EarlySelectKV: Anticipatory KV Selection via Cross-Layer Query Proxies

Anonymous ACL submission

Abstract

Query-conditioned sparse KV selection reduces long-context inference cost by attending to a small subset of historical KV entries at each decoding step. However, these selectors typically wait for the target-layer query before scoring candidates and generating selected indices, leaving routing-dependent work on the immediate pre-attention path. We introduce EARLYSELECTKV, a training-free transformation that uses proxy queries from earlier hidden states to precompute selected KV indices, while final attention still uses the real target-layer query.

We instantiate EarlySelectKV on representative query-conditioned selectors, with RocketKV-HSA as the primary setting. Proxy-fidelity analysis shows that early query anchors preserve the routing evidence needed by the selector, and that layer-wise anchor choices balance preparation window and routing stability. Across LongBench, NIAH/Passkey, and RULER, ES-RocketKV preserves native RocketKV performance, differing by only $+0.018$ points on LongBench, $+0.25$ percentage points on NIAH/Passkey, and approximately -0.01 points on RULER. In a CPU-offloaded TensorRT-LLM deployment, early index availability moves routing and metadata preparation off the GPU attention path, improving end-to-end output throughput by up to 12.4% and reducing GPU-resident K-summary state by 11.1% of the KV-cache pool. These results highlight routing timing as an additional optimization dimension for sparse long-context decoding.

1 Introduction

Long-context inference in large language models is increasingly constrained by the memory capacity and bandwidth required to store and access historical key-value (KV) states (Kwon et al., 2023). Since the KV cache grows linearly with context length, batch size, and model depth, this cost becomes a major bottleneck in long-document understanding, repository-level code reasoning, extended dialogue, and tool-using workflows (Liu et al., 2024).

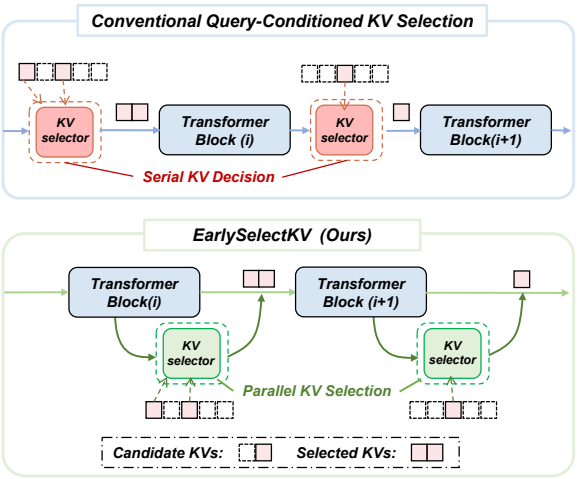


Figure 1: Conventional query-conditioned KV selection and EARLYSELECTKV. Conventional selectors wait for the target-layer query and place index generation on the immediate pre-attention path; EarlySelectKV derives a proxy query from preceding hidden states and precomputes KV indices in an earlier computation window.

A common strategy is to reduce the number of historical KV entries read by attention. Eviction and compression methods maintain a compact cache by retaining recent tokens, attention sinks, or tokens estimated to be important from prompt-time observations (Xiao et al., 2024; Zhang et al., 2023; Li et al., 2024). Dynamic sparse methods instead keep a candidate cache and select a small KV subset at each decoding step according to the current query (Tang et al., 2024; Choi and Cho, 2025; Singhania et al., 2024). This query-conditioned interface adapts to changing evidence needs during generation, but its cost is not determined only by the number of selected tokens.

As shown in Figure 1, conventional query-conditioned selectors must obtain the target-layer query before scoring candidate pages or tokens, producing selected indices, and preparing the corresponding KV entries. This creates a serial routing dependency before sparse attention. As the sparse budget decreases, attention computation is reduced, whereas routing computation and metadata access remain exposed on the same critical path. Thus, after attention has been sparsified, deciding which KV entries to access can become a visible source of latency.

This exposes a different optimization target: not which KV entries to keep, but when their indices can be determined. We observe that routing and value aggregation require different levels of query fidelity. The exact target-layer query is needed for attention over the selected KV states, whereas routing only needs to preserve the relative preference among candidates.

We propose EarlySelectKV, a training-free transformation for query-conditioned sparse KV selectors. It advances index generation without changing the sparse attention computation itself: the proxy query is used only for routing, while the real query still performs attention over the original selected KV states. We integrate EarlySelectKV with representative query-conditioned sparse KV selectors, including RocketKV, Quest, and Loki, demonstrating its plug-and-play applicability across different routing designs. Rather than redesigning the selector, EarlySelectKV advances the availability of selected KV indices, making query-conditioned sparsity more amenable to system-level scheduling.

As a representative case, ES-RocketKV matches RocketKV on LongBench (+0.018), Needle-in-a-Haystack (+0.25%), and RULER (-0.01), indicating negligible accuracy degradation. This parity makes selected indices a reliable scheduling signal. In a CPU-offloaded deployment, the early availability of selected indices allows routing and metadata preparation to be staged before the target attention step, reducing exposed stalls on the GPU attention path and lowering GPU-resident auxiliary state.

Our contributions are summarized as follows:

- We propose EarlySelectKV, a training-free anticipatory KV cache selection framework that uses query anchors generated from preceding hidden states to reduce the routing overhead exposed on the critical path of dynamic sparse decoding.
- EarlySelectKV is compatible with existing query-conditioned selectors. It preserves their original candidate construction and scoring rules, while supporting flexible anchor choices for different lookahead windows and query-projection approximation strategies.
- Experiments on long-context benchmarks, including LongBench and Needle-in-a-Haystack, show that EarlySelectKV maintains task performance and selection fidelity while reducing the visible routing overhead of exact-query routing.

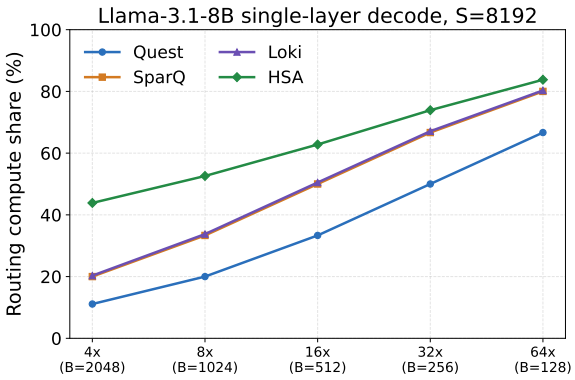


Figure 2: Theoretical routing overhead under sparse attention budgets. We analyze a single Llama-3.1-8B layer with an 8K KV cache and vary the selected-token budget. As the budget becomes smaller, sparse attention computation decreases, while routing becomes a larger fraction of the remaining attention-side computation.

2 Related Work

Architecture-level KV-cache reduction. Architecture-level approaches reduce KV-cache cost by modifying how attention represents or reuses key-value states. MQA (Shazeer, 2019), GQA (Ainslie et al., 2023), and MLA (DeepSeek-AI et al., 2024) share or compress KV representations across query heads, while other work reuses KV states across layers (Brandon et al., 2024), skips selected attention layers (Jiang et al., 2024), or combines local and global attention patterns (Sun et al., 2024). These methods provide efficient KV-cache usage by design, but are tightly coupled with model architecture and often require pretraining-time integration or architecture-specific deployment.

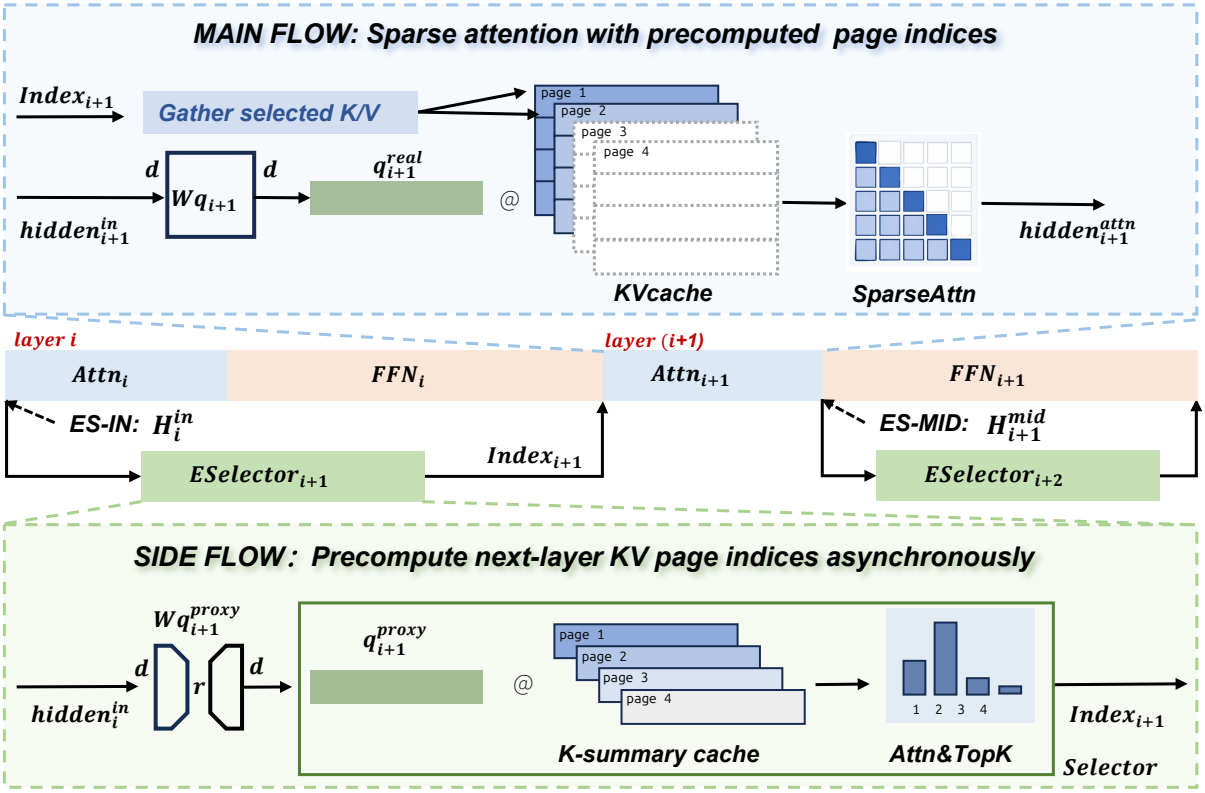


Figure 3: Overview of EarlySelectKV. The main flow gathers selected KV using precomputed page indices and performs sparse attention with the real query. The side flow constructs a proxy query from an earlier hidden state and precomputes next-layer page indices. Early and Mid taps represent two timing–fidelity trade-offs: the former offers a longer preparation window, while the latter provides higher selection fidelity.

Inference-time KV-cache pruning. Inference-time pruning methods reduce memory by retaining a compact token history under a fixed cache budget. H2O (Zhang et al., 2023) keeps recent and heavy-hitter tokens, StreamingLLM (Xiao et al., 2024) preserves attention sinks and a recent window, and prompt-stage methods such as SnapKV (Li et al., 2024) and LAQ (Wang et al., 2025) select important prompt KV entries. These methods are typically training-free and effective at reducing cache footprint, but their retained history is largely fixed after pruning and provides limited per-step adaptivity during decoding.

Query-conditioned KV routing. Query-conditioned routing methods keep a candidate cache and dynamically select a small KV subset at each decoding step according to the current query. Quest (Tang et al., 2024) scores page-level representatives, while SparQ and Loki (Choi and Cho, 2025; Singhania et al., 2024), ScoutAttention and KVPR (Zhang et al., 2026; Jiang et al., 2025), and MagicPIG (Chen et al., 2024) reduce routing cost through channel sparsity, offloading, or hashing-based approximation. RocketKV (Behnam et al., 2025) combines prompt-stage compression with Hybrid Sparse Attention through coarse-to-fine query-conditioned routing. These methods preserve per-token adaptivity and can maintain accuracy under small KV budgets, but routing still depends on target-layer queries, leaving scoring and index generation on the immediate pre-attention path.

3 Method

3.1 Problem Setup

Long-context sparse decoding reduces inference cost by limiting the KV entries accessed by attention, while query-conditioned sparse selectors introduce an additional routing step before sparse attention. This step accesses auxiliary routing states, such as page metadata or summary caches, and performs index-generation operations such as page scoring or top-k selection. Although these operations do not participate in value aggregation, they gate sparse attention by producing the selected KV indices, placing routing-dependent work on the im-

mediate pre-attention path.

Figure 2 illustrates the attention-side impact of this routing cost. As the selected-token budget becomes smaller, sparse attention itself becomes cheaper, while routing does not necessarily shrink at the same rate. Consequently, routing occupies a larger fraction of the remaining attention-side computation. This suggests that, after attention has been sparsified, the cost of deciding which KV entries to access can become a non-negligible component of dynamic sparse attention.

Existing work mainly addresses this issue by making routing cheaper or more accurate. In contrast, we focus on when routing is performed. Exact-query routing must wait until the target-layer query is produced, so the selected KV indices become available only on the immediate pre-attention path. This motivates making the indices available earlier while preserving the behavior of the original selector.

3.2 Overview

EarlySelectKV is an anticipatory routing framework for query-conditioned sparse KV selectors. As shown in Figure 3, EarlySelectKV decomposes dynamic KV selection into a main flow and a side flow. The main flow performs sparse attention at the target layer: it reads precomputed page indices, gathers the corresponding selected KV entries, and computes attention with the real query of the target layer. The side flow runs before the main flow reaches the target attention layer. It constructs a proxy query from a preceding hidden state, performs lightweight selection over a K-summary cache or another routing view, and precomputes the page indices required by the next layer. This design moves page-index generation out of the immediate dependency before target attention.

Given the real query $q^{\ell}_{t,\mathrm{real}}$ of layer ℓ at decoding step t, a query-conditioned sparse selector can be abstracted as

$$\mathcal{I}^{\ell}_{t} = \mathcal{R}_{\ell}\left(q^{\ell}_{t,\mathrm{real}}, \mathcal{M}^{\ell}_{t}; B\right), \tag{1}$$

where $\mathcal{R}_{\ell}$ denotes the underlying routing rule, $\mathcal{M}^{\ell}_{t}$ is the routing view used by the selector, and B is the routing budget. The routing view $\mathcal{M}^{\ell}_{t}$ abstracts the selector's searchable state, ranging from the full KV cache to compact proxies such as candidate caches or summaries. In standard execution, $\mathcal{I}^{\ell}_{t}$ cannot be produced until $q^{\ell}_{t,\mathrm{real}}$ becomes available; consequently, KV gathering and sparse attention are both blocked by this routing step.

EarlySelectKV replaces the real query in Equation 1 with a proxy query constructed earlier:

$$\widehat{\mathcal{I}}^{\ell}_{t} = \mathcal{R}_{\ell}\left(q^{\ell}_{t,\mathrm{proxy}}, \mathcal{M}^{\ell}_{t}; B\right). \tag{2}$$

This transformation modifies only the query signal that drives routing, allowing the required page indices to be prepared before attention consumes the corresponding KV entries. It preserves the selector structure, and empirically maintains accuracy in our evaluated settings.

3.3 Proxy Query Settings

For any existing query-conditioned selector, EarlySelectKV wraps it into an anticipatory selection module, which we call ESelector. Given a hidden state $\tilde{H}_{t,\ell}$ that is available before the target attention layer, ESelector first constructs a proxy query:

$$q^{\ell}_{t,\mathrm{proxy}} = W^{\ell}_{q,\mathrm{proxy}}\,\mathrm{LN}_{\ell}(\tilde{H}_{t,\ell}), \tag{3}$$

and then invokes the original selector to produce the precomputed indices. Here, $W^{\ell}_{q,\mathrm{proxy}}$ can be the native query projection W^{ℓ}_{q} of the target layer or its low-rank approximation. The native projection gives a closer match to the target-layer query space, whereas the low-rank variant lowers the cost of proxy-query construction. Both choices are training-free. Appendix A further analyzes how the approximation rank trades proxy-query construction cost for task performance.

The key configuration of ESelector is the choice of $\tilde{H}_{t,\ell}$. Anchor placement determines the trade-off between preparation window and selection fidelity. We study two fixed anchors, ES-In and ES-Mid, and use offline layer-sensitivity profiling to derive a model-specific layer-wise assignment between them.

ES-In. ES-In feeds ESelector with an earlier block input. For example, H^{in}_{i} can be used to construct the proxy query for layer $i+1$ and precompute Index_{i+1}. Since this anchor is available before layer i is computed, ES-In provides a longer preparation window for routing, metadata lookup, and KV preparation. It is suitable for layers that tolerate earlier proxy queries, especially when latency hiding is the main goal.

ES-Mid. ES-Mid uses a later intermediate hidden state as the ESelector input. For a pre-LN Transformer block, the post-attention state is

$$H^{\mathrm{mid}}_{\ell} = H^{\mathrm{in}}_{\ell} + \mathrm{Attn}_{\ell}\left(\mathrm{LN}_{\ell}(H^{\mathrm{in}}_{\ell}), K^{\ell}, V^{\ell}\right), \tag{4}$$

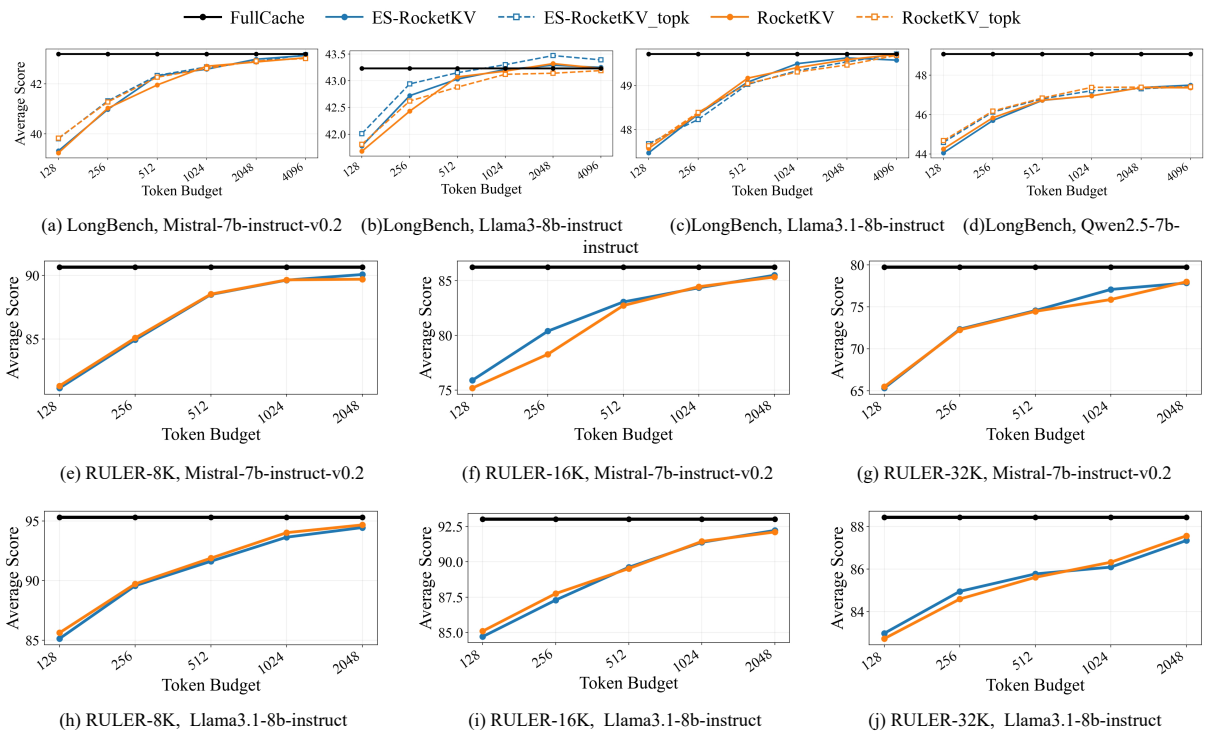


Figure 4: Task score comparison between ES-RocketKV and RocketKV on LongBench and RULER. The first row reports LongBench results, while the second and third rows report RULER results under different context lengths.

which already incorporates attention-level contextual information through the residual connection. As a result, ES-Mid provides a stronger proxy-query representation than ES-In, but leaves a shorter preparation window. It is preferable for layers that are more sensitive to proxy-query approximation.

ES-Mix. ES-Mix uses offline layer-sensitivity profiling to choose between ES-In and ES-Mid for each layer. Insensitive layers use ES-In to maximize the preparation window, whereas sensitive layers use ES-Mid to preserve selector fidelity. The resulting assignment is fixed during inference and balances routing earliness with selection stability without changing the underlying selector.

The actual behavior of ES-In, ES-Mid, and ES-Mix depends on the model and layer structure; we analyze this empirically in Section 4.4.

3.4 ESelector Instantiations

EarlySelectKV can wrap query-conditioned selectors whose routing decisions are driven by a query signal. We instantiate it on Quest, Loki, and RocketKV, with RocketKV serving as the primary experimental setting. RocketKV is built on Hybrid Sparse Attention (HSA), which partitions the historical KV cache into sequence pages and represents each page with lightweight summaries, such as $K_{\max}$ and $K_{\min}$, for query-conditioned page scoring. For all instantiations, EarlySelectKV preserves the original routing view, budget, and scoring rule, and replaces only the routing query with the proxy query. We evaluate the additional Quest and Loki instantiations in Section 4.3.

4 Experiments

4.1 Experimental Setup

We evaluate EarlySelectKV on four long-context language models: Llama-3.1-8B-Instruct, Llama-3-8B-Instruct (Grattafiori et al., 2024), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), and Qwen2.5-7B-Instruct (Qwen et al., 2025). The evaluation covers LongBench (Bai et al., 2024), Needle-in-a-Haystack (NIAH) (Kamradt, 2023), and RULER (Hsieh et al., 2024), testing the method on general long-context understanding, long-range retrieval, and synthetic long-context reasoning tasks.

Our main comparison is between RocketKV and ES-RocketKV. They differ only in the routing-query source: RocketKV uses the target-layer real query, whereas ES-RocketKV uses a proxy query from a preceding hidden state. We further report FullCache as the dense-attention reference, which attends to the entire KV cache without sparse

Budget	Method		
	TopK	RocketKV	ES-RocketKV
Mistral-7B-v0.2-Instruct			
64	100.0	**100.0**	**100.0**
128	100.0	**100.0**	**100.0**
Llama3-8B-Instruct			
64	100.0	**98.0**	**98.0**
128	100.0	**100.0**	**100.0**
Llama3.1-8B-Instruct			
64	100.0	**88.0**	**88.0**
128	100.0	**100.0**	**100.0**
Qwen2.5-7B-Instruct			
64	76.0	**48.0**	**48.0**
128	76.0	64.0	**66.0**

Table 1: Matching accuracy (%) on NIAH/Passkey under low KV-token budgets. Across four instruction-tuned backbones, ES-RocketKV closely matches RocketKV, showing negligible degradation in passkey retrieval accuracy.

selection. Unless otherwise stated, all ESelector variants use ES-Mid as the default anchor for stronger proxy-query representations. For stability, all sparse methods keep layer 0 dense and apply sparse selection from layer 1 onward. Since ESelector precomputes indices before the target-layer keys are available, we reserve the local budget outside routing and keep the most recent tail page by default. ESelector routes only over the remaining historical cache.

4.2 Benchmark Performance

For all evaluation tasks, we vary the token budget from a low-budget setting of 128 to 4096. We report average benchmark scores for LongBench and accuracy for NIAH and RULER. Detailed per-task results are reported in Appendix B.

LongBench Benchmark. We compare FullCache, RocketKV, ES-RocketKV, and the corresponding TopK variants on LongBench. The TopK variants use the same sparse path and budget, but replace routing-based candidates with exact top-k attention-mass candidates, serving as an oracle-style attention-mass reference under the same sparse budget. The first row of Figure 4 summarizes the LongBench averages; detailed subtask scores and average tables are provided in Appendix B.

ES-RocketKV closely matches RocketKV across all evaluated models and budgets. Since the two methods differ only in the routing query, this shows that the KV selection needed by RocketKV can be reliably approximated from an early proxy query. Averaged over all LongBench settings, ES-RocketKV differs from RocketKV by only 0.018 points; even at the most restrictive 128-token budget, the average gap is only 0.03 points. Across models, the gap remains below 0.1 points, indicating negligible quality loss from early routing.

As the token budget increases, the gap between sparse methods and FullCache decreases, suggesting that LongBench degradation mainly comes from the sparse budget rather than the proxy-query approximation. TopK variants are not consistently better than routing-based variants, indicating that preserving the highest-attention tokens is not always the best predictor of task score on generation-heavy long-context tasks.

NIAH Benchmark. Table 1 reports representative NIAH passkey retrieval results; the full budget sweep and detailed setup are provided in Appendix B.2. Across the full evaluation, ES-RocketKV closely matches RocketKV, with an average gap of only about 0.25 percentage points. The results are identical on the Llama and Mistral models and remain comparable on the more budget-sensitive Qwen2.5-7B-Instruct, suggesting that proxy-query routing preserves the evidence-localization signal needed for long-context retrieval.

RULER Benchmark. We further evaluate ES-RocketKV on RULER-8K, RULER-16K, and RULER-32K, as shown in the second and third rows of Figure 4; detailed scores are provided in Appendix B. RULER covers retrieval, variable tracking, word-frequency counting, and question answering, making it a broader test of sparse KV selection under long-context reasoning. Across all RULER settings, ES-RocketKV closely matches RocketKV with an overall gap of approximately -0.01 points, showing that early proxy-query routing introduces no systematic accuracy loss.

Both sparse methods move farther from FullCache as context length increases, since limited-budget KV selection becomes harder on longer contexts. However, ES-RocketKV follows nearly the same budget-scaling trend as RocketKV, indicating that it preserves RocketKV's accuracy–budget behavior while moving routing earlier.

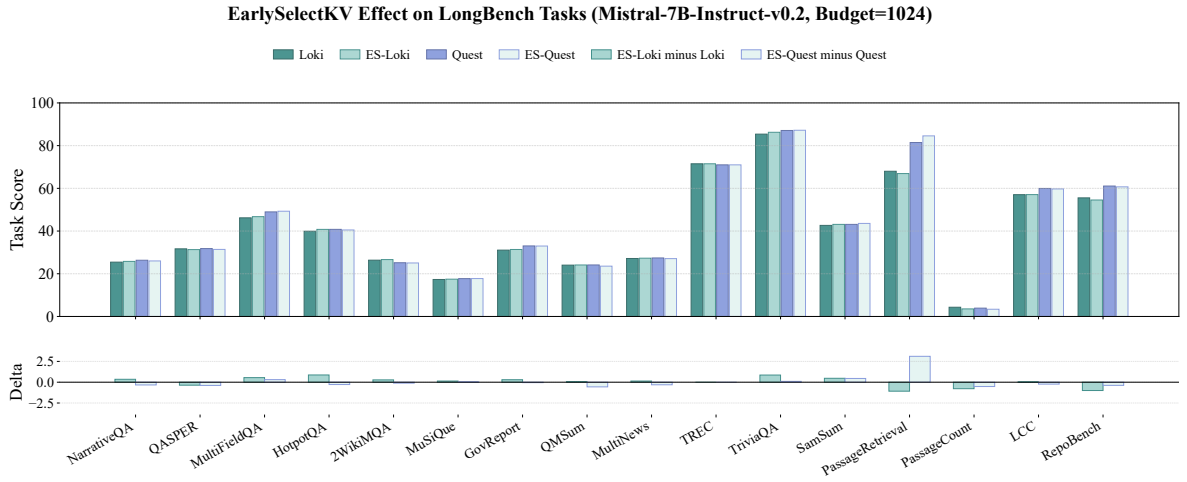


Figure 5: Additional selector results on LongBench using Mistral-7B-Instruct-v0.2 with a token budget of 1024. The upper panel reports task-level scores, and the lower panel shows the score difference between each ES variant and its corresponding base selector.

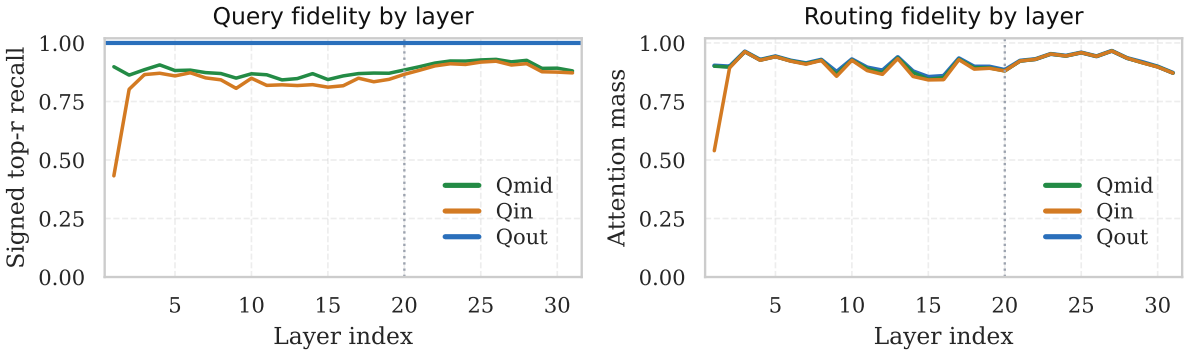


Figure 6: Proxy-query fidelity across decoder layers. Left: signed top-r channel recall of Q_{in} and Q_{mid} with respect to Q_{out}. Right: attention mass retained by tokens selected using each query, evaluated under the Q_{out} attention distribution. The dotted line marks the transition layer used by ES-Mix.

4.3 Applicability to Other Selectors

To test whether EarlySelectKV is specific to RocketKV, we further apply the same selector-preserving transformation to Quest and Loki. Quest uses query-conditioned page-level routing, while Loki relies on lightweight query-dependent scoring.

Figure 5 reports LongBench results on Mistral-7B-Instruct-v0.2 with a token budget of 1024. The ES variants closely match their corresponding base selectors, indicating that early proxy-query routing is not tied to HSA page summaries or RocketKV's cache-reduction pipeline.

4.4 Proxy Query Fidelity

The benchmark results show that ES-RocketKV closely matches native RocketKV at the task level. We next examine whether proxy queries preserve the routing evidence used by the native selector. Let Q_{out} denote the native target-layer routing query, and let Q_{in} and Q_{mid} denote proxy queries constructed from the block input and the post-attention intermediate state. Since EarlySelectKV only changes the routing query, proxy fidelity should be evaluated by routing behavior rather than exact query reconstruction.

We use two diagnostics: signed top-r channel recall, which measures agreement on the dominant signed routing dimensions of Q_{out}, and retained attention mass, which measures how much Q_{out}-based attention mass is covered by proxy-selected tokens. The diagnostic setup is described in Appendix C.

Figure 6 shows a stable layer-wise pattern. In early and middle layers, Q_{mid} provides a stronger

routing signal than Q_{in}, with higher signed-channel recall and retained attention mass. In later layers, the gap between the two anchors becomes negligible. This pattern motivates ES-Mix, a profile-guided layer-wise assignment that uses ES-Mid for anchor-sensitive layers and ES-In for layers where earlier anchors preserve stable routing behavior. The assignment is computed offline and fixed during inference. Appendix C further studies calibration-distribution sensitivity, showing that the absolute routing fidelity changes across probes but the layer-wise anchor preference remains stable.

4.5 Runtime Benefits

Deployment setup. We run runtime experiments on a single NVIDIA RTX 6000D GPU using TensorRT-LLM (NVIDIA, 2023) with the PyTorch backend and CUDA Graphs enabled. Both RocketKV and ES-RocketKV use the same sparse attention configuration and FP8 K-summary caches. ES-RocketKV offloads routing to an Intel Xeon Platinum host CPU, where proxy-query construction and routing are executed in BF16 with AVX-512 BF16 dot-product instructions and FP32 accumulation. To reduce CPU-side overhead, we use a rank-256 SVD approximation of the query projection. Because ES-RocketKV changes both when routing indices are produced and where routing work is executed, the runtime results should be interpreted as deployment-level gains from anticipatory index availability under CPU-offloaded routing, rather than as an isolated microbenchmark of proxy-query construction.

Throughput gain. Figure 7 evaluates practical end-to-end output throughput on a 40K-token input with a 4K prompt budget. ES-RocketKV consistently improves throughput over RocketKV, with gains increasing from 1.3% at 128 output tokens to 12.4% at 8192 output tokens. This trend matches the scope of the optimization: prefill dominates short-output requests, while longer outputs expose more successive decode steps where precomputed indices remove routing-related stalls from the GPU attention path. Appendix D further isolates steady-state post-context decoding, reporting 7.67%–13.14% decode-throughput gains. It also analyzes multi-batch decoding to test whether CPU-side routing and data movement become bottlenecks; the results show the same trend, with this overhead largely hidden behind the FFN-dominated decode path.

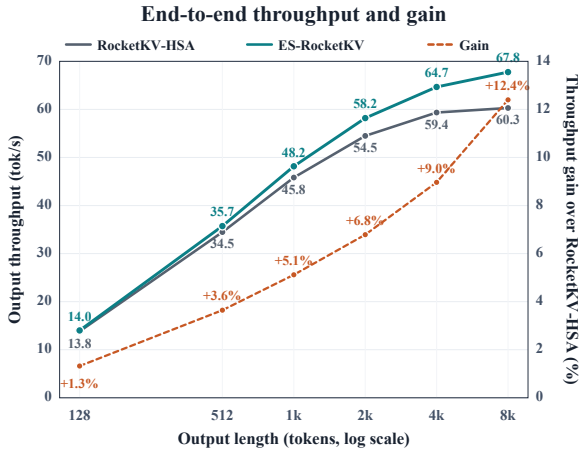


Figure 7: End-to-end output throughput on a 40K-token input with a 4K prompt budget and batch size 1. Solid lines report the output throughput of RocketKV and ES-RocketKV, while the dashed line shows the relative throughput gain of ES-RocketKV over RocketKV.

GPU memory saving. Offloading routing also removes the need to keep K-summary auxiliary states on the GPU. In our configuration, this saves about 5.3 GiB of GPU memory, corresponding to 11.1% of the KV-cache pool. This memory reduction reflects the CPU-offloaded deployment choice; EarlySelectKV makes this placement practical by exposing selected indices early enough for routing and metadata preparation to be staged before the target attention layer.

5 Conclusion

We present EARLYSELECTKV, a training-free transformation for query-conditioned sparse KV selectors. By replacing the target-layer routing query with a proxy from an earlier hidden state, EarlySelectKV makes selected KV indices available before the target attention layer while preserving the underlying selector structure. Instantiated primarily on RocketKV as ES-RocketKV, EarlySelectKV closely matches native RocketKV on LongBench, Needle-in-a-Haystack, and RULER in task score and selection behavior. Our fidelity analysis shows that proxy anchors expose a trade-off between preparation window and routing stability, and our runtime study shows that early index availability can yield practical throughput and memory benefits in deployment-oriented settings. Overall, EarlySelectKV highlights routing timing as an additional optimization dimension for sparse long-context decoding.

Limitations

EARLYSELECTKV targets query-conditioned sparse KV selectors and is less suitable for fixed, query-independent, or separately trained routing policies. Our main experiments focus on RocketKV, with additional selector results reported for Quest and Loki; broader validation on larger models, more selector families, and more diverse workloads remains future work. ES-Mix relies on offline layer-wise profiling to assign anchors. Although this assignment is fixed at inference time and requires no training, its robustness across domains and prompt distributions remains to be further studied. Finally, our runtime study uses a CPU-offloaded TensorRT-LLM implementation to demonstrate how early index availability can be exploited in deployment. Other inference engines or fully GPU-resident implementations may expose different scheduling opportunities and bottlenecks, so the exact throughput gains may vary across systems.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.

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. 2024. 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)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Payman Behnam, Yaosheng Fu, Ritchie Zhao, Po-An Tsai, Zhiding Yu, and Alexey Tumanov. 2025. Rocketkv: accelerating long-context llm inference via two-stage kv cache compression. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25, Vancouver, Canada. JMLR.org.

William Brandon, Mayank Mishra, Aniruddha Nrusimha, Rameswar Panda, and Jonathan Ragan-Kelley. 2024. Reducing transformer key-value cache size with cross-layer attention. In *Advances in Neural Information Processing Systems*, volume 37, pages 86927–86957. Curran Associates, Inc.

Zhuoming Chen, Ranajoy Sadhukhan, Zihao Ye, Yang Zhou, Jianyu Zhang, Niklas Nolte, Yuandong Tian, Matthijs Douze, Leon Bottou, Zhihao Jia, and Beidi Chen. 2024. Magicpig: Lsh sampling for efficient llm generation. *Preprint*, arXiv:2410.16179.

Seonggyu Choi and Hyungmin Cho. 2025. Sparq: An accelerator architecture for large language models with joint sparsity and quantization techniques. In *Proceedings of the 26th ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, and Tools for Embedded Systems*, LCTES '25, page 3–15, New York, NY, USA. Association for Computing Machinery.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, and 138 others. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Chaoyi Jiang, Lei Gao, Hossein Entezari Zarch, and Murali Annavaram. 2025. KVPR: Efficient LLM inference with I/O-aware KV cache partial recomputation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19474–19488, Vienna, Austria. Association for Computational Linguistics.

Yikun Jiang, Huanyu Wang, Lei Xie, Hanbin Zhao, Chao Zhang, Hui Qian, and John C.S. Lui. 2024. D-llm: A token adaptive computing resource allocation strategy for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 1725–1749. Curran Associates, Inc.

Greg Kamradt. 2023. Llmtest_needleinahaystack. `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`. Accessed: 2026-05-24.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. Snapkv: Llm knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, volume 37, pages 22947–22970. Curran Associates, Inc.

Tianyang Liu, Canwen Xu, and Julian McAuley. 2024. Repobench: Benchmarking repository-level code auto-completion systems. In *International Conference on Learning Representations*, volume 2024, pages 47832–47850.

NVIDIA. 2023. Tensorrt-llm. https://github.com/NVIDIA/TensorRT-LLM.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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

Prajwal Singhania, Siddharth Singh, Shwai He, Soheil Feizi, and Abhinav Bhatele. 2024. Loki: Low-rank keys for efficient sparse attention. In *Advances in Neural Information Processing Systems*, volume 37, pages 16692–16723. Curran Associates, Inc.

Yutao Sun, Li Dong, Yi Zhu, Shaohan Huang, Wenhui Wang, Shuming Ma, Quanlu Zhang, Jianyong Wang, and Furu Wei. 2024. You only cache once: Decoder-decoder architectures for language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 7339–7361. Curran Associates, Inc.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. 2024. Quest: query-aware sparsity for efficient long-context llm inference. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, Vienna, Austria. JMLR.org.

Yixuan Wang, Shiyu Ji, Yijun Liu, Yuzhuang Xu, Yang Xu, Qingfu Zhu, and Wanxiang Che. 2025. Lookahead Q-cache: Achieving more consistent KV cache eviction via pseudo query. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34158–34174, Suzhou, China. Association for Computational Linguistics.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pages 21875–21895.

Qiuyang Zhang, Kai Zhou, Ding Tang, Kai Lu, Cheng Li, Zhenyu Yang, Peng Xu, and Jiguang Wan. 2026. Scoutattention: Efficient kv cache offloading via layer-ahead cpu pre-computation for llm inference. *Preprint*, arXiv:2603.27138.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang "Atlas" Wang, and Beidi Chen. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 34661–34710. Curran Associates, Inc.

A Effect of Low-Rank Proxy Projection

We study how the approximation rank affects low-rank proxy-query construction. Given a rank-r approximation of the target-layer query projection,

$$W_q^\ell \approx U_r^\ell \Sigma_r^\ell (V_r^\ell)^\top, \tag{5}$$

the proxy query is computed as

$$q_{t,\text{proxy}}^\ell = U_r^\ell \Sigma_r^\ell (V_r^\ell)^\top \, \text{LN}_\ell(\tilde{H}_{t,\ell}). \tag{6}$$

The rank r controls a direct cost–fidelity trade-off: smaller ranks reduce proxy-query construction cost, while larger ranks better preserve the target-layer query space.

Figure 8 reports LongBench results on Mistral-7B-Instruct-v0.2 with a KV token budget of 512. The original RocketKV obtains an average score of 41.96. A low rank of $r = 64$ reduces the average score to 41.32, mainly due to larger drops on retrieval-sensitive tasks such as PassageRetrieval. Increasing the rank quickly closes the gap: $r = 128$ reaches 41.82, and $r = 256$ and $r = 512$ obtain 42.09 and 42.03, respectively. The performance therefore saturates around $r = 256$, with little benefit from further increasing the rank. We use rank-256 in runtime experiments as a practical balance between proxy-query overhead and task performance.

B Detailed Benchmark Results

This appendix provides detailed numerical results omitted from the main text. For LongBench, we report task-level scores for each evaluated model and averaged scores across tasks. For RULER, we report averaged scores under different context lengths and token budgets.

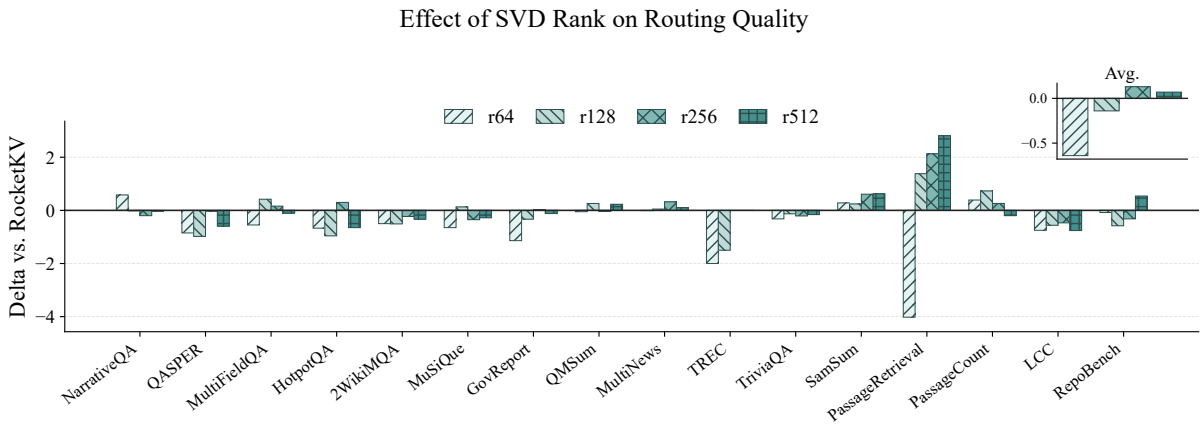


Figure 8: Effect of the low-rank proxy-query projection on LongBench using Mistral-7B-Instruct-v0.2 with a KV token budget of 512. Scores are compared against RocketKV with the full target-layer query projection.

B.1 LongBench Results

Task-level results. Tables 4 to 7 report the full LongBench subtask scores for each evaluated model.

Average results. Table 2 reports averaged LongBench scores under different models and token budgets. These values provide the numerical details behind the LongBench trends shown in the first row of Figure 4.

B.2 Needle-in-a-Haystack Results

We provide the full Needle-in-a-Haystack (NIAH) / Passkey budget sweep omitted from the main text. Accuracy is measured by exact matching of the inserted passkey. For each model, we sweep token budgets from 64 to 4096 and compare RocketKV, ES-RocketKV variants, their TopK counterparts and the ideal TopK baseline when available. Table 8 reports the complete results, while the main text includes only representative budget settings for compactness. In addition, Figure 12 provides a qualitative comparison of the selected-token patterns produced by RocketKV, ES-In, and ES-Mid.

B.3 RULER Results

Table 9 reports averaged RULER scores under different context lengths and token budgets. These values provide the numerical details behind the RULER trends shown in the second and third rows of Figure 4.

C Additional Diagnostics for Proxy Query Fidelity

Additional diagnostics use Mistral-7B-Instruct-v0.2 with a KV token budget of 1024 on 10 LongAlpaca-12k calibration prompts, yielding 38,967 decode-layer records. Mistral is reported as the representative case because other evaluated backbones show similar qualitative layer-wise trends; Qwen2.5-7B-Instruct is the exception, where no clear split is observed, so we do not configure ES-Mix for that model.

On this calibration set, Q_{mid} remains more faithful than Q_{in} overall: cosine similarity / signed top-r recall / retained attention mass are $0.9834/0.8847/0.9128$ for Q_{mid}, compared with $0.9552/0.8465/0.8985$ for Q_{in}. The advantage is concentrated before layer 20, where retained mass is 0.9028 for Q_{mid} and 0.8808 for Q_{in}; after layer 20, the two anchors are nearly identical, with retained masses 0.9309 and 0.9306. Figure 9 provides auxiliary cosine-similarity and page-recall views that are consistent with this layer-wise trend.

Calibration-distribution sensitivity. We further compare the LongAlpaca-12k probe with a Qasper-based probe under the same model, budget, and routing configuration. LongAlpaca-12k is harder in absolute routing fidelity, lowering retained mass by 0.0315–0.0358 across Q_{out}, Q_{mid}, and Q_{in}. However, the layer-wise anchor preference remains stable: the $Q_{\text{mid}} - Q_{\text{in}}$ retained-mass gap is positive before layer 20 on both probes and nearly zero afterward, with a per-layer gap correlation of 0.9997. Threshold-based ES-In/ES-Mid assignments agree on 87.1%–93.5% of layers across tested tolerances, supporting the ES-Mix split used in the main text.

Auxiliary metrics. Cosine similarity measures vector-level alignment with Q_{out}, while page recall against Exact-TopK measures discrete agreement with selected pages. We treat both as auxiliary diag-

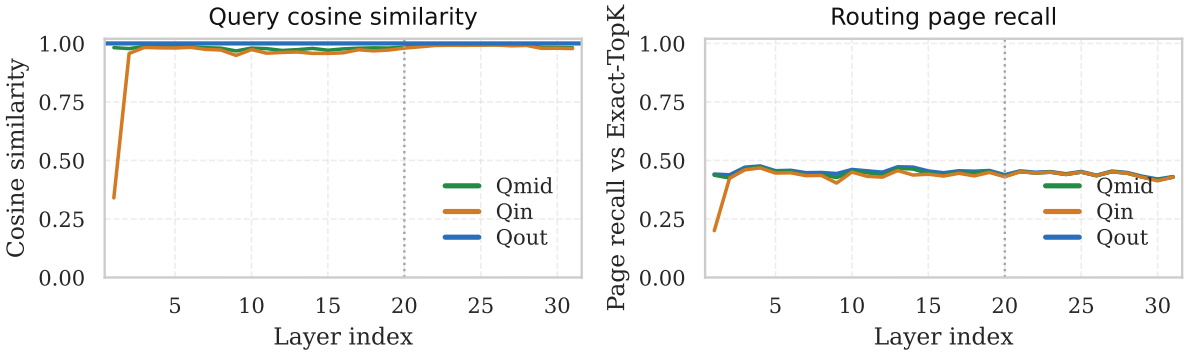


Figure 9: Auxiliary proxy-query diagnostics on the LongAlpaca-12k calibration set. Left: cosine similarity to Q_{out}. Right: page recall against Exact-TopK selection.

nostics: cosine similarity alone does not determine routing quality, and page recall is stricter than retained attention mass because it requires matching the same pages. Accordingly, the main analysis relies on signed top-r recall and retained attention mass, which more directly reflect the routing evidence and attention coverage used by the selector.

Budget	Method			
	FullCache	TopK	RocketKV	ES-RocketKV
Llama3-8B-Instruct				
128	43.23	43.72	41.68	**41.78**
256	43.23	43.58	42.43	**42.72**
512	43.23	43.50	**43.07**	43.03
1024	43.23	43.31	43.18	**43.21**
2048	43.23	43.31	**43.32**	43.29
4096	43.23	43.36	43.23	**43.25**
Llama3.1-8B-Instruct				
128	49.72	49.66	**47.57**	47.47
256	49.72	49.72	**48.36**	48.35
512	49.72	49.66	**49.17**	49.08
1024	49.72	49.70	49.41	**49.50**
2048	49.72	49.56	49.59	**49.63**
4096	49.72	49.73	**49.71**	49.58
Mistral-7B-Instruct-v0.2				
128	43.20	42.02	39.24	**39.32**
256	43.20	42.68	**41.03**	40.98
512	43.20	43.01	41.96	**42.33**
1024	43.20	43.04	**42.71**	42.59
2048	43.20	43.02	42.88	**42.99**
4096	43.20	43.05	43.05	**43.14**
Qwen2.5-7B-Instruct				
128	49.07	47.34	**44.26**	44.06
256	49.07	47.36	**45.84**	45.70
512	49.07	47.66	**46.73**	46.72
1024	49.07	47.46	**46.95**	**46.95**
2048	49.07	47.50	**47.37**	**47.37**
4096	49.07	47.60	47.36	**47.49**

Table 2: Average LongBench scores of different methods under different token budgets.

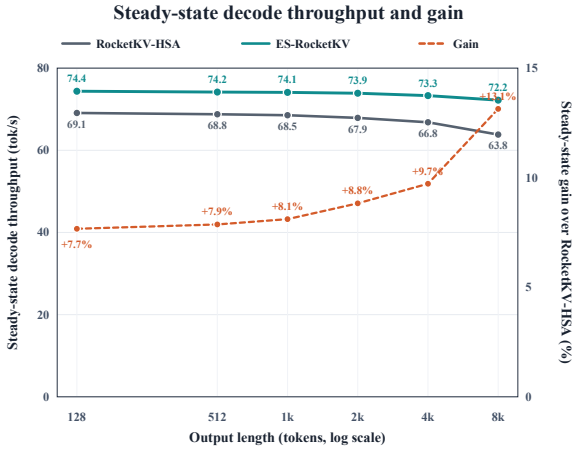


Figure 10: Steady-state decode throughput under different output lengths. The input length is 40K tokens and the prompt budget is 4K tokens. Solid lines report post-context decode throughput of RocketKV-HSA and ES-RocketKV, while the dashed line reports the relative gain of ES-RocketKV.

Benchmark results of different anchor settings. Tables 10 and 11 report averaged LongBench and RULER scores for ES-In, ES-Mid, and ES-Mix. ES-In offers the largest preparation window but may lose fidelity in sensitive layers, while ES-Mid is more conservative and usually preserves selection quality. ES-Mix uses Q_{mid} only where its fidelity advantage is meaningful and switches to Q_{in} once the gap becomes small. The results are consistent with the fidelity diagnostics, showing that ES-Mix balances task accuracy with scheduling opportunity.

D Detailed Runtime Breakdown

Figure 10 reports a steady-state post-context decode breakdown for the runtime experiment in Section 4.5. Unlike the end-to-end metric in Figure 7,

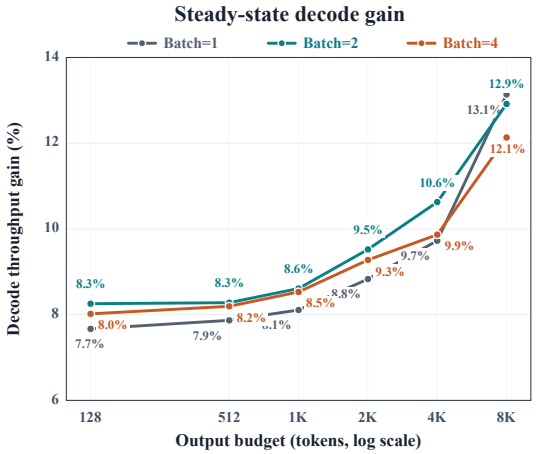


Figure 11: Steady-state decode-throughput gain of ES-RocketKV over RocketKV under different batch sizes. The measurement excludes the context step and the first decode token. The x-axis shows output budget in tokens on a log scale.

Item	Setting
Base selector	RocketKV-HSA; ES-Quest/ES-Loki also tested
Default anchor	ES-Mid
ES-Mix	Offline layer-wise profile; fixed at inference
Sparse layers	Dense layer 0; sparse from layer 1
Local budget	Reserved outside routing; keep tail page
Budget range	128–4096 tokens
Decoding	Greedy
K-summary	FP8
Low-rank proxy	SVD rank 256 for runtime
SVD factors	Computed offline from W_q^ℓ
Proxy profiling	Mistral-7B, LongAlpaca-12k calib. $n = 10$, budget 1024
Profiling records	38,967 decode-layer records
Runtime input	40K input, 4K prompt budget
Runtime OSL	128–8192
Backend	TensorRT-LLM, PyTorch backend, CUDA Graphs
GPU	1× NVIDIA RTX 6000D
CPU offload	Intel Xeon Platinum host CPU
CPU routing	BF16 AVX-512 dot product, FP32 accumulation

Table 3: Key reproducibility settings for EARLYSELECTKV.

this measurement excludes the context step and uses TRT-LLM (NVIDIA, 2023) iteration statistics after prefill. It therefore isolates the decode phase where routing and sparse-attention index preparation are repeatedly invoked.

ES-RocketKV improves steady-state decode throughput across all output lengths, with gains increasing from 7.67% at 128 output tokens to 13.14% at 8192 output tokens. As generation proceeds, the active K-summary view and metadata footprint grow, making routing a larger source of exposed latency when it remains on the synchronous GPU sparse-attention path. ES-RocketKV mitigates this effect by exposing selected indices earlier and decoupling routing and metadata preparation from target attention. This keeps the GPU consuming prepared indices rather than waiting on routing, preserving a throughput advantage throughout decoding.

We further evaluate whether CPU-side routing and index transfer become bottlenecks under higher concurrency. Figure 11 reports steady-state decode-throughput gains for batch sizes 1, 2, and 4. Across output lengths, the gain remains in a similar range for all batch sizes: 7.67%–13.14% for batch 1, 8.25%–12.91% for batch 2, and 8.02%–12.13% for batch 4. This indicates that CPU-side routing and data movement remain largely hidden behind the FFN-dominated decode path in this setting, rather than becoming the dominant bottleneck as batch size increases.

We also tested GPU-side overlap by launching selector computation on a side CUDA stream. This provides limited benefit in our setting because decoder FFN and attention kernels already occupy most GPU compute and memory-bandwidth resources. CPU-offloaded routing is more effective because it removes routing work from contested GPU resources rather than scheduling it on another GPU stream.

E Reproducibility Details

Most experimental and implementation details are described alongside the corresponding experiments. For convenience, we summarize the key settings needed to reproduce EARLYSELECTKV in Table 3. Unless otherwise stated, ES variants differ from their base selectors only in the routing-query source.

Code availability. We provide an anonymized code repository at `https://anonymous.4open.science/r/EarlySelectKV-76B3/`. The repository includes the implementation of EARLYSELECTKV, benchmark evaluation scripts, and configuration files for reproducing the main experimental results.

Budget	Method	NarrativeQA	QASPER	MultiFieldQA	HotpotQA	2WikiMQA	MuSiQue	GovReport	QMSum	MultiNews	TREC	TriviaQA	SamSum	PassageRetrieval	PassageCount	LCC	RepoBench
								Llama3-8B-Instruct									
FULL	FullCache	21.17	43.17	46.54	46.81	39.93	23.05	30.94	22.70	27.97	73.00	90.00	42.16	66.50	8.00	60.63	49.07
64	TopK	19.97	42.53	46.29	47.32	39.90	22.75	29.20	22.37	27.10	72.50	90.36	42.16	67.00	7.50	64.88	57.64
	RocketKV	18.44	**30.47**	**46.57**	**46.31**	**34.83**	21.34	**20.72**	**21.14**	**23.82**	55.00	**89.68**	38.20	**65.50**	**6.00**	62.36	57.92
	ES-RocketKV	**19.19**	28.77	46.42	44.88	34.78	**21.47**	20.62	21.09	23.72	**57.00**	89.49	**38.66**	**65.50**	5.50	**64.22**	**58.16**
128	TopK	20.55	43.15	45.16	46.81	41.52	22.12	29.19	22.42	27.45	72.50	90.86	42.17	66.50	8.50	64.76	55.91
	RocketKV	18.15	40.40	45.92	**46.28**	36.89	21.92	21.75	21.13	**24.19**	**66.50**	**90.05**	**40.86**	66.50	**5.00**	**63.18**	**58.11**
	ES-RocketKV	**18.77**	**41.41**	**46.15**	46.16	**38.23**	**22.27**	**22.13**	**21.27**	24.03	**66.50**	89.12	40.10	**67.00**	**5.00**	63.09	57.26
256	TopK	21.15	43.12	45.72	46.62	41.13	22.70	29.93	22.32	27.42	72.50	91.00	42.29	66.50	8.00	64.12	52.73
	RocketKV	19.37	**41.84**	45.89	46.43	38.82	22.91	23.74	**21.70**	24.54	**70.50**	89.36	41.68	**67.00**	6.00	63.46	55.64
	ES-RocketKV	**19.75**	41.57	**46.68**	**46.79**	**39.19**	**23.14**	**23.82**	21.49	**24.58**	**70.50**	**89.87**	**42.34**	66.50	**7.00**	**64.21**	**56.12**
512	TopK	21.49	43.51	45.29	46.91	41.38	22.72	30.37	22.67	27.59	72.50	91.00	42.36	66.50	8.00	62.43	51.34
	RocketKV	**19.97**	**43.00**	**46.36**	46.93	**40.21**	**23.08**	**26.75**	**22.31**	**26.05**	**71.50**	89.95	**42.84**	66.50	**7.00**	63.35	53.38
	ES-RocketKV	19.51	42.90	45.96	**47.31**	39.86	22.75	26.31	22.14	25.90	**71.50**	**90.08**	42.76	**67.00**	6.50	**63.67**	**54.41**
1024	TopK	21.00	43.37	45.97	47.20	41.15	22.79	30.63	22.90	27.70	72.50	91.00	41.84	66.50	7.50	61.23	49.67
	RocketKV	**20.92**	**43.45**	**45.98**	**46.63**	40.21	**23.09**	**28.67**	22.10	**26.90**	**73.00**	**90.00**	**41.97**	**66.50**	**8.00**	62.12	51.35
	ES-RocketKV	20.85	43.30	45.56	**46.63**	**40.65**	**23.09**	28.60	**22.13**	26.78	**73.00**	**90.00**	41.96	**66.50**	7.50	**62.22**	**52.52**
2048	TopK	21.23	43.09	46.60	46.97	40.83	22.77	30.77	22.91	27.80	72.50	91.00	41.91	66.50	8.00	60.86	49.17
	RocketKV	21.24	**43.67**	46.33	46.18	**40.41**	23.01	**30.05**	**22.57**	27.61	**73.00**	**90.00**	**42.41**	**66.50**	**8.50**	61.16	**50.45**
	ES-RocketKV	**21.33**	43.29	**46.53**	**46.68**	**40.41**	**23.05**	29.72	22.50	**27.64**	**73.00**	**90.00**	41.75	**66.50**	**8.50**	**61.45**	50.25
4096	TopK	21.18	43.32	46.59	47.05	41.04	22.79	31.18	22.74	27.90	72.50	90.83	42.33	66.50	8.00	60.92	48.96
	RocketKV	**21.29**	**43.74**	46.32	**46.80**	39.93	**22.98**	30.50	**22.91**	27.80	**73.00**	**90.00**	41.70	**66.50**	**8.00**	60.68	**49.61**
	ES-RocketKV	21.25	43.33	**46.50**	46.47	**40.48**	**22.98**	**30.52**	22.86	**27.83**	**73.00**	**90.00**	**42.40**	**66.50**	**8.00**	**60.72**	49.17

Table 4: Task-level LongBench scores for Llama3-8B-Instruct.

Budget	Method	NarrativeQA	QASPER	MultiFieldQA	HotpotQA	2WikiMQA	MuSiQue	GovReport	QMSum	MultiNews	TREC	TriviaQA	SamSum	PassageRetrieval	PassageCount	LCC	RepoBench
								Llama3.1-8B-Instruct									
FULL	FullCache	32.12	46.58	55.73	57.84	48.61	30.82	34.78	25.42	27.07	73.00	92.14	43.76	99.50	7.33	65.36	55.42
64	TopK	31.35	45.44	55.80	58.40	48.92	31.56	33.73	24.41	26.28	73.00	92.25	43.06	99.50	7.92	65.12	53.72
	RocketKV	27.17	29.99	**57.32**	56.65	**44.88**	**28.77**	22.85	**23.24**	23.30	56.50	91.07	**40.10**	**99.00**	**9.00**	**63.50**	**52.23**
	ES-RocketKV	**27.35**	**31.46**	56.42	**56.87**	44.19	27.29	**23.17**	23.15	**23.37**	**58.00**	**91.70**	39.98	98.50	7.83	63.20	51.39
128	TopK	30.99	45.27	55.84	58.07	49.03	31.84	34.25	25.57	26.56	73.00	92.25	44.23	99.50	7.58	65.83	54.78
	RocketKV	**30.26**	**41.12**	**57.26**	57.15	**47.93**	**29.85**	**24.10**	23.70	23.84	**67.00**	91.73	**41.55**	**99.50**	6.83	**65.16**	54.12
	ES-RocketKV	29.76	40.68	56.31	**58.17**	46.76	29.29	23.97	**24.25**	**24.07**	**67.00**	**91.75**	41.17	**99.50**	**7.90**	64.55	**54.36**
256	TopK	30.01	45.38	55.33	57.69	49.64	32.20	34.68	25.54	26.75	73.00	91.95	44.24	99.50	8.00	66.04	55.52
	RocketKV	30.54	**44.03**	**56.22**	57.28	**47.61**	30.22	26.12	24.52	**24.42**	**71.00**	91.78	42.55	**100.00**	7.00	65.06	**55.41**
	ES-RocketKV	**30.69**	42.35	55.45	**57.99**	47.28	**31.15**	**26.59**	**24.65**	24.25	70.50	**91.81**	**42.58**	99.50	**8.00**	**65.63**	55.21
512	TopK	30.29	46.12	55.40	57.58	49.22	32.20	34.50	25.37	26.97	73.00	91.97	43.63	99.50	7.92	65.47	55.42
	RocketKV	31.60	**45.99**	55.73	**58.40**	**48.28**	**30.18**	29.85	**25.03**	**25.51**	**72.50**	91.64	**43.46**	**99.50**	**7.00**	**66.06**	**55.96**
	ES-RocketKV	**31.75**	45.49	**55.89**	58.18	48.08	29.77	**30.02**	24.68	25.48	**72.50**	**92.11**	43.03	**99.50**	**7.00**	65.98	55.81
1024	TopK	30.67	46.26	55.72	57.73	49.22	32.04	34.75	25.68	27.01	73.00	91.97	43.13	99.50	7.50	65.21	55.75
	RocketKV	**31.25**	**45.76**	55.81	57.68	**48.71**	31.00	**32.24**	**25.60**	**26.91**	**73.00**	91.64	43.72	**99.50**	**6.75**	65.58	55.49
	ES-RocketKV	31.16	45.22	**55.87**	**57.78**	48.68	**32.02**	32.17	25.48	26.66	**73.00**	**91.97**	**43.89**	**99.50**	6.50	**66.13**	**55.97**
2048	TopK	31.28	46.05	55.58	57.33	48.29	31.61	34.59	25.48	27.27	73.00	91.97	43.51	99.50	7.25	65.06	55.22
	RocketKV	31.39	46.41	**56.08**	57.41	48.71	30.66	**33.43**	**25.58**	26.97	**73.00**	91.64	**43.65**	**99.50**	**8.42**	**65.31**	55.29
	ES-RocketKV	**31.57**	**46.43**	55.84	**57.94**	**48.92**	**31.90**	33.41	25.48	**27.16**	**73.00**	**91.97**	43.64	**99.50**	6.17	65.18	**56.04**
4096	TopK	31.61	46.33	55.82	57.58	48.68	32.05	34.49	25.39	27.34	73.00	91.97	43.50	99.50	8.00	64.97	55.42
	RocketKV	**31.67**	46.53	**56.05**	57.54	48.52	31.55	34.02	**25.20**	**27.27**	72.50	**91.97**	**43.89**	**99.50**	**8.52**	**65.17**	55.46
	ES-RocketKV	31.12	**46.56**	55.89	**58.09**	**48.54**	**31.85**	**34.09**	25.10	26.97	**73.00**	**91.97**	43.35	**99.50**	6.07	65.13	**56.04**

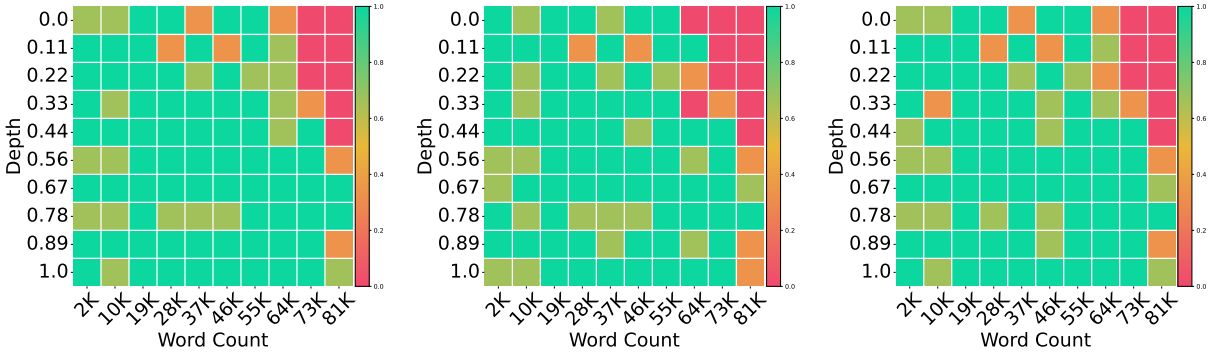
Table 5: Task-level LongBench scores for Llama3.1-8B-Instruct.

Budget	Method	NarrativeQA	QASPER	MultiFieldQA	HotpotQA	2WikiMQA	MuSiQue	GovReport	QMSum	MultiNews	TREC	TriviaQA	SamSum	PassageRetrieval	PassageCount	LCC	RepoBench
								Mistral-7B-Instruct-v0.2									
FULL	FullCache	28.00	32.61	48.43	42.26	26.44	19.01	33.16	24.29	27.16	71.00	86.92	43.61	83.80	2.95	59.97	61.52
64	TopK	25.46	24.03	44.39	37.87	24.23	17.75	30.42	19.76	27.05	71.00	87.11	40.80	75.60	2.43	58.55	57.99
	RocketKV	22.24	19.12	**48.10**	**37.73**	**23.80**	16.08	23.04	22.61	24.26	**59.00**	**86.43**	39.42	**50.01**	**4.01**	**55.47**	**55.84**
	ES-RocketKV	**22.58**	**19.71**	47.56	36.95	22.90	**16.29**	**23.41**	**22.70**	**24.72**	**59.00**	86.41	**39.94**	43.52	3.99	55.03	55.30
128	TopK	25.75	30.06	48.21	39.28	25.44	17.88	31.68	21.86	27.13	71.00	87.27	42.95	82.37	2.53	59.21	59.62
	RocketKV	**25.17**	**26.43**	**48.86**	38.10	23.90	**18.47**	**23.22**	**23.17**	**24.41**	68.50	**87.61**	**42.00**	58.62	**3.61**	**57.78**	57.92
	ES-RocketKV	23.55	26.22	48.51	**39.48**	**25.35**	17.25	23.00	22.69	24.22	**69.50**	87.26	41.74	**61.57**	2.99	57.00	**58.79**
256	TopK	26.08	32.01	48.40	39.64	25.15	17.97	32.49	23.51	27.27	71.00	87.29	44.36	83.94	3.24	59.93	60.67
	RocketKV	25.50	28.05	**48.99**	38.43	26.42	**18.55**	**25.99**	**23.84**	24.74	**70.50**	86.85	**42.85**	74.26	**3.29**	**58.64**	**59.54**
	ES-RocketKV	**25.79**	**28.81**	48.78	**38.51**	**26.74**	18.44	25.59	23.83	**24.79**	70.00	**86.93**	42.24	**75.31**	2.75	57.72	59.44
512	TopK	26.99	31.54	48.89	41.22	25.32	18.84	33.12	23.91	27.53	71.00	87.08	44.40	84.51	3.59	59.49	60.65
	RocketKV	26.46	30.97	**48.77**	40.34	25.84	18.44	28.95	23.87	25.96	**70.50**	87.08	**43.25**	77.93	**3.21**	**59.74**	**60.12**
	ES-RocketKV	**26.99**	**31.03**	48.46	**40.76**	**26.05**	**19.12**	**29.25**	**24.11**	**26.14**	**70.50**	**87.65**	43.17	**81.83**	3.00	59.19	60.04
2048	TopK	27.57	32.09	48.49	41.90	25.49	19.05	33.24	24.07	27.21	71.00	87.04	43.98	84.01	3.02	59.16	60.95
	RocketKV	27.23	**31.74**	**48.70**	41.66	25.32	18.91	**32.63**	24.49	27.12	**71.00**	**87.13**	43.49	83.29	2.97	59.51	60.91
	ES-RocketKV	**27.51**	31.45	48.28	**42.00**	**25.87**	**19.13**	32.45	**24.50**	**27.20**	**71.00**	87.08	**43.62**	**84.24**	**3.07**	**59.53**	**60.96**
4096	TopK	27.28	32.68	48.57	41.95	25.60	19.16	33.45	24.18	27.15	71.00	86.86	43.96	83.51	2.81	59.21	61.47
	RocketKV	**27.32**	32.06	48.31	42.10	26.06	**19.06**	33.25	24.56	27.38	**71.00**	**87.15**	43.51	**83.83**	2.94	59.29	**61.00**
	ES-RocketKV	27.27	**32.41**	**48.59**	**42.27**	**26.28**	18.96	**33.43**	**24.58**	**27.48**	**71.00**	**87.15**	**44.27**	83.33	**3.04**	**59.37**	60.88

Table 6: Task-level LongBench scores for Mistral-7B-Instruct-v0.2.

Budget	Method	NarrativeQA	QASPER	MultiFieldQA	HotpotQA	2WikiMQA	MuSiQue	GovReport	QMSum	MultiNews	TREC	TriviaQA	SamSum	PassageRetrieval	PassageCount	LCC	RepoBench
								Qwen2.5-7B-Instruct									
FULL	FullCache	27.77	45.33	51.63	58.00	46.13	29.88	33.51	24.28	25.81	73.00	88.41	47.02	100.00	9.50	59.86	64.92
64	TopK	28.57	42.68	49.33	54.18	43.26	27.85	29.69	23.35	24.77	70.00	87.62	46.18	96.25	9.50	59.30	61.86
	RocketKV	20.88	**29.82**	**50.17**	**50.90**	**39.19**	24.96	**21.37**	**21.41**	**20.77**	**50.50**	**83.07**	**40.87**	**86.58**	**8.00**	**50.80**	**50.79**
	ES-RocketKV	**21.56**	27.97	50.13	49.92	37.50	**25.09**	20.38	21.09	20.61	49.00	82.91	39.56	83.83	7.50	49.01	50.40
128	TopK	28.30	43.73	49.13	54.26	45.12	28.62	28.52	23.37	24.41	69.50	87.58	46.44	97.25	9.50	59.13	62.50
	RocketKV	26.05	38.90	**51.30**	**53.53**	**42.80**	**26.49**	**23.43**	**21.66**	**22.13**	**61.00**	83.90	42.74	**94.00**	7.50	**55.34**	**57.37**
	ES-RocketKV	**26.31**	**39.19**	50.71	52.56	42.15	25.51	22.77	21.59	21.84	59.50	**84.33**	**42.98**	93.75	**8.00**	53.58	56.54
256	TopK	28.37	42.76	48.64	54.91	45.45	28.30	29.30	23.61	24.03	70.50	86.14	46.01	98.50	9.50	58.76	63.04
	RocketKV	**28.65**	**41.72**	**51.71**	54.43	**43.76**	27.43	**25.37**	22.29	**22.20**	**64.50**	85.31	44.47	**95.42**	**7.50**	58.32	**60.37**
	ES-RocketKV	27.31	41.68	51.16	**54.52**	43.65	**27.96**	25.33	**22.66**	21.90	**64.50**	**85.48**	**44.55**	94.75	**7.50**	**58.39**	59.87
512	TopK	28.19	42.79	49.47	55.33	45.07	28.42	29.98	23.81	24.61	71.50	86.44	46.22	98.50	9.50	58.72	63.57
	RocketKV	**27.81**	**42.40**	**49.69**	55.43	44.05	28.89	27.92	**23.38**	23.48	**66.00**	**87.63**	45.16	**96.75**	**9.50**	**58.80**	60.76
	ES-RocketKV	27.60	42.27	49.64	**55.83**	**44.24**	**29.23**	**28.16**	23.17	**23.65**	65.50	87.38	**45.78**	96.25	**9.50**	58.50	**60.78**
1024	TopK	27.11	43.14	49.63	55.39	44.48	29.37	29.05	23.89	25.13	70.50	86.41	45.67	98.50	9.00	58.94	63.15
	RocketKV	28.77	41.65	49.31	54.31	43.73	**30.05**	28.64	23.68	**23.95**	**68.50**	**85.77**	**46.01**	**97.00**	**9.00**	**58.75**	**62.05**
	ES-RocketKV	**29.18**	**41.70**	**49.85**	**55.06**	**43.77**	30.00	**28.75**	**23.91**	23.90	67.50	85.74	45.55	**97.00**	**9.00**	58.63	61.74
2048	TopK	27.93	42.77	49.73	55.99	44.50	29.33	29.20	23.52	25.06	71.00	86.20	45.98	98.25	9.00	58.88	62.72
	RocketKV	**28.97**	**42.38**	49.40	54.45	**44.66**	29.26	**30.06**	23.93	**25.28**	**69.50**	85.53	**45.94**	**98.00**	**9.00**	**58.84**	62.68
	ES-RocketKV	28.69	42.31	**49.92**	**55.44**	44.65	**29.64**	29.82	**24.10**	24.91	69.00	**85.57**	45.44	**98.00**	**9.00**	58.46	**62.92**
4096	TopK	27.82	42.58	48.93	56.40	45.12	29.75	30.20	23.85	24.86	71.00	86.40	46.22	97.75	9.00	58.98	62.78
	RocketKV	**27.85**	**42.76**	49.37	**55.53**	43.94	29.52	29.25	**24.12**	**25.18**	**70.50**	85.80	**45.78**	**97.75**	**9.00**	58.81	62.60
	ES-RocketKV	27.63	42.63	**49.89**	55.27	**44.57**	**29.66**	**30.33**	24.07	25.13	70.00	**86.39**	45.74	**97.75**	**9.00**	**58.96**	**62.81**

Table 7: Task-level LongBench scores for Qwen2.5-7B-Instruct.



(a) RocketKV (b) ES-In (c) ES-Mid

Figure 12: Attention heatmaps on Qwen2.5-7B-Instruct with a token budget of 256, comparing RocketKV, ES-In, and ES-Mid.

Model	Method	Acc.@64	First 100% Budget
Mistral-7B-v0.2	TopK	100.0	64
	RocketKV	100.0	64
	ES-In	100.0	64
	ES-Mid	100.0	64
	ES-Mix	100.0	64
Llama3-8B	TopK	100.0	64
	RocketKV	98.0	128
	ES-In	100.0	64
	ES-Mid	98.0	128
	ES-Mix	100.0	64
Llama3.1-8B	TopK	100.0	64
	RocketKV	88.0	128
	ES-In	85.0	–
	ES-Mid	88.0	128
	ES-Mix	86.0	128

Qwen2.5-7B-Instruct				
Budget	TopK	RocketKV	ES-In	ES-Mid
64	76.0	48.0	**56.0**	48.0
128	76.0	64.0	**67.0**	66.0
256	74.0	**81.0**	79.0	80.0
512	75.0	76.0	77.0	**78.0**
1024	76.0	**78.0**	76.0	76.0
2048	75.0	76.0	76.0	**77.0**
4096	74.0	**77.0**	75.0	76.0

Table 8: NIAH/Passkey exact matching accuracy. All settings use `10x10x3_7digits`; context lengths are 4096 words for Llama3-8B-Instