatele. Loki: Low-rank keys for efficient sparse attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2406.02542.

Yi Su, Zhenxu Tian, Dan Qiao, Yuechi Zhou, Juntao Li, and Min Zhang. LongFlow: Efficient KV cache compression for reasoning models, 2026. arXiv:2603.11504.

Hanshi Sun, Li-Wen Chang, Wenlei Bao, et al. ShadowKV: KV cache in shadows for high-throughput long-context LLM inference. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*. PMLR, 2025. Spotlight.

Afroze Syed, Ivan Goldwasser, and Jason Tichy. Accelerate large-scale LLM inference and KV cache offload with CPU-GPU memory sharing. NVIDIA Technical Blog, 2025.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context LLM inference. In *ICML*, 2024.

Team GLM. ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

Alexander Tian, Aditya Ghai, Sanjit Neelam, Zaal Vasania, and Akshay Mishra. COBS: Cumulant order block sparse attention. *arXiv preprint arXiv:2607.09052*, 2026.

Georgios Tzachristas, Lei Deng, Ioannis Tzachristas, Gong Zhang, and Renhai Chen. A mathematical theory of top-k sparse attention via total variation distance, 2025. arXiv:2512.07647.

Bailin Wang, Dan Friedman, Tao Lei, and Chong Wang. SPLA: Block sparse plus linear attention for long context modeling. *arXiv preprint arXiv:2601.22379*, 2026a.

Xinghao Wang, Pengyu Wang, Xiaoran Liu, Fangxu Liu, Jason Chu, Kai Song, and Xipeng Qiu. Prism: Spectral-aware block-sparse attention. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026b. arXiv:2602.08426.

Wei Wu, Zhuoshi Pan, Kun Fu, Chao Wang, Liyi Chen, Yunchu Bai, Tianfu Wang, Zheng Wang, and Hui Xiong. TokenSelect: Efficient long-context inference and length extrapolation for LLMs via dynamic token-level KV cache selection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025. arXiv:2411.02886.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. InfLLM: Training-free long-context extrapolation for LLMs with an efficient context memory. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a. arXiv:2402.04617.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *ICLR*, 2024b.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. DuoAttention: Efficient long-context LLM inference with retrieval and streaming heads. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.10819.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

June Yong Yang, Byeongwook Kim, Jeongin Bae, et al. No token left behind: Reliable KV cache compression via importance-aware mixed precision quantization. *arXiv preprint arXiv:2402.18096*, 2024.

Junzhe Yang and Xiaoyu Shen. Adaptive mass-segmented KV compression for long-context reasoning. *arXiv preprint arXiv:2605.23200*, 2026.

Lijie Yang, Zhihao Zhang, Zhuofu Chen, Zikun Li, and Zhihao Jia. TidalDecode: Fast and accurate LLM decoding with position persistent sparse attention. In *International Conference on Learning Representations (ICLR)*, 2025b.

Xu Yang, Jiapeng Zhang, Dongyang Zhao, Guo Chen, and Zhuo Tang. Self-indexing KVCache: Predicting sparse attention from compressed keys. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 27675–27683, 2026a. arXiv:2603.14224.

Yuxuan Yang, Feiyang Ren, Bowen Zeng, Dalin Zhang, Jinpeng Chen, Gang Chen, and Huan Li. HARD-KV: Head-adaptive regularization for decoding-time KV compression. *arXiv preprint arXiv:2606.28831*, 2026b.

Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun. Orca: A distributed serving system for Transformer-Based generative models. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pp. 521–538, 2022.

Jiayi Yuan, Cameron Shinn, Kai Xu, Jingze Cui, George Klimiashvili, Guangxuan Xiao, Perkz Zheng, Bo Li, et al. BLASST: Dynamic BLocked attention sparsity via softmax thresholding. In *Proceedings of Machine Learning and Systems (MLSys)*, 2026. Best Paper Award; arXiv:2512.12087.

Jingyang Yuan, Huazuo Gao, Damai Dai, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23078–23097, 2025. Best Paper Award; arXiv:2502.11089.

Z.ai. GLM-5.2: Built for long-horizon tasks. `https://huggingface.co/blog/zai-org/glm-52-blog`, 2026. Open-weight model release, June 2026.

Hailin Zhang, Xiaodong Ji, Yilin Chen, et al. PQCache: Product quantization-based KVCache for long context LLM inference. *Proceedings of the ACM on Management of Data*, 3(3):201:1–201:30, 2025. arXiv:2407.12820.

Haoyue Zhang, Hualei Zhang, Xiaosong Ma, Jie Zhang, and Song Guo. LazyEviction: Lagged KV eviction with attention pattern observation for efficient long reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36335–36352, 2026. arXiv:2506.15969.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞bench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15262–15277, 2024. arXiv:2402.13718.

Zhenyu Zhang et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *NeurIPS*, 2023.

Kan Zhu, Tian Tang, Qinyu Xu, et al. Tactic: Adaptive sparse attention with clustering and distribution fitting for long-context LLMs. *arXiv preprint arXiv:2502.12216*, 2025.

Yuxuan Zhu, David H. Yang, Mohammad Mohammadi Amiri, Keerthiram Murugesan, Tejaswini Pedapati, and Pin-Yu Chen. OjaKV: Context-aware online low-rank KV cache compression. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026. arXiv:2509.21623.

Adam Zweiger, Xinghong Fu, Han Guo, and Yoon Kim. Fast KV compaction via attention matching. *arXiv preprint arXiv:2602.16284*, 2026. Accepted as a poster at ICML 2026.

Appendix

Table of Contents

A Proofs 17

A.1 Coverage and minimax selection *17*

A.2 Shared projections and page-local bases *17*

A.3 Retention guarantee *18*

A.4 Group-shared selection *19*

A.5 Moment approximation and outlier capture *20*

B Detailed Results 21

B.1 Representation scope *21*

B.2 Retrieval and long-document QA *22*

B.2.1 Baselines 22

B.2.2 LongBench-v1 22

B.2.3 RULER-16K 23

B.2.4 RULER-32K 25

B.2.5 InfiniteBench 26

B.3 Long-form reasoning *28*

B.3.1 Baselines 28

B.3.2 AIME26 28

B.3.3 MATH-500 28

B.4 GH200 decode efficiency *29*

B.4.1 GPU-resident batch-1 decode 29

B.4.2 Host-backed batched decode 29

B.5 KV-cache quantization (KIVI): full grid *30*

C Extended Studies 31

C.1 GQA *31*

C.2 Rotary Position Embedding *31*

C.3 Page size *31*

C.4 Mixture-of-experts: ERNIE-4.5-21B-A3B-Thinking *32*

C.5 Large dense: DeepSeek-R1-Distill-Qwen-32B *32*

C.6 Learned sparse attention: Qwen3.8-Flash-Next *32*

C.7 H200 decode efficiency: measurement protocol and full numbers *33*

A PROOFS

A.1 COVERAGE AND MINIMAX SELECTION

Lemma 1 (Coverage and deterministic value-blind selection)**.** *Let $\|\mathbf{v}_i\| \le R$ for every cached value.*

(i) For any nonempty page set S and query $\mathbf{q}$, let $\varepsilon = 1-\gamma_S(\mathbf{q})$. If S^c is nonempty, the mass-weighted mean values of the kept and dropped tokens satisfy

$$\mathbf{o}(\mathbf{q}) - \mathbf{o}_S(\mathbf{q}) = \varepsilon\big(\bar{\mathbf{v}}_{S^c} - \bar{\mathbf{v}}_S\big).$$

In all cases, $\|\mathbf{o}_S(\mathbf{q}) - \mathbf{o}(\mathbf{q})\| \le 2R\varepsilon$. When S^c is empty, $\mathbf{o}_S = \mathbf{o}$.

(ii) At a budget of k pages, the top-k pages by the exact LSE score of Eq. (1) *maximize coverage and minimize the bound in (i). They are minimax-optimal for worst-case output error among deterministic value-blind selection rules, with ties broken arbitrarily.*

Throughout, $i \in S$ means $i \in \bigcup_{j\in S} P_j$ for a page set S. For $i \in P_j$, write $\boldsymbol{\delta}_i = \mathbf{k}_i - \boldsymbol{\mu}_j$.

Proof of Lemma 1(i). Let $p_i = \alpha_i$ be the softmax weights over all tokens, $Z_S = \sum_{i\in S} p_i = \gamma_S$. If S^c is empty, $Z_S = 1$ and $\mathbf{o}_S = \mathbf{o}$, so the bound holds. Otherwise, define $\bar{\mathbf{v}}_S = \frac{1}{Z_S}\sum_{i\in S} p_i\mathbf{v}_i$, $\bar{\mathbf{v}}_{S^c} = \frac{1}{1-Z_S}\sum_{i\notin S} p_i\mathbf{v}_i$. The renormalized restricted output is $\mathbf{o}_S = \bar{\mathbf{v}}_S$, while $\mathbf{o} = Z_S\bar{\mathbf{v}}_S + (1-Z_S)\bar{\mathbf{v}}_{S^c}$. Subtracting, $\mathbf{o} - \mathbf{o}_S = (1-Z_S)(\bar{\mathbf{v}}_{S^c} - \bar{\mathbf{v}}_S) = \varepsilon(\bar{\mathbf{v}}_{S^c} - \bar{\mathbf{v}}_S)$. The mass-weighted means are convex combinations of the $\mathbf{v}_i$, so $\|\bar{\mathbf{v}}_{S^c}\|, \|\bar{\mathbf{v}}_S\| \le R$ and $\|\mathbf{o} - \mathbf{o}_S\| \le 2R\varepsilon = 2R(1-\gamma_S)$. □

Definition 1 (Deterministic value-blind selection rule)**.** Fix a budget k. A selection rule is a map from the keys and query, $(\{\mathbf{k}_i\}, \mathbf{q})$, to a k-subset of pages. It is *deterministic value-blind* if its output does not depend on the cached values $\{\mathbf{v}_i\}$: two problem instances with identical keys and query but different values receive the same selected set.

Proof of Lemma 1(ii). Coverage $\gamma_S = \sum_{j\in S} m_j$ with page-mass share $m_j = e^{s_j}/\sum_{j'} e^{s_{j'}}$. Maximizing an additive set function over k-subsets is attained by the k largest m_j, i.e. the top-k by s_j; with mandatory pages fixed, take the largest remaining masses in the free slots. This minimizes $\varepsilon = 1-\gamma_S$ and the bound of part (i). If k equals the number of pages, the unique set has zero error. Otherwise, for minimax optimality among the rules of Definition 1, fix the keys and the query, so each candidate set S determines its dropped mass ε_S, independent of any rule's knowledge of $\{\mathbf{v}_i\}$. A deterministic value-blind rule picks S without seeing $\{\mathbf{v}_i\}$, and the exact identity of part (i) gives error $\varepsilon_S\|\bar{\mathbf{v}}_{S^c} - \bar{\mathbf{v}}_S\|$. Over assignments with $\max_i \|\mathbf{v}_i\| \le R$ this never exceeds $2R\varepsilon_S$, and the assignment $\mathbf{v}_i = R\mathbf{u}$ on S^c, $\mathbf{v}_i = -R\mathbf{u}$ on S (any unit $\mathbf{u}$) attains it, so any fixed deterministic value-blind rule's worst-case error at the set it picks is exactly $2R\varepsilon_S$. It is minimized by minimizing ε_S via the mass top-k. □

Randomized selectors. For three equal-mass single-token pages and $k = 1$, any deterministic choice has worst-case error $4R/3$. Uniform random selection has expected error $\frac{1}{3}\sum_i \|\mathbf{v}_i - \bar{\mathbf{v}}\| \le R$, since $\sum_i \|\mathbf{v}_i - \bar{\mathbf{v}}\|^2 = \sum_i \|\mathbf{v}_i\|^2 - 3\|\bar{\mathbf{v}}\|^2 \le 3R^2$. Thus the minimax statement in Lemma 1 requires deterministic selection.

A.2 SHARED PROJECTIONS AND PAGE-LOCAL BASES

Lemma 2 (Envelope content-blindness)**.** *The envelope score of Tang et al. (2024), divided by $\sqrt{d}$, upper-bounds every within-page logit, but depends on a page only through its per-dimension key extremes: pages with equal extremes receive identical scores for every query, and for $d \ge 2$ and $B \ge 2$ there are equal-extreme page pairs whose exact log-masses separate without bound.*

Proof of Lemma 2. Let $u_{j,c} = \max_{i\in P_j} k_{i,c}$ and $\ell_{j,c} = \min_{i\in P_j} k_{i,c}$. The envelope score is

$$e_j(\mathbf{q}) = \sum_{c=1}^{d} \max\big(q_c u_{j,c},\, q_c \ell_{j,c}\big).$$

For any $i \in P_j$, $k_{i,c} \in [\ell_{j,c}, u_{j,c}]$ implies $q_c k_{i,c} \le \max(q_c u_{j,c}, q_c \ell_{j,c})$ in every coordinate. Summing and dividing by $\sqrt{d}$ upper-bounds each token logit. Equal extremes give equal envelope scores for every query.

For separation, take $t > 0$. Let P' contain $(t, t, 0, \dots)$ and $(-t, -t, 0, \dots)$, and let P contain $(t, -t, 0, \dots)$ and $(-t, t, 0, \dots)$; pad both to B keys with zeros. Their extremes agree, but for $\mathbf{q} = (\mathbf{e}_1 + \mathbf{e}_2)/\sqrt{2}$ every key of P has zero logit, whereas P' has one with logit $t\sqrt{2}/\sqrt{d}$. Thus $s_{P'}(\mathbf{q}) - s_P(\mathbf{q}) \ge t\sqrt{2}/\sqrt{d} - \log B \to \infty$. □

Proof of Proposition 1. Choose a unit $\mathbf{u} \perp \mathrm{span}(\mathbf{W})$. Let P contain B zero keys, and let P_c replace two of them by $c\,\mathbf{u}$ and $-c\,\mathbf{u}$. Both centroids and all projected keys are zero, so any scorer of the query, centroid, and projected keys gives the pages the same score. For $\mathbf{q} = \mathbf{u}$, $s_P(\mathbf{q}) = \log B$ while $s_{P_c}(\mathbf{q}) = \log(B - 2 + 2\cosh(c/\sqrt{d})) \to \infty$. The centered keys of P_c lie in $\mathrm{span}(\mathbf{u})$; a page-specific rank-one basis containing $\mathbf{u}$ reconstructs them exactly. The zero page is exact with any basis. □

Corollary 1 (The per-page basis is residual-optimal)**.** *Fix a page j and rank $r < d$. Among summaries $(\boldsymbol{\mu}_j, \mathbf{H}, \{\mathbf{H}^\top \boldsymbol{\delta}_i\})$ over orthonormal $\mathbf{H} \in \mathbb{R}^{d\times r}$, the spectral basis $\mathbf{V}_j$ minimizes the squared reconstruction residual $\tau_{j,r}(\mathbf{H}) = \sum_{i\in P_j} \|\boldsymbol{\delta}_i - \mathbf{H}\mathbf{H}^\top \boldsymbol{\delta}_i\|^2$. When $\sigma_{j,r} > \sigma_{j,r+1}$ the minimizing* subspace $\mathrm{span}(\mathbf{H}) = \mathrm{span}(\mathbf{V}_j)$ *is unique. Every orthonormal basis of that subspace attains the same residual and Frobenius-tail bound of Prop. 2.*

Proof of Corollary 1. For any orthonormal $\mathbf{H} \in \mathbb{R}^{d\times r}$,

$$\tau_{j,r}(\mathbf{H}) = \|\mathbf{D}_j - \mathbf{D}_j \mathbf{H}\mathbf{H}^\top\|_F^2 \ge \sum_{\ell > r} \sigma_{j,\ell}^2.$$

The Eckart–Young theorem (Eckart & Young, 1936) gives this rank-r minimum, and $\mathbf{H} = \mathbf{V}_j$ attains it. A strict gap $\sigma_{j,r} > \sigma_{j,r+1}$ makes the leading r-dimensional eigenspace of $\mathbf{D}_j^\top \mathbf{D}_j$ the unique minimizing subspace. Rotating an orthonormal basis within it leaves the projector $\mathbf{H}\mathbf{H}^\top$, and hence the residual, unchanged.

For the Frobenius-tail bound, set $\mathbf{q}_{\perp,\mathbf{H}} := (\mathbf{I} - \mathbf{H}\mathbf{H}^\top)\mathbf{q}$. Since $r < d$,

$$\sup_{\|\mathbf{q}\|\le 1} \frac{\|\mathbf{q}_{\perp,\mathbf{H}}\|\sqrt{\tau_{j,r}(\mathbf{H})}}{\sqrt{d}} = \frac{\sqrt{\tau_{j,r}(\mathbf{H})}}{\sqrt{d}}.$$

Thus minimizing the residual also minimizes this worst-case bound. □

A.3 Retention guarantee

Step 1: per-query, per-page score error.

Proposition 2 (Score error characterization)**.** *Fix $\mathbf{q}$ and a page j; let $\hat{s}_j$ be the score of Eq.* (2) *with fully unquantized summaries, $\boldsymbol{\delta}_i^\perp = \boldsymbol{\delta}_i - \mathbf{V}_j \mathbf{c}_i$ the reconstruction residuals, and $\mathbf{q}_\perp = \mathbf{q} - \mathbf{V}_j \mathbf{V}_j^\top \mathbf{q}$. Then*

$$\left|\hat{s}_j(\mathbf{q}) - s_j(\mathbf{q})\right| \;\le\; \max_{i\in P_j} \frac{\left|\mathbf{q}_\perp^\top \boldsymbol{\delta}_i^\perp\right|}{\sqrt{d}} \;\le\; \frac{\|\mathbf{q}_\perp\|\,\sigma_{j,r+1}}{\sqrt{d}} \;\le\; \frac{\|\mathbf{q}_\perp\|\,\sqrt{\tau_{j,r}}}{\sqrt{d}},$$

and $\hat{s}_j = s_j$ whenever $r \ge \mathrm{rank}(\mathbf{D}_j)$.

Proof. For each $i \in P_j$ the reconstructed logit differs from the true one by

$$\frac{\mathbf{q}^\top(\mathbf{V}_j \mathbf{c}_i - \boldsymbol{\delta}_i)}{\sqrt{d}} = -\frac{\mathbf{q}_\perp^\top \boldsymbol{\delta}_i^\perp}{\sqrt{d}},$$

using $\mathbf{V}_j^\top \boldsymbol{\delta}_i^\perp = \mathbf{0}$. Log-sum-exp is 1-Lipschitz in the sup norm of its arguments (its gradient is a softmax, hence a convex combination), which gives the first inequality. For the second, $(\boldsymbol{\delta}_i^\perp)^\top$ is the i-th row of $\mathbf{D}_j(\mathbf{I} - \mathbf{V}_j\mathbf{V}_j^\top)$, so $\|\boldsymbol{\delta}_i^\perp\| \le \|\mathbf{D}_j(\mathbf{I} - \mathbf{V}_j\mathbf{V}_j^\top)\|_2 = \sigma_{j,r+1}$ by optimality of the truncated SVD in spectral norm, and Cauchy–Schwarz completes it. The weaker third form follows from

$$\|\boldsymbol{\delta}_i^\perp\|^2 \le \sum_{i'\in P_j} \|\boldsymbol{\delta}_{i'}^\perp\|^2 = \sum_{\ell > r} \sigma_{j,\ell}^2 = \tau_{j,r}.$$

If $r \geq \mathrm{rank}(\mathbf{D}_j)$, every residual vanishes and $\hat{s}_j = s_j$. □

Taking the maximum over queries and pages proves Theorem 1(i).

Step 2: the group-share sandwich. Write $\eta = \max_{g \leq G} \max_j |\hat{s}_j(\mathbf{q}_g) - s_j(\mathbf{q}_g)|$ as above. Termwise, $e^{\hat{s}_j(\mathbf{q}_g)} \in [e^{-\eta}, e^{\eta}] \cdot e^{s_j(\mathbf{q}_g)}$ for every page j and every query $\mathbf{q}_g$ in the group; normalizing (which multiplies numerator and denominator by factors in the same range) doubles the exponent, giving $\hat{m}_g(j) \in [e^{-2\eta}, e^{2\eta}] \cdot m_g(j)$ for every query in the group. Averaging a family of ratios each sandwiched in $[e^{-2\eta}, e^{2\eta}]$ preserves the sandwich on the average, $\bar{\hat{m}}_j = \frac{1}{G}\sum_g \hat{m}_g(j) \in [e^{-2\eta}, e^{2\eta}] \cdot \frac{1}{G}\sum_g m_g(j) = [e^{-2\eta}, e^{2\eta}] \cdot \bar{m}_j$, i.e.

$$\left|\log \bar{\hat{m}}_j - \log \bar{m}_j\right| \leq 2\eta \qquad \text{for every page } j.$$

Step 3: robustness to bounded score error.

Corollary 2 (Selection is robust to any bounded score error)**.** *Let $w_j > 0$ be a weight for page j, and let g_j be ranking scores satisfying $\max_j |g_j - \log w_j - \zeta| \leq \Delta$ for some page-independent offset ζ. For $J \geq k$, fix a mandatory page set F with $|F| \leq k$. Let S_g and $S^\star$ both contain F and fill their remaining slots with the largest g_j and w_j, respectively, among pages outside F. Then $\sum_{j \in S_g} w_j \geq e^{-2\Delta} \sum_{j \in S^\star} w_j$.*

Proof. Subtracting ζ from every g_j changes no selected set, so take $\zeta = 0$. Since S_g maximizes $\sum_{p \in S} e^{g_p}$ among k-page sets containing F,

$$\sum_{p \in S_g} w_p \;\geq\; e^{-\Delta} \sum_{p \in S_g} e^{g_p} \;\geq\; e^{-\Delta} \sum_{p \in S^\star} e^{g_p} \;\geq\; e^{-2\Delta} \sum_{p \in S^\star} w_p,$$

where the outer inequalities hold termwise from $|g_j - \log w_j| \leq \Delta$. Dividing by $\sum_j w_j$ gives the same ratio for normalized weights. □

Apply Cor. 2 with $w_j = \bar{m}_j$, $g_j = \log \bar{\hat{m}}_j$, F equal to the sink and most recent pages, and $\Delta = 2\eta$ from Step 2. For any shared set S, $\sum_{j \in S} \bar{m}_j = \frac{1}{G}\sum_g \gamma_S(\mathbf{q}_g) = \bar{\gamma}(S)$. Cor. 3 identifies the exact-share maximizer $S^\star$ within the same family of sets, so $\bar{\gamma}(S_g) \geq e^{-4\eta}\bar{\gamma}(S^\star) = e^{-4\eta}\bar{\gamma}^\star$, proving Theorem 1(ii).

Step 4: from coverage to output error. Lemma 1(i) applies to every query in the group against the one shared selection S. Every query $\mathbf{q}_g$ reads the same KV head, hence the same cached values and the same bound R, so $\|\mathbf{o}_S(\mathbf{q}_g) - \mathbf{o}(\mathbf{q}_g)\| \leq 2R(1 - \gamma_S(\mathbf{q}_g))$ for every g. Averaging over the group,

$$\frac{1}{G}\sum_{g \leq G} \left\|\mathbf{o}_S(\mathbf{q}_g) - \mathbf{o}(\mathbf{q}_g)\right\| \;\leq\; 2R\Big(1 - \frac{1}{G}\sum_{g \leq G} \gamma_S(\mathbf{q}_g)\Big) \;=\; 2R\big(1 - \bar{\gamma}(S)\big).$$

Substituting the Step 3 bound gives Theorem 1(iii). □

A.4 GROUP-SHARED SELECTION

Corollary 3 (Group-shared selection)**.** *Let $m_g(j)$ be query $\mathbf{q}_g$'s exact normalized page-mass share and $\bar{m}_j = \frac{1}{G}\sum_g m_g(j)$ its group average. For $k \geq 2$, among shared sets of k pages that always include the sink and most recent page, the top-$(k-2)$ by $\bar{m}_j$ over the remaining pages maximizes the equal-weight group-average coverage $\frac{1}{G}\sum_g \gamma_S(\mathbf{q}_g)$, hence minimizes the correspondingly averaged bound of Lemma 1(i).*

Proof. Let $Z_g = \sum_j e^{s_j(\mathbf{q}_g)}$ and $m_g(j) = e^{s_j(\mathbf{q}_g)}/Z_g$. For any shared set S,

$$\frac{1}{G}\sum_g \gamma_S(\mathbf{q}_g) = \frac{1}{G}\sum_g \sum_{j \in S} m_g(j) = \sum_{j \in S} \bar{m}_j.$$

The mandatory sink and most recent pages contribute the same amount to every candidate set. The largest remaining $k-2$ shares therefore maximize group-average coverage. All queries in the group share one KV head and value-norm bound R; averaging Lemma 1(i) gives $2R(1 - \sum_{j \in S} \bar{m}_j)$, minimized by the same set. □

A.5 MOMENT APPROXIMATION AND OUTLIER CAPTURE

Let $\boldsymbol{\phi}_{j,\ell}$ denote the right singular vectors of $\mathbf{D}_j$. For a completed page $|P_j| = B$, write $a_j = \mathbf{q}^\top \boldsymbol{\mu}_j/\sqrt{d}$ and $z_i = \mathbf{q}^\top \boldsymbol{\delta}_i/\sqrt{d}$. Its exact score is

$$s_j(\mathbf{q}) = a_j + \log \sum_{i \in P_j} e^{z_i} = a_j + \log B + \log \mathbb{E}_i\big[e^{z_i}\big], \tag{3}$$

where $\mathbb{E}_i$ averages over the page's keys.

Proposition 3 (Quadratic approximation to page LSE)**.** *Fix a query* $\mathbf{q}$ *and page* j*; let* $\{z_i\}$ *be the centered per-key logits of Eq.* (3) *with empirical cumulants* κ_m*, and let* $L_j = \max_{i \in P_j} |z_i|$*. Remove the page-independent* $\log B$ *and define the rank-*r *quadratic score*

$$\hat{s}_j^{\mathrm{quad},r} := a_j + \frac{1}{2Bd} \sum_{\ell \le r} \sigma_{j,\ell}^2 (\boldsymbol{\phi}_{j,\ell}^\top \mathbf{q})^2.$$

With

$$R_j^{\mathrm{cum}} := \log \mathbb{E}_i[e^{z_i}] - \tfrac{1}{2}\kappa_2, \qquad R_j^{\mathrm{rank}} := \frac{1}{2Bd} \sum_{\ell > r} \sigma_{j,\ell}^2 (\boldsymbol{\phi}_{j,\ell}^\top \mathbf{q})^2,$$

the decomposition $s_j(\mathbf{q}) - \log B = \hat{s}_j^{\mathrm{quad},r} + R_j^{\mathrm{cum}} + R_j^{\mathrm{rank}}$ *is exact. The remainders obey*

$$|R_j^{\mathrm{cum}}| \le L_j^3/3, \qquad R_j^{\mathrm{rank}} \le \sigma_{j,r+1}^2 \, \|\mathbf{q}\|^2/(2Bd).$$

Proof of Proposition 3. By Eq. (3), $K(t) \coloneqq \log \mathbb{E}_i[e^{t z_i}]$ is the cumulant generating function of the empirical deviations $\{z_i\}$, finite and smooth for all real t, with $K(0) = 0$, $K'(0) = \kappa_1 = 0$ (the page is centered), $K''(0) = \kappa_2$, and $s_j(\mathbf{q}) - a_j - \log B = K(1)$. Taylor's theorem with integral remainder at $t = 1$ gives

$$K(1) = \tfrac{1}{2}\kappa_2 + \int_0^1 \tfrac{(1-t)^2}{2} \, K'''(t)\, dt.$$

Here $K'''(t)$ is the third cumulant of the exponentially tilted law $\pi_t(i) \propto e^{t z_i}$, which is supported on the same deviations. Under it $|z| \le L_j$, so its variance is at most L_j^2 and, since $|z - \mathbb{E}_t z| \le 2L_j$ pointwise, $|K'''(t)| = |\mathbb{E}_t(z - \mathbb{E}_t z)^3| \le 2L_j \operatorname{Var}_t(z) \le 2L_j^3$. Integrating, $|R_j^{\mathrm{cum}}| \le 2L_j^3 \int_0^1 \frac{(1-t)^2}{2}\, dt = L_j^3/3$. For the rank term, $\kappa_2 = \mathbb{E}_i z_i^2 = \mathbf{q}^\top \mathbf{S}_j \mathbf{q}/(Bd)$ with $\mathbf{S}_j = \sum_\ell \sigma_{j,\ell}^2 \boldsymbol{\phi}_{j,\ell} \boldsymbol{\phi}_{j,\ell}^\top$. Separating the top r terms of $\mathbf{q}^\top \mathbf{S}_j \mathbf{q}$ gives the stated rank remainder. Since the $\boldsymbol{\phi}_{j,\ell}$ are orthonormal, $\sum_{\ell > r} (\boldsymbol{\phi}_{j,\ell}^\top \mathbf{q})^2 \le \|\mathbf{q}\|^2$; multiplying by the largest omitted singular value gives $R_j^{\mathrm{rank}} \le \sigma_{j,r+1}^2 \|\mathbf{q}\|^2/(2Bd)$. □

Lemma 3 (Outlier capture)**.** *Fix* $\mathbf{q}$*, a page* j *with centered deviations* $\{\boldsymbol{\delta}_i\}$*, and* $1 \le r < d$*. For every* $i^\star \in P_j$*,*

$$\sum_{\ell \le r} \sigma_{j,\ell}^2 (\boldsymbol{\phi}_{j,\ell}^\top \mathbf{q})^2 \ge (\mathbf{q}^\top \boldsymbol{\delta}_{i^\star})^2 - \sigma_{j,r+1}^2 \|\mathbf{q}\|^2,$$

$$\sigma_{j,r+1}^2 \le \lambda_r\Big(\textstyle\sum_{i \ne i^\star} \boldsymbol{\delta}_i \boldsymbol{\delta}_i^\top\Big),$$

with λ_r *the* r*-th largest eigenvalue.*

Proof of Lemma 3. Write $\mathbf{S}_j = \boldsymbol{\delta}_{i^\star} \boldsymbol{\delta}_{i^\star}^\top + \mathbf{E}$, where $\mathbf{E} = \sum_{i \ne i^\star} \boldsymbol{\delta}_i \boldsymbol{\delta}_i^\top \succeq 0$. The omitted singular values are at most $\sigma_{j,r+1}$, so

$$\begin{aligned} \sum_{\ell \le r} \sigma_{j,\ell}^2 (\boldsymbol{\phi}_{j,\ell}^\top \mathbf{q})^2 &= \mathbf{q}^\top \mathbf{S}_j \mathbf{q} - \sum_{\ell > r} \sigma_{j,\ell}^2 (\boldsymbol{\phi}_{j,\ell}^\top \mathbf{q})^2 \\ &\ge \mathbf{q}^\top \mathbf{S}_j \mathbf{q} - \sigma_{j,r+1}^2 \|\mathbf{q}\|^2. \end{aligned}$$

The first claim follows from $\mathbf{S}_j \succeq \boldsymbol{\delta}_{i^\star} \boldsymbol{\delta}_{i^\star}^\top$. For the second, Weyl's inequality for this rank-one split gives

$$\sigma_{j,r+1}^2 = \lambda_{r+1}(\mathbf{S}_j) \le \lambda_2(\boldsymbol{\delta}_{i^\star} \boldsymbol{\delta}_{i^\star}^\top) + \lambda_r(\mathbf{E}) = \lambda_r(\mathbf{E}).$$

The rank-one term has zero second eigenvalue. □

B Detailed Results

This appendix gives the representation-scope controls of §3, the per-task scores, and the GH200 efficiency protocol behind §5. *Oracle* uses exact-LSE page scores and the share-average GQA selector (App. A.4).

B.1 Representation scope

All controls use post-RoPE keys from the same RULER-32K records, with $B{=}16$ and the shared GQA selector. At $b{=}128$, eight page slots are available per KV head; the sink and current pages occupy two of them. At $b{=}256$, fourteen of the sixteen slots remain for selection.

For the hybrid controls, each completed page retains its centroid $\boldsymbol{\mu}_j$ from §3, while the shared basis contains the top r eigenvectors of $\sum_{j\in\mathcal{S}} \mathbf{H}_j$. The sequence fit pools completed prompt pages and freezes after prefill. Global fits pool page-centered residuals from the independent WikiText-2 raw validation split (Merity et al., 2016) in 4K or 32K windows and freeze before evaluation. The fully shared controls instead fit one centroid and basis to the original keys of the sequence or calibration set. Their sequence fit includes the incomplete final prompt page; the hybrid fit does not.

All arms use int4 bases and int8 centroids and coefficients with bf16 scales. Within the r4 storage tier, page-local, hybrid, and fully shared fits use ranks 4, 20, and 28, respectively, at 490.0, 487.8, and 488.4 selector bytes per page and KV head. The corresponding r8 tier uses ranks 8, 40, and 48 at 818.0, 813.5, and 814.1 bytes. The global 4K and 32K labels specify calibration-window length, not RULER context length. Table B.1 gives both tiers at $b \in \{128, 256\}$.

Table B.1: Representation-scope results on RULER-32K. The scope columns give the centroid and basis fits; Page/Page is LOCKS. Mass, recall, and task scores are percentages.

Centroid	Basis	Rank	Mass	Recall	Score
r4 storage tier, $b{=}128$					
Page	Page	4	**85.8**	**90.5**	**80.1**
Page	Sequence	20	84.4	81.3	77.0
Page	Global 4K	20	84.6	81.9	76.3
Page	Global 32K	20	84.5	81.9	75.2
Sequence	Sequence	28	81.2	60.2	47.6
Global 4K	Global 4K	28	81.3	58.0	63.1
Global 32K	Global 32K	28	82.2	63.3	56.9
r4 storage tier, $b{=}256$					
Page	Page	4	**87.6**	**86.1**	**84.9**
Page	Sequence	20	86.3	73.7	81.4
Page	Global 4K	20	86.5	74.2	81.4
Page	Global 32K	20	86.4	74.4	81.1
Sequence	Sequence	28	83.6	51.6	65.1
Global 4K	Global 4K	28	83.6	45.6	70.0
Global 32K	Global 32K	28	84.4	53.7	72.4
r8 storage tier, $b{=}128$					
Page	Page	8	**86.0**	**93.6**	**83.4**
Page	Sequence	40	84.8	83.6	79.4
Page	Global 4K	40	85.1	84.2	80.0
Page	Global 32K	40	84.9	84.2	78.8
Sequence	Sequence	48	83.2	70.9	69.3
Global 4K	Global 4K	48	83.2	65.2	74.4
Global 32K	Global 32K	48	83.9	73.4	77.0
r8 storage tier, $b{=}256$					
Page	Page	8	**87.7**	**90.6**	**85.5**
Page	Sequence	40	86.7	77.1	80.0
Page	Global 4K	40	87.0	77.4	81.0
Page	Global 32K	40	86.8	77.8	82.2
Sequence	Sequence	48	85.4	62.8	74.8
Global 4K	Global 4K	48	85.2	52.6	79.7
Global 32K	Global 32K	48	85.9	64.5	79.6

B.2 RETRIEVAL AND LONG-DOCUMENT QA

B.2.1 BASELINES

LongBench-v1 uses the full split; RULER uses the 16K and 32K settings, all with Llama-3.1-8B. FullKV supplies the dense reference score. The retrieval baselines are:

- **Quest.** Our GQA port selects per query head. A group attends the union of its heads' pages, between b and Gb tokens (G=4 for Llama-3.1-8B).
- **ShadowKV.** Its 384-token outlier block and 32-token local window count within b, leaving 96, 608, and 1632 tokens for additional selection at b=512, 1024, and 2048. At $b \leq 256$ the fixed block does not fit, so those dense-fallback points are omitted.
- **RocketKV.** Splits b between attention and top-k estimation; attended tokens are $b/2$, measured decode traffic is 0.77–$1.00\times$ nominal, and the full cache remains resident.
- **KVzip.** Retains the question and generated turn beyond b, so decode reads $b + |q| + g$ after g generated tokens.

B.2.2 LONGBENCH-V1

Table B.2: LongBench-v1 scores by subset (Llama-3.1-8B), with all methods at each budget b.

Subtask	FullKV	Oracle	r8i4	r4i4	r2i4	Quest	KVzip	ShadowKV	RocketKV
budget $b = 64$									
narrativeqa	29.4	28.7	29.3	28.0	28.4	17.3	21.2	N/A	26.3
qasper	47.2	44.9	45.5	45.9	43.7	27.3	21.1	N/A	28.0
multifieldqa_en	55.4	56.6	56.0	55.3	56.4	35.8	34.7	N/A	55.9
hotpotqa	58.2	59.5	58.5	57.1	57.9	42.8	52.0	N/A	57.1
2wikimqa	51.9	50.9	50.8	50.6	50.5	36.0	44.8	N/A	48.4
musique	32.9	33.0	31.7	31.5	32.4	21.6	24.5	N/A	28.5
dureader	33.3	32.2	31.9	29.4	30.3	14.1	19.7	N/A	23.1
gov_report	35.4	29.5	29.8	28.9	26.9	16.3	15.7	N/A	22.2
qmsum	24.8	24.2	24.1	23.9	24.0	18.0	20.4	N/A	22.5
multi_news	27.2	25.9	25.8	25.1	24.8	18.2	18.6	N/A	23.6
trec	29.5	28.0	29.0	27.0	24.0	22.8	9.5	N/A	19.5
triviaqa	92.1	91.6	92.0	91.3	91.0	85.1	91.5	N/A	91.2
samsum	40.9	38.5	38.6	38.8	38.7	25.1	38.6	N/A	37.6
passage_retrieval_en	100.0	98.5	97.5	98.0	97.0	62.5	99.5	N/A	98.5
Avg	**47.0**	**45.9**	**45.7**	**45.1**	**44.7**	**31.6**	**36.6**	**N/A**	**41.6**
budget $b = 128$									
narrativeqa	29.4	28.7	28.8	28.2	28.2	22.0	21.9	N/A	26.7
qasper	47.2	46.5	46.6	46.6	47.2	37.5	27.6	N/A	40.3
multifieldqa_en	55.4	57.1	56.6	56.4	55.9	46.3	40.5	N/A	57.1
hotpotqa	58.2	59.1	59.1	59.2	59.3	52.5	51.2	N/A	58.9
2wikimqa	51.9	52.8	51.0	50.5	51.0	47.0	45.7	N/A	51.0
musique	32.9	31.6	31.6	31.4	31.6	26.0	25.0	N/A	31.0
dureader	33.3	34.1	35.0	33.9	33.0	20.1	20.1	N/A	27.1
gov_report	35.4	34.1	33.6	32.6	32.0	21.1	17.0	N/A	24.2
qmsum	24.8	24.1	24.5	24.6	24.0	19.8	21.0	N/A	23.3
multi_news	27.2	26.3	26.3	26.5	26.2	24.2	23.6	N/A	24.0
trec	29.5	28.0	30.0	29.5	28.5	25.5	11.0	N/A	23.5
triviaqa	92.1	91.7	91.8	91.8	91.8	90.0	91.5	N/A	91.4
samsum	40.9	38.7	38.4	38.2	38.8	33.4	38.9	N/A	38.8
passage_retrieval_en	100.0	99.5	100.0	100.0	100.0	92.2	99.5	N/A	99.5
Avg	**47.0**	**46.6**	**46.7**	**46.4**	**46.2**	**39.8**	**38.2**	**N/A**	**44.1**
budget $b = 256$									
narrativeqa	29.4	28.6	28.1	28.1	29.0	26.0	23.2	N/A	29.6
qasper	47.2	45.8	46.5	46.2	46.1	44.0	39.5	N/A	43.8
multifieldqa_en	55.4	56.9	56.8	56.3	56.0	51.9	50.5	N/A	56.9
hotpotqa	58.2	58.3	58.6	58.4	58.7	55.6	52.5	N/A	58.4
2wikimqa	51.9	52.5	51.6	51.7	51.5	49.3	48.5	N/A	51.9
musique	32.9	32.0	31.8	32.4	33.2	30.1	25.1	N/A	32.2
dureader	33.3	34.0	33.7	33.4	33.9	27.1	19.6	N/A	30.3
gov_report	35.4	34.4	34.2	33.6	33.9	29.4	20.8	N/A	26.6
qmsum	24.8	24.4	25.0	24.7	24.7	22.0	22.2	N/A	24.5
multi_news	27.2	27.1	26.8	26.8	26.6	26.7	25.8	N/A	24.5
trec	29.5	30.0	30.5	30.0	30.0	26.6	21.1	N/A	30.0

Table B.2 (cont.)

Subtask	FullKV	Oracle	r8i4	r4i4	r2i4	Quest	KVzip	ShadowKV	RocketKV
triviaqa	92.1	91.8	92.1	91.5	92.5	90.8	91.5	N/A	91.7
samsum	40.9	39.5	38.5	38.8	39.1	37.4	39.0	N/A	40.0
passage_retrieval_en	100.0	99.5	100.0	100.0	100.0	97.0	99.0	N/A	100.0
Avg	**47.0**	**46.8**	**46.7**	**46.6**	**46.8**	**43.8**	**41.3**	**N/A**	**45.7**
budget $b = 512$									
narrativeqa	29.4	29.0	29.1	29.1	28.3	28.5	24.9	29.5	29.9
qasper	47.2	45.9	46.6	47.2	46.6	45.3	44.4	45.6	46.4
multifieldqa_en	55.4	55.6	56.0	56.4	56.8	52.8	54.0	56.7	58.3
hotpotqa	58.2	58.4	59.0	58.8	58.8	57.9	55.3	58.6	59.1
2wikimqa	51.9	51.8	51.8	52.1	51.9	51.7	49.8	51.0	50.2
musique	32.9	32.4	32.4	32.4	32.7	32.3	28.3	33.1	32.6
dureader	33.3	33.1	33.9	33.2	33.0	32.3	24.0	32.1	32.2
gov_report	35.4	34.8	34.8	34.5	34.2	33.8	26.4	32.7	30.3
qmsum	24.8	24.9	24.9	24.6	24.6	24.4	23.6	24.8	25.1
multi_news	27.2	27.2	27.2	27.2	27.2	27.7	26.8	26.8	25.8
trec	29.5	30.0	30.5	30.5	30.5	28.0	28.5	29.5	30.0
triviaqa	92.1	92.1	92.2	92.1	91.7	91.0	91.9	91.7	92.2
samsum	40.9	40.4	40.4	39.6	40.1	37.3	39.3	38.4	40.2
passage_retrieval_en	100.0	100.0	100.0	100.0	99.5	99.5	97.0	98.0	99.5
Avg	**47.0**	**46.8**	**47.1**	**47.0**	**46.8**	**45.9**	**43.9**	**46.3**	**46.6**
budget $b = 1024$									
narrativeqa	29.4	29.5	30.0	29.7	30.0	28.7	28.3	29.8	28.7
qasper	47.2	47.0	47.3	46.8	47.0	45.7	46.4	46.3	47.0
multifieldqa_en	55.4	55.8	55.8	55.6	56.0	53.7	56.2	57.0	56.3
hotpotqa	58.2	58.8	59.6	59.6	59.0	57.5	58.9	59.2	59.0
2wikimqa	51.9	52.4	52.1	51.9	51.6	52.2	51.7	50.7	51.5
musique	32.9	32.5	33.0	32.8	32.6	32.8	32.2	33.7	31.6
dureader	33.3	33.9	33.2	33.3	33.1	33.5	31.0	33.2	32.6
gov_report	35.4	35.2	35.0	34.9	34.5	34.8	31.7	33.9	32.7
qmsum	24.8	24.9	24.8	24.9	24.4	25.0	24.5	24.9	25.4
multi_news	27.2	27.1	26.9	27.2	27.1	27.3	26.8	26.9	26.5
trec	29.5	30.5	30.0	30.0	30.0	29.0	30.0	30.0	29.5
triviaqa	92.1	92.1	92.1	92.2	92.2	91.1	91.9	91.8	92.2
samsum	40.9	40.2	40.5	40.6	40.3	39.5	39.2	40.6	40.3
passage_retrieval_en	100.0	100.0	100.0	100.0	99.5	99.5	91.0	99.5	100.0
Avg	**47.0**	**47.1**	**47.2**	**47.1**	**46.9**	**46.5**	**45.7**	**47.0**	**46.7**
budget $b = 2048$									
narrativeqa	29.4	30.3	30.2	30.0	30.3	29.7	29.1	29.6	30.1
qasper	47.2	47.0	47.3	47.4	47.2	46.2	47.6	46.4	46.9
multifieldqa_en	55.4	54.8	55.0	54.9	54.9	55.3	56.4	56.2	55.0
hotpotqa	58.2	59.0	59.5	59.5	59.7	57.7	59.2	59.4	58.7
2wikimqa	51.9	51.7	51.6	51.2	51.0	52.4	51.5	50.7	51.6
musique	32.9	32.9	33.1	33.1	32.9	31.6	33.4	33.1	32.3
dureader	33.3	33.6	33.2	33.6	33.5	34.2	33.4	32.6	32.8
gov_report	35.4	35.1	35.5	35.1	34.5	35.7	34.3	34.5	33.8
qmsum	24.8	25.3	25.1	25.2	24.7	25.3	24.9	25.1	24.9
multi_news	27.2	27.1	27.1	26.9	26.9	26.9	26.9	26.9	27.2
trec	29.5	30.0	30.0	30.0	30.0	29.5	30.5	30.0	30.5
triviaqa	92.1	92.1	92.2	92.2	92.1	91.6	92.1	91.8	92.2
samsum	40.9	40.4	40.9	40.3	40.6	40.5	40.0	40.7	40.6
passage_retrieval_en	100.0	100.0	100.0	100.0	99.5	100.0	98.5	99.5	100.0
Avg	**47.0**	**47.1**	**47.2**	**47.1**	**47.0**	**46.9**	**47.0**	**46.9**	**46.9**

B.2.3 RULER-16K

Table B.3: RULER-16K scores by task (Llama-3.1-8B), with all methods at each budget b.

Subtask	FullKV	Oracle	r8i4	r4i4	r2i4	Quest	KVzip	ShadowKV	RocketKV
budget $b = 64$									
NIAH-S1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	N/A	100.0
NIAH-S2	100.0	100.0	100.0	100.0	98.0	74.0	0.0	N/A	100.0
NIAH-S3	100.0	100.0	100.0	100.0	96.0	4.0	0.0	N/A	100.0
NIAH-MK1	100.0	100.0	100.0	98.0	96.0	88.0	0.0	N/A	100.0
NIAH-MK2	100.0	98.0	98.0	78.0	80.0	12.0	0.0	N/A	98.0
NIAH-MK3	98.0	70.0	44.0	14.0	4.0	0.0	0.0	N/A	86.0

Table B.3 (cont.)

Subtask	FullKV	Oracle	r8i4	r4i4	r2i4	Quest	KVzip	ShadowKV	RocketKV
NIAH-MV	98.5	79.5	82.5	77.0	67.0	26.5	0.0	N/A	44.0
NIAH-MQ	99.0	90.0	87.5	81.5	74.5	25.5	0.0	N/A	37.5
VT	100.0	82.8	78.4	76.8	76.0	59.6	88.4	N/A	20.8
CWE	87.4	5.0	6.6	6.0	3.0	6.0	0.2	N/A	0.8
FWE	96.7	75.3	78.0	76.0	72.0	44.0	0.0	N/A	76.7
QA-1	84.0	82.0	82.0	82.0	82.0	44.0	56.0	N/A	84.0
QA-2	62.0	58.0	58.0	52.0	50.0	30.0	38.0	N/A	54.0
Avg	**94.3**	**80.0**	**78.1**	**72.4**	**69.1**	**39.5**	**21.7**	**N/A**	**69.4**
budget $b = 128$									
NIAH-S1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	N/A	100.0
NIAH-S2	100.0	100.0	100.0	100.0	100.0	88.0	2.0	N/A	100.0
NIAH-S3	100.0	100.0	100.0	100.0	100.0	32.0	0.0	N/A	100.0
NIAH-MK1	100.0	100.0	100.0	100.0	100.0	98.0	2.0	N/A	100.0
NIAH-MK2	100.0	100.0	100.0	96.0	96.0	40.0	0.0	N/A	100.0
NIAH-MK3	98.0	100.0	100.0	78.0	62.0	0.0	0.0	N/A	100.0
NIAH-MV	98.5	96.0	96.5	96.5	90.0	65.0	0.0	N/A	84.5
NIAH-MQ	99.0	97.5	97.0	97.5	96.0	59.0	0.0	N/A	93.0
VT	100.0	95.6	94.0	96.4	98.0	81.2	96.8	N/A	89.2
CWE	87.4	19.4	19.2	13.6	13.2	7.8	0.2	N/A	5.8
FWE	96.7	82.7	80.0	80.0	82.0	62.7	0.0	N/A	86.7
QA-1	84.0	86.0	88.0	88.0	88.0	76.0	62.0	N/A	88.0
QA-2	62.0	60.0	62.0	62.0	60.0	42.0	38.0	N/A	60.0
Avg	**94.3**	**87.5**	**87.4**	**85.2**	**83.5**	**57.8**	**23.2**	**N/A**	**85.2**
budget $b = 256$									
NIAH-S1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	N/A	100.0
NIAH-S2	100.0	100.0	100.0	100.0	100.0	98.0	16.0	N/A	100.0
NIAH-S3	100.0	100.0	100.0	100.0	100.0	58.0	0.0	N/A	100.0
NIAH-MK1	100.0	100.0	100.0	100.0	100.0	98.0	16.0	N/A	100.0
NIAH-MK2	100.0	100.0	100.0	100.0	100.0	80.0	46.0	N/A	100.0
NIAH-MK3	98.0	100.0	100.0	100.0	90.0	0.0	0.0	N/A	100.0
NIAH-MV	98.5	98.0	98.0	97.5	96.5	88.0	3.0	N/A	97.5
NIAH-MQ	99.0	99.0	98.0	99.5	99.0	79.5	7.5	N/A	98.0
VT	100.0	99.6	99.6	99.6	98.8	84.8	98.0	N/A	95.6
CWE	87.4	43.4	46.0	35.2	34.4	21.6	0.2	N/A	16.4
FWE	96.7	90.0	86.7	88.7	88.7	75.3	1.3	N/A	84.0
QA-1	84.0	88.0	88.0	88.0	88.0	86.0	64.0	N/A	84.0
QA-2	62.0	60.0	62.0	62.0	60.0	52.0	38.0	N/A	62.0
Avg	**94.3**	**90.6**	**90.6**	**90.0**	**88.9**	**70.9**	**30.0**	**N/A**	**87.5**
budget $b = 512$									
NIAH-S1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
NIAH-S2	100.0	100.0	100.0	100.0	100.0	98.0	94.0	100.0	100.0
NIAH-S3	100.0	100.0	100.0	100.0	100.0	82.0	50.0	100.0	98.0
NIAH-MK1	100.0	100.0	100.0	100.0	100.0	100.0	94.0	100.0	100.0
NIAH-MK2	100.0	100.0	100.0	100.0	100.0	96.0	100.0	98.0	100.0
NIAH-MK3	98.0	98.0	98.0	98.0	98.0	18.0	68.0	78.0	100.0
NIAH-MV	98.5	98.5	97.0	97.5	97.0	94.5	38.5	91.5	98.0
NIAH-MQ	99.0	98.5	98.0	98.5	98.5	92.0	65.0	95.0	98.5
VT	100.0	99.6	99.6	99.6	99.6	94.8	98.4	92.4	98.4
CWE	87.4	58.6	57.2	53.2	55.4	30.0	0.2	16.6	36.4
FWE	96.7	90.0	90.0	91.3	94.0	81.3	37.3	83.3	88.0
QA-1	84.0	86.0	88.0	88.0	88.0	86.0	76.0	84.0	86.0
QA-2	62.0	60.0	60.0	62.0	62.0	58.0	52.0	60.0	62.0
Avg	**94.3**	**91.5**	**91.4**	**91.4**	**91.7**	**79.3**	**67.2**	**84.5**	**89.6**
budget $b = 1024$									
NIAH-S1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
NIAH-S2	100.0	100.0	100.0	100.0	100.0	96.0	100.0	100.0	100.0
NIAH-S3	100.0	100.0	100.0	100.0	100.0	94.0	100.0	100.0	100.0
NIAH-MK1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
NIAH-MK2	100.0	100.0	100.0	100.0	100.0	98.0	100.0	100.0	100.0
NIAH-MK3	98.0	98.0	98.0	98.0	98.0	44.0	98.0	94.0	98.0
NIAH-MV	98.5	98.0	98.0	98.0	98.5	97.5	77.5	94.0	98.0
NIAH-MQ	99.0	99.5	99.5	99.5	98.5	95.5	92.5	98.0	99.0
VT	100.0	100.0	100.0	100.0	98.4	96.4	98.8	96.8	99.2
CWE	87.4	69.2	72.2	65.8	70.4	50.0	3.8	59.0	54.6
FWE	96.7	93.3	93.3	92.0	94.0	84.0	76.7	93.3	88.7
QA-1	84.0	86.0	88.0	88.0	86.0	82.0	88.0	86.0	86.0
QA-2	62.0	64.0	60.0	60.0	60.0	54.0	56.0	60.0	62.0
Avg	**94.3**	**92.9**	**93.0**	**92.4**	**92.6**	**84.0**	**83.9**	**90.9**	**91.2**

Table B.3 (cont.)

Subtask	FullKV	Oracle	r8i4	r4i4	r2i4	Quest	KVzip	ShadowKV	RocketKV
budget $b = 2048$									
NIAH-S1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
NIAH-S2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
NIAH-S3	100.0	100.0	100.0	100.0	100.0	98.0	100.0	100.0	100.0
NIAH-MK1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
NIAH-MK2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
NIAH-MK3	98.0	98.0	98.0	98.0	98.0	82.0	100.0	98.0	98.0
NIAH-MV	98.5	98.5	98.0	98.0	98.0	98.0	97.0	92.5	98.0
NIAH-MQ	99.0	99.5	98.5	98.5	98.5	98.0	98.0	98.0	99.0
VT	100.0	100.0	100.0	100.0	100.0	97.2	100.0	100.0	99.6
CWE	87.4	80.4	79.6	77.6	78.0	67.6	50.4	73.8	73.2
FWE	96.7	94.0	94.0	94.0	95.3	89.3	92.0	92.7	92.7
QA-1	84.0	86.0	88.0	86.0	86.0	84.0	88.0	86.0	86.0
QA-2	62.0	62.0	62.0	62.0	60.0	60.0	62.0	62.0	64.0
Avg	**94.3**	**93.7**	**93.7**	**93.4**	**93.4**	**90.3**	**91.3**	**92.5**	**93.1**

B.2.4 RULER-32K

Table B.4: RULER-32K scores by task (Llama-3.1-8B), with all methods at each budget b.

Subtask	FullKV	Oracle	r8i4	r4i4	r2i4	Quest	KVzip	ShadowKV	RocketKV
budget $b = 64$									
NIAH-S1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	N/A	100.0
NIAH-S2	100.0	100.0	100.0	100.0	100.0	78.0	0.0	N/A	100.0
NIAH-S3	100.0	100.0	100.0	100.0	92.0	0.0	0.0	N/A	100.0
NIAH-MK1	100.0	100.0	98.0	96.0	94.0	82.0	0.0	N/A	98.0
NIAH-MK2	100.0	92.0	86.0	60.0	64.0	10.0	0.0	N/A	90.0
NIAH-MK3	100.0	20.0	20.0	0.0	0.0	0.0	0.0	N/A	10.0
NIAH-MV	99.5	80.0	79.5	77.5	67.0	22.5	0.0	N/A	26.0
NIAH-MQ	99.5	89.5	85.0	86.5	76.5	25.5	0.0	N/A	2.0
VT	100.0	80.4	80.4	75.2	72.0	66.8	80.0	N/A	11.6
CWE	28.8	1.0	1.8	0.8	0.2	2.6	0.0	N/A	0.4
FWE	82.7	58.7	61.3	56.7	55.3	38.0	0.0	N/A	56.0
QA-1	90.0	88.0	88.0	88.0	88.0	48.0	62.0	N/A	86.0
QA-2	54.0	50.0	50.0	50.0	52.0	28.0	34.0	N/A	44.0
Avg	**88.8**	**73.8**	**73.1**	**68.5**	**66.2**	**38.6**	**21.2**	**N/A**	**55.7**
budget $b = 128$									
NIAH-S1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	N/A	100.0
NIAH-S2	100.0	100.0	100.0	100.0	100.0	92.0	0.0	N/A	100.0
NIAH-S3	100.0	100.0	100.0	100.0	100.0	28.0	0.0	N/A	100.0
NIAH-MK1	100.0	100.0	100.0	100.0	100.0	100.0	0.0	N/A	100.0
NIAH-MK2	100.0	100.0	100.0	96.0	94.0	38.0	0.0	N/A	100.0
NIAH-MK3	100.0	90.0	84.0	40.0	20.0	0.0	0.0	N/A	98.0
NIAH-MV	99.5	95.0	96.0	95.0	91.5	49.5	0.0	N/A	84.0
NIAH-MQ	99.5	96.5	97.5	98.0	95.0	53.5	0.0	N/A	95.0
VT	100.0	89.6	90.4	92.8	89.2	75.2	85.2	N/A	80.0
CWE	28.8	4.0	1.8	2.6	0.8	3.6	0.0	N/A	0.2
FWE	82.7	68.7	68.7	73.3	71.3	53.3	0.0	N/A	65.3
QA-1	90.0	90.0	90.0	88.0	90.0	72.0	62.0	N/A	90.0
QA-2	54.0	52.0	56.0	56.0	58.0	32.0	38.0	N/A	56.0
Avg	**88.8**	**83.5**	**83.4**	**80.1**	**77.7**	**53.6**	**21.9**	**N/A**	**82.2**
budget $b = 256$									
NIAH-S1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	N/A	100.0
NIAH-S2	100.0	100.0	100.0	100.0	100.0	92.0	2.0	N/A	100.0
NIAH-S3	100.0	100.0	100.0	100.0	100.0	44.0	0.0	N/A	100.0
NIAH-MK1	100.0	100.0	100.0	100.0	100.0	100.0	2.0	N/A	100.0
NIAH-MK2	100.0	100.0	100.0	100.0	100.0	58.0	6.0	N/A	100.0
NIAH-MK3	100.0	100.0	96.0	92.0	70.0	0.0	0.0	N/A	100.0
NIAH-MV	99.5	97.5	98.5	97.5	97.5	76.5	0.0	N/A	97.0
NIAH-MQ	99.5	99.0	99.0	99.0	99.5	81.0	2.0	N/A	98.0
VT	100.0	93.6	92.4	92.4	91.6	84.4	91.2	N/A	91.6
CWE	28.8	5.6	8.4	3.4	4.0	4.0	0.0	N/A	1.6
FWE	82.7	71.3	71.3	73.3	72.7	66.0	0.0	N/A	67.3
QA-1	90.0	90.0	90.0	90.0	90.0	80.0	64.0	N/A	90.0
QA-2	54.0	54.0	56.0	56.0	56.0	48.0	36.0	N/A	54.0

Table B.4 (cont.)

Subtask	FullKV	Oracle	r8i4	r4i4	r2i4	Quest	KVzip	ShadowKV	RocketKV
Avg	**88.8**	**85.5**	**85.5**	**84.9**	**83.2**	**64.1**	**23.3**	**N/A**	**84.6**
budget $b = 512$									
NIAH-S1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
NIAH-S2	100.0	100.0	100.0	100.0	100.0	98.0	42.0	100.0	100.0
NIAH-S3	100.0	100.0	100.0	100.0	100.0	70.0	2.0	100.0	100.0
NIAH-MK1	100.0	100.0	100.0	100.0	100.0	100.0	30.0	98.0	100.0
NIAH-MK2	100.0	100.0	100.0	100.0	100.0	88.0	46.0	96.0	100.0
NIAH-MK3	100.0	100.0	100.0	98.0	98.0	0.0	0.0	30.0	100.0
NIAH-MV	99.5	98.5	98.5	98.5	98.5	92.5	8.0	90.0	98.5
NIAH-MQ	99.5	97.5	98.0	98.5	99.5	90.0	25.0	95.5	99.5
VT	100.0	96.4	94.0	95.2	94.0	84.8	92.8	86.0	92.4
CWE	28.8	14.4	12.6	9.6	10.4	6.2	0.0	1.0	5.4
FWE	82.7	70.7	70.7	70.7	71.3	74.0	0.0	71.3	70.0
QA-1	90.0	90.0	90.0	90.0	90.0	86.0	70.0	90.0	90.0
QA-2	54.0	56.0	56.0	58.0	56.0	48.0	40.0	52.0	56.0
Avg	**88.8**	**86.4**	**86.1**	**86.0**	**86.0**	**72.1**	**35.1**	**77.7**	**85.5**
budget $b = 1024$									
NIAH-S1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
NIAH-S2	100.0	100.0	100.0	100.0	100.0	98.0	96.0	100.0	100.0
NIAH-S3	100.0	100.0	100.0	100.0	100.0	86.0	66.0	100.0	100.0
NIAH-MK1	100.0	100.0	100.0	100.0	100.0	100.0	96.0	100.0	100.0
NIAH-MK2	100.0	100.0	100.0	100.0	100.0	98.0	98.0	100.0	100.0
NIAH-MK3	100.0	100.0	100.0	100.0	96.0	22.0	42.0	92.0	100.0
NIAH-MV	99.5	99.0	99.5	99.5	99.0	96.0	46.0	98.5	99.0
NIAH-MQ	99.5	98.0	97.5	97.0	99.0	94.0	76.5	98.5	99.0
VT	100.0	96.8	97.2	97.2	97.2	88.8	94.8	93.2	94.8
CWE	28.8	18.4	16.0	13.8	17.4	12.8	0.0	14.6	15.0
FWE	82.7	72.7	73.3	72.0	71.3	76.0	20.0	70.7	66.7
QA-1	90.0	90.0	90.0	90.0	90.0	86.0	84.0	90.0	90.0
QA-2	54.0	56.0	60.0	60.0	60.0	50.0	44.0	54.0	56.0
Avg	**88.8**	**87.0**	**87.2**	**86.9**	**86.9**	**77.5**	**66.4**	**85.5**	**86.2**
budget $b = 2048$									
NIAH-S1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
NIAH-S2	100.0	100.0	100.0	100.0	100.0	96.0	100.0	100.0	100.0
NIAH-S3	100.0	100.0	100.0	100.0	100.0	90.0	100.0	100.0	100.0
NIAH-MK1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
NIAH-MK2	100.0	100.0	100.0	100.0	100.0	98.0	100.0	100.0	100.0
NIAH-MK3	100.0	100.0	98.0	98.0	96.0	64.0	96.0	100.0	100.0
NIAH-MV	99.5	99.5	99.5	99.5	99.5	98.0	76.0	99.0	99.5
NIAH-MQ	99.5	99.0	98.5	99.0	99.0	97.5	91.0	99.0	98.0
VT	100.0	98.0	98.8	98.4	98.4	93.6	96.0	96.8	96.8
CWE	28.8	25.2	25.0	19.8	21.6	13.0	0.2	20.0	32.2
FWE	82.7	74.0	74.7	75.3	74.7	80.7	63.3	72.7	71.3
QA-1	90.0	90.0	90.0	90.0	90.0	90.0	86.0	90.0	90.0
QA-2	54.0	58.0	60.0	60.0	60.0	56.0	54.0	54.0	56.0
Avg	**88.8**	**88.0**	**88.0**	**87.7**	**87.6**	**82.8**	**81.7**	**87.0**	**88.0**

B.2.5 INFINITEBENCH

We evaluate GLM-4-9B-Chat-1M on 11 tasks, excluding `math_calc` because its 30K-token generation exceeds baseline support under the common protocol. On this 100K-context benchmark, LOCKS reaches FullKV parity (43.6 vs 43.0 at b=2048), while the other deployable selectors score lower.

Table B.5: InfiniteBench scores by task and budget (GLM-4-9B-Chat-1M). Entries are means with 95% bootstrap CI half-widths; N/A marks budgets below ShadowKV's fixed block.

Task	FullKV	exact-LSE	r8i4	RocketKV	ShadowKV	Quest
budget $b = 64$						
passkey	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0	86.0 ±9.6	N/A	100.0 ±0.0
num-str	100.0 ±0.0	98.0 ±3.9	100.0 ±0.0	68.0 ±12.8	N/A	70.0 ±12.7
kv-retr	28.0 ±12.7	2.0 ±3.9	0.0 ±0.0	6.0 ±6.6	N/A	0.0 ±0.0
dialog-qa	34.0 ±12.9	20.0 ±10.9	16.0 ±10.3	2.0 ±3.9	N/A	12.0 ±8.9
book-sum	27.6 ±2.1	18.0 ±2.3	16.4 ±2.3	14.2 ±1.8	N/A	14.9 ±1.1
book-choice	82.0 ±10.6	86.0 ±9.5	82.0 ±10.4	78.0 ±11.5	N/A	60.0 ±13.5
book-qa-en	16.2 ±5.7	19.2 ±7.2	18.2 ±6.8	10.4 ±5.7	N/A	8.1 ±4.9

Table B.5 (cont.)

Task	FullKV	exact-LSE	r8i4	RocketKV	ShadowKV	Quest
book-qa-zh	16.8 ±3.8	14.7 ±4.9	16.7 ±4.9	13.8 ±5.1	N/A	7.4 ±2.8
math-find	30.0 ±12.6	24.0 ±11.7	26.0 ±12.0	22.0 ±11.3	N/A	28.0 ±12.3
code-run	6.0 ±6.5	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	N/A	0.0 ±0.0
code-debug	32.0 ±13.0	28.0 ±12.5	34.0 ±13.2	36.0 ±13.3	N/A	32.0 ±12.6
Avg	**43.0** ±2.1	**37.3** ±2.0	**37.2** ±2.2	**30.6** ±2.4	N/A	**30.2** ±2.6
budget $b = 128$						
passkey	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0	98.0 ±3.9	N/A	100.0 ±0.0
num-str	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0	66.0 ±13.1	N/A	84.0 ±10.1
kv-retr	28.0 ±12.7	2.0 ±3.9	2.0 ±3.8	16.0 ±10.2	N/A	0.0 ±0.0
dialog-qa	34.0 ±12.9	36.0 ±13.3	28.0 ±12.4	10.0 ±8.3	N/A	24.0 ±11.8
book-sum	27.6 ±2.1	17.8 ±2.1	18.1 ±2.3	15.5 ±2.0	N/A	13.8 ±1.6
book-choice	82.0 ±10.6	82.0 ±10.6	84.0 ±10.2	82.0 ±10.6	N/A	72.0 ±12.5
book-qa-en	16.2 ±5.7	18.6 ±6.9	20.0 ±6.8	14.3 ±6.5	N/A	14.7 ±6.9
book-qa-zh	16.8 ±3.8	14.3 ±3.9	14.9 ±4.5	16.9 ±6.0	N/A	6.3 ±2.2
math-find	30.0 ±12.6	32.0 ±12.8	30.0 ±12.6	26.0 ±12.1	N/A	30.0 ±12.4
code-run	6.0 ±6.5	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	N/A	2.0 ±3.8
code-debug	32.0 ±13.0	38.0 ±13.5	34.0 ±13.2	38.0 ±13.5	N/A	14.0 ±9.7
Avg	**43.0** ±2.1	**40.1** ±2.4	**39.2** ±2.3	**34.8** ±2.2	N/A	**32.8** ±2.0
budget $b = 256$						
passkey	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0	98.0 ±3.9	N/A	100.0 ±0.0
num-str	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0	84.0 ±10.2	N/A	96.0 ±5.4
kv-retr	28.0 ±12.7	16.0 ±10.3	6.0 ±6.6	18.0 ±11.0	N/A	0.0 ±0.0
dialog-qa	34.0 ±12.9	32.0 ±12.9	30.0 ±12.9	24.0 ±11.9	N/A	16.0 ±10.1
book-sum	27.6 ±2.1	21.5 ±2.4	21.4 ±2.3	18.8 ±1.8	N/A	14.8 ±1.2
book-choice	82.0 ±10.6	82.0 ±10.6	82.0 ±10.6	82.0 ±10.6	N/A	76.0 ±11.9
book-qa-en	16.2 ±5.7	21.0 ±7.5	21.6 ±7.5	18.7 ±7.3	N/A	17.9 ±7.3
book-qa-zh	16.8 ±3.8	16.5 ±4.5	16.3 ±4.6	18.3 ±5.8	N/A	10.8 ±3.6
math-find	30.0 ±12.6	32.0 ±12.9	30.0 ±12.6	26.0 ±12.0	N/A	32.0 ±12.8
code-run	6.0 ±6.5	2.0 ±3.9	2.0 ±3.8	0.0 ±0.0	N/A	0.0 ±0.0
code-debug	32.0 ±13.0	38.0 ±13.5	36.0 ±13.4	36.0 ±13.5	N/A	36.0 ±13.4
Avg	**43.0** ±2.1	**41.9** ±2.6	**40.5** ±2.4	**38.5** ±2.7	N/A	**36.3** ±2.3
budget $b = 512$						
passkey	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0	96.0 ±5.4	100.0 ±0.0	100.0 ±0.0
num-str	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0	68.0 ±13.0	98.0 ±3.8	88.0 ±9.1
kv-retr	28.0 ±12.7	24.0 ±11.9	14.0 ±9.7	28.0 ±12.7	0.0 ±0.0	0.0 ±0.0
dialog-qa	34.0 ±12.9	28.0 ±12.4	26.0 ±12.2	26.0 ±12.2	14.0 ±9.4	24.0 ±11.5
book-sum	27.6 ±2.1	24.8 ±2.4	23.3 ±2.2	21.6 ±2.1	17.6 ±2.0	12.9 ±1.5
book-choice	82.0 ±10.6	84.0 ±10.1	82.0 ±10.6	84.0 ±10.2	78.0 ±11.5	72.0 ±12.4
book-qa-en	16.2 ±5.7	20.5 ±7.5	19.0 ±7.4	19.5 ±7.5	19.1 ±7.0	14.5 ±5.8
book-qa-zh	16.8 ±3.8	18.0 ±4.6	17.9 ±4.6	16.7 ±5.1	17.0 ±5.2	10.0 ±3.1
math-find	30.0 ±12.6	32.0 ±12.8	34.0 ±13.0	24.0 ±11.6	22.0 ±11.4	32.0 ±12.8
code-run	6.0 ±6.5	2.0 ±3.8	2.0 ±3.8	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
code-debug	32.0 ±13.0	32.0 ±13.1	34.0 ±13.2	40.0 ±13.6	30.0 ±12.4	28.0 ±12.5
Avg	**43.0** ±2.1	**42.3** ±3.0	**41.1** ±2.4	**38.5** ±2.7	**36.0** ±2.1	**34.7** ±2.7
budget $b = 1024$						
passkey	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0
num-str	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0	98.0 ±3.8	100.0 ±0.0	82.0 ±10.7
kv-retr	28.0 ±12.7	28.0 ±12.5	28.0 ±12.5	36.0 ±13.5	6.0 ±6.6	0.0 ±0.0
dialog-qa	34.0 ±12.9	32.0 ±13.0	36.0 ±13.3	28.0 ±12.5	34.0 ±13.0	26.0 ±11.9
book-sum	27.6 ±2.1	24.9 ±2.1	26.5 ±2.1	22.1 ±2.2	19.3 ±2.3	15.6 ±1.9
book-choice	82.0 ±10.6	84.0 ±10.1	82.0 ±10.6	84.0 ±10.2	78.0 ±11.5	76.0 ±11.7
book-qa-en	16.2 ±5.7	19.0 ±7.4	18.7 ±7.2	19.6 ±7.6	19.3 ±7.5	17.2 ±6.0
book-qa-zh	16.8 ±3.8	18.0 ±4.6	18.5 ±4.8	16.0 ±3.8	19.2 ±4.1	13.7 ±4.1
math-find	30.0 ±12.6	32.0 ±12.8	32.0 ±12.8	30.0 ±12.6	28.0 ±12.3	30.0 ±12.7
code-run	6.0 ±6.5	4.0 ±5.4	4.0 ±5.4	2.0 ±3.8	0.0 ±0.0	2.0 ±3.8
code-debug	32.0 ±13.0	34.0 ±13.2	36.0 ±13.4	36.0 ±13.4	32.0 ±12.7	32.0 ±12.8
Avg	**43.0** ±2.1	**43.3** ±2.9	**43.8** ±2.6	**42.9** ±2.4	**39.6** ±2.0	**35.9** ±2.6
budget $b = 2048$						
passkey	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0	98.0 ±3.9	100.0 ±0.0	100.0 ±0.0
num-str	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0	92.0 ±7.5	100.0 ±0.0	74.0 ±12.1
kv-retr	28.0 ±12.7	30.0 ±12.7	28.0 ±12.6	36.0 ±13.6	14.0 ±9.7	0.0 ±0.0
dialog-qa	34.0 ±12.9	34.0 ±13.1	38.0 ±13.5	32.0 ±13.0	32.0 ±12.7	26.0 ±12.0
book-sum	27.6 ±2.1	27.6 ±2.1	27.4 ±2.3	23.7 ±2.3	20.1 ±2.0	16.4 ±2.0
book-choice	82.0 ±10.6	84.0 ±10.1	82.0 ±10.6	82.0 ±10.6	76.0 ±11.8	80.0 ±11.1
book-qa-en	16.2 ±5.7	17.9 ±6.8	18.0 ±6.8	20.5 ±7.6	17.5 ±6.8	16.8 ±5.8
book-qa-zh	16.8 ±3.8	17.3 ±4.2	17.8 ±4.2	17.0 ±4.5	20.0 ±5.4	16.0 ±4.9
math-find	30.0 ±12.6	32.0 ±12.8	32.0 ±12.8	30.0 ±12.6	32.0 ±12.8	32.0 ±12.9
code-run	6.0 ±6.5	6.0 ±6.6	4.0 ±5.4	0.0 ±0.0	2.0 ±3.8	2.0 ±3.8

Table B.5 (cont.)

Task	FullKV	exact-LSE	r8i4	RocketKV	ShadowKV	Quest
code-debug	32.0 ±13.0	34.0 ±13.2	32.0 ±13.0	34.0 ±13.2	30.0 ±12.4	30.0 ±12.5
Avg	**43.0** ±2.1	**43.9** ±2.8	**43.6** ±2.6	**42.3** ±2.4	**40.3** ±2.3	**35.7** ±2.6

B.3 LONG-FORM REASONING

B.3.1 BASELINES

Reasoning scores average seeds 0–3. AIME26 and MATH-500 use 38,912- and 16,384-token generation caps, respectively. KVzip and ShadowKV have no reasoning results. RocketKV is omitted: its prompt budget $\mathrm{PB} = \mathrm{tcb} - \mathrm{max_new}$ leaves nearly the full prompt resident during long generation. The reasoning baselines use these settings:

- **R-KV** (Cai et al., 2025). Published reasoning settings: λ=0.1, buffer 128, retain ratio 0.2.
- **TriAttention** (Mao et al., 2026). The HF-reasoning configuration protects the prompt, selects over generated tokens, and reduces query-head scores by the GQA-group max.
- **LazyEviction** (Zhang et al., 2026). We use α=10^{-4} for Qwen3-4B and 5×10^{-4} for Llama. Its published 363-token window at 1492 capacity cannot fit the smaller budgets, so we set $W \approx 0.243b$ throughout the sweep. This scaling is our adaptation. Its per-query-head budget can multiply GQA-group residency by G.

B.3.2 AIME26

AIME26 combines the 2026 I and II exams (30 problems) (Mathematical Association of America, 2026), held after Qwen3's public release.

Table B.6: AIME 2026 I+II accuracy across per-(layer, KV-head) token budgets (30 problems; Qwen3-4B, thinking on, avg@4).

b	FullKV	Oracle	r8i4	r4i4	r2i4	Quest	R-KV	TriAtt	LazyEvict
64	59.2	0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.0
128	59.2	28.3	25.8	26.7	25.8	0.0	0.0	0.0	0.0
256	59.2	52.5	51.7	55.0	54.2	0.0	0.0	0.0	0.0
512	59.2	62.5	60.8	64.2	57.5	0.0	0.0	3.3	1.7
1024	59.2	62.5	65.0	64.2	65.0	5.8	4.2	22.5	6.7
2048	59.2	59.2	63.3	65.8	60.8	33.3	24.2	45.0	34.2

B.3.3 MATH-500

We use a 16,384-token generation cap, following the reasoning-model protocol of R-KV (Cai et al., 2025).

Table B.7: MATH-500 accuracy across per-(layer, KV-head) token budgets (Qwen3-4B, thinking on, avg@4, 16,384-token generation cap).

b	FullKV	Oracle	r8i4	r4i4	r2i4	Quest	R-KV	TriAtt	LazyEvict
64	94.0	56.0	54.5	55.0	53.0	1.5	5.0	11.5	0.5
128	94.0	78.0	80.5	81.5	82.5	3.0	8.0	20.5	1.5
256	94.0	92.0	91.0	90.5	90.5	9.0	31.5	44.5	19.5
512	94.0	94.0	92.0	92.0	92.5	43.5	60.0	66.0	45.0
1024	94.0	94.0	94.0	91.5	93.5	68.5	77.0	81.5	75.0
2048	94.0	93.5	93.5	93.5	93.5	87.5	90.5	91.5	92.0

B.4 GH200 DECODE EFFICIENCY

B.4.1 GPU-RESIDENT BATCH-1 DECODE

Figure 4(a–c) uses one GH200 with GLM-4-9B-Chat-1M, TP$=1$, bf16 weights and GPU-resident original KV, B=16, full CUDA-graph decode, 8192-token prefill chunks, and no prefix cache. Runs use greedy generation with EOS ignored. Each 16K–512K short run emits 100 tokens. TTFT includes LOCKS summary or Quest envelope construction; mean ITL averages the 99 subsequent `engine.step()` calls. Table B.8 gives the values behind panels (a,b).

Quest is our optimized GPU port. It selects up to 2048 tokens per query head and keeps two of its 40 layers dense. Quest-U adapts the authors' upstream CUDA pipeline to bf16/GQA and vLLM, retaining the same per-query-head budget and dense layers.

Table B.8: GH200 batch-1 context sweep with GPU-resident bf16 KV. Quest is the optimized port plotted in Fig. 4; Quest-U adapts the upstream CUDA pipeline. Build-inclusive TTFT averages three prefills except Quest-U at 512K (one); mean ITL averages 99 complete decode steps.

	TTFT (s)					Mean ITL (ms)				
Context	FA3	FI	Quest	Quest-U	r4i4	FA3	FI	Quest	Quest-U	r4i4
16K	0.61	0.63	0.61	0.62	0.65	7.08	7.07	7.54	7.85	7.05
32K	1.52	1.60	1.53	1.53	1.60	7.49	7.43	7.69	8.66	7.20
64K	4.28	4.53	4.28	4.28	4.41	8.21	8.14	7.82	10.23	7.55
128K	13.47	14.47	13.49	13.50	13.82	9.68	9.56	8.11	13.38	8.26
256K	47.47	50.44	47.53	47.54	48.22	12.59	12.42	9.13	19.65	9.01
512K	178.62	186.53	178.53	178.45	179.96	18.42	18.03	10.74	32.17	10.11

Panel (c) follows one continuous 32K-prefill, 128K-generation request per method with resident KV. The plotted ITL averages `engine.step()` time over 10 tokens; the final 11-step window includes request completion. Table B.9 includes all post-first-token intervals.

Table B.9: GH200 long generation: 32K input and 128K new tokens with GPU-resident KV. All 131,071 decode intervals are included.

Method	TTFT (s)	Mean ITL (ms)	Decode wall (s)
FA3	1.52	8.89	1167.4
FlashInfer	1.59	8.81	1157.2
Quest	1.53	8.04	1056.9
LOCKS r4i4	1.62	8.10	1064.5

Figure 4(d) pairs the r4i4, Quest, and FullKV AIME26 avg@4 scores in Table B.6 with separate Qwen3-4B latency runs. Each timing run has a 100-token input and 38,912 generated tokens; mean ITL averages the 38,911 complete decode steps.

B.4.2 HOST-BACKED BATCHED DECODE

Table 3 uses the same GH200 and model. Dense KV stays in GPU HBM. LOCKS r4i4 keeps full KV in Grace memory and frequently used pages in a smaller GPU cache, leaving room for more simultaneous requests. Each row reports throughput at its shown batch, so the speedup measures serving capacity even when LOCKS takes longer per decode step. Timings cover 192 decode steps without cache logging. Hot-page shares come from separate runs that generated the same tokens, so counting cache accesses did not slow the reported throughput.

B.5 KV-CACHE QUANTIZATION (KIVI): FULL GRID

Table B.10 gives the full KIVI (Liu et al., 2024) grid: 4- and 2-bit KV, both summary-building orders, all budgets and ranks on RULER-32K (13 tasks).

Table B.10: RULER-32K average across KV-cache precision, summary rank, and budget (13 tasks, Llama-3.1-8B). POST builds summaries from bf16 keys; PRE builds them from quantized keys.

			4-bit		2-bit	
b	rank	bf16	POST	PRE	POST	PRE
64	r=8	73.1	72.8	73.0	68.0	67.6
	r=4	68.5	68.6	68.8	64.9	63.7
	r=2	66.2	65.4	65.9	60.0	60.5
128	r=8	83.4	83.3	83.7	78.7	76.7
	r=4	80.1	78.1	79.4	75.6	74.8
	r=2	77.7	76.1	76.5	72.7	72.8
256	r=8	85.5	85.2	85.2	81.9	80.7
	r=4	84.9	83.8	84.4	80.4	79.0
	r=2	83.2	82.7	82.2	78.2	76.5
512	r=8	86.1	86.2	85.9	83.4	82.8
	r=4	86.0	85.7	85.7	83.7	82.7
	r=2	86.0	85.6	85.3	81.6	80.7
1024	r=8	87.2	87.3	87.1	84.9	85.3
	r=4	86.9	86.6	86.5	83.8	84.3
	r=2	86.9	86.6	86.5	84.0	83.2
2048	r=8	88.0	87.6	87.4	85.0	85.3
	r=4	87.7	87.2	87.6	84.5	84.9
	r=2	87.6	87.5	87.8	84.8	84.3

C Extended Studies

Unless noted, experimental settings follow App. B.

C.1 GQA

With exact per-head scores, share-average leads the other shared-page combine rules on all measures in Table C.1.

Table C.1: Per-query-head coverage under GQA combine rules at b=2048 using exact per-head scores (Qwen3-4B, G=4). Mean reports group-average coverage; worst head uses the minimum across a group's query heads. Both summarize evaluated traces; bold marks LOCKS's rule.

	reasoning		retrieval	
Combine rule	mean	worst head	mean	worst head
Share-average	**0.872**	**0.785**	**0.879**	**0.787**
Group-max	0.850	0.748	0.860	0.757
Group-mass	0.856	0.756	0.866	0.763
Per-head oracle (up to $G\times$ traffic/work)	0.893	0.812	0.898	0.812
FullKV	1.000	1.000	1.000	1.000

C.2 Rotary Position Embedding

Post-RoPE fits, used by LOCKS, give similar mass and recall to pre-RoPE fits (Table C.2).

Table C.2: RoPE placement for page-local rank-8 LOCKS summaries on RULER-32K (B=16, b=1024). Entries report retained mass and exact-LSE page recall.

	Llama-3.1-8B		Qwen3-4B	
RoPE fit	mass	recall	mass	recall
pre-RoPE	0.8936	0.9461	0.8770	0.9362
post-RoPE	0.8935	0.9365	0.8767	0.9306

C.3 Page size

At B=16, rank 8 captures a median 89% of within-page key variance on both models, versus 67.8% for size-matched isotropic pages. Table C.3 extends Table 5 across budgets and page sizes.

Table C.3: LongBench-v1 and RULER accuracy across token budget and page size (Llama-3.1-8B, rank-8 int4 basis). FullKV is independent of budget and page size.

	LongBench-v1			RULER-16K			RULER-32K		
b	B=16	B=32	B=64	B=16	B=32	B=64	B=16	B=32	B=64
FullKV		47.0			94.3			88.8	
256	46.7	46.8	46.8	90.6	89.7	89.0	85.5	82.7	83.2
512	47.1	46.9	47.0	91.4	91.3	91.5	86.1	83.8	83.7
1024	47.2	47.1	47.1	93.0	92.1	92.7	87.2	83.5	84.8
2048	47.2	47.1	47.0	93.7	93.0	93.6	88.0	84.5	85.1

C.4 MIXTURE-OF-EXPERTS: ERNIE-4.5-21B-A3B-THINKING

Table C.4 extends the reasoning study to ERNIE-4.5-21B-A3B-Thinking (Baidu ERNIE Team, 2025), a mixture-of-experts model with 21B total and ∼3B active parameters.

Table C.4: Accuracy on AIME26 and the first 50 MATH-500 questions across token budgets for ERNIE-4.5-21B-A3B-Thinking. Seed 0; exact-LSE and LOCKS ranks 8, 4, and 2 with int4 bases.

AIME26 (FullKV: 66.7)

		LOCKS		
b	Oracle	r8i4	r4i4	r2i4
64	0.0	0.0	0.0	0.0
128	33.3	23.3	36.7	30.0
256	46.7	56.7	50.0	50.0
512	63.3	63.3	56.7	56.7
1024	60.0	60.0	66.7	63.3
2048	60.0	66.7	66.7	66.7

MATH-500 (FullKV: 90.0)

		LOCKS		
b	Oracle	r8i4	r4i4	r2i4
64	46.0	48.0	54.0	48.0
128	78.0	72.0	76.0	76.0
256	82.0	88.0	90.0	88.0
512	90.0	88.0	90.0	90.0
1024	92.0	90.0	88.0	88.0
2048	86.0	88.0	92.0	86.0

C.5 LARGE DENSE: DEEPSEEK-R1-DISTILL-QWEN-32B

Table C.5 repeats the reasoning study on the 32B dense DeepSeek-R1-Distill-Qwen-32B (DeepSeek-AI, 2025), using the main B=16 page size.

Table C.5: Accuracy on AIME26 and the first 50 MATH-500 questions across token budgets for DeepSeek-R1-Distill-Qwen-32B. Seed 0; exact-LSE and LOCKS ranks 8, 4, and 2 with int4 bases.

AIME26 (FullKV: 53.3)

		LOCKS		
b	Oracle	r8i4	r4i4	r2i4
64	6.7	16.7	23.3	16.7
128	50.0	53.3	50.0	43.3
256	46.7	56.7	50.0	56.7
512	70.0	46.7	56.7	66.7
1024	60.0	60.0	60.0	53.3
2048	63.3	60.0	63.3	56.7

MATH-500 (FullKV: 94.0)

		LOCKS		
b	Oracle	r8i4	r4i4	r2i4
64	76.0	82.0	80.0	76.0
128	94.0	90.0	88.0	92.0
256	94.0	92.0	94.0	94.0
512	92.0	94.0	90.0	92.0
1024	94.0	94.0	94.0	94.0
2048	90.0	94.0	94.0	94.0

C.6 LEARNED SPARSE ATTENTION: QWEN3.8-FLASH-NEXT

In Qwen3.8-Flash-Next (Qiu et al., 2026), we replace decode-time selection in its 12 learned-QSA layers; prefill and Gated DeltaNet layers remain unchanged. Methods share the FP8 checkpoint, bf16 KV cache, H200 TP2 setup, prompts, and sampling seeds.

Table C.6: Qwen3.8-Flash-Next-FP8 accuracy across token budgets on AIME26 and the first 50 MATH-500 questions (one sample per question).

	AIME26				MATH-500			
Method	b=64	128	256	512	b=64	128	256	512
QSA (native)	66.7	76.7	80.0	86.7	100.0	98.0	94.0	96.0
LOCKS r1i4	53.3	80.0	86.7	86.7	96.0	96.0	96.0	96.0
Quest	20.0	46.7	80.0	76.7	84.0	96.0	96.0	96.0

QSA may append three tail tokens, while LOCKS (B=4) and Quest count every selected token within b.

C.7 H200 DECODE EFFICIENCY: MEASUREMENT PROTOCOL AND FULL NUMBERS

For this GPU-resident byte model, a bf16 KV page at B=16 and d=128 holds 8192 B. Including scales, rank-8, 4, and 2 summaries use 818, 490, and 326 B/page, giving summary fractions $\alpha =$ 0.100, 0.060, and 0.040. Modeled per-step HBM reads are $\alpha N + b$ KV-token equivalents.

Configuration. We use GLM-4-9B-Chat-1M on one H200 NVL (TP= 1, bf16, B=16, b=2048). Latency comparisons are among vLLM decode backends: LOCKS r8i4 and the dense FlashAttention-3 (FA3) and FlashInfer controls. Only runs with active r8i4 scoring on every decode step are included; prefill uses FA3.

Measurement. After warm-up, clock stabilization, and CUDA-graph capture, one prefill feeds three 64-step decode windows. We retain lockstep windows without NVML-detected throttling, excluding end-of-sequence effects. TTFT uses three repeated prefills; its build-inclusive mode adds the one-time summary build. Table C.7 reports TPOT mean ± sd across 192 steps and build mean ± sd across 8–10 boots.

Table C.7: Decode efficiency from 16K to 1M context at batch 1, TP= 1 (GLM-4-9B-Chat-1M, one H200 NVL). TPOT reports mean ± sd over 192 steps; build reports mean ± sd over 8–10 boots.

	TPOT (ms)				steady TTFT (s)		build
ctx	FA3	FI	r8i4	speedup	FA3	r8i4	(ms)
16K	6.93 ± 0.05	7.20 ± 0.05	**7.08** ± 0.21	0.98×	0.63	0.63	547 ± 349
32K	7.33 ± 0.03	6.91 ± 0.04	**7.24** ± 0.21	0.96×	1.56	1.57	810 ± 525
64K	7.97 ± 0.08	8.06 ± 0.04	**7.38** ± 0.20	1.08×	4.42	4.41	916 ± 395
128K	8.98 ± 0.05	8.74 ± 0.05	**8.27** ± 0.22	1.06×	13.9	14.1	2904 ± 2282
256K	11.63 ± 0.09	11.89 ± 0.08	**8.96** ± 0.23	1.30×	49.6	49.7	1323 ± 317
512K	17.24 ± 0.18	16.17 ± 0.18	**10.47** ± 0.28	1.54×	184.1	184.2	1993 ± 253
1M	27.50 ± 0.40	26.19 ± 0.45	**12.95** ± 0.51	2.02×	699.0	699.4	3717 ± 756

Batched decode (Table C.8). For each backend and context, we sweep batch size on one live batch without re-prefill, applying the same window checks. Speedup is against the faster dense backend; batch-1 values reuse Table C.7. The frontier is the largest batch fitting one H200 NVL at 90% utilization; larger combinations are marked OOM.

Table C.8: Batched decode with GLM-4-9B-Chat-1M on one H200 NVL (r8i4, b=2048). Cells give LOCKS per-token speedup over the faster of FA3 and FlashInfer at matched context and batch, measured in three 64-step windows on one live batch. Bold marks the largest batch feasible for both arms at 90% GPU memory utilization; the final column gives LOCKS aggregate throughput there.

	batch size						frontier
context	1	2	4	8	16	32	LOCKS (tok/s)
256K	1.30	1.43	**1.80**	OOM	OOM	OOM	275
128K	1.06	1.21	1.44	**1.59**	OOM	OOM	483
64K	1.08	1.08	1.17	1.33	**1.43**	OOM	872
32K	0.96	1.01	1.02	1.07	1.14	**1.26**	1503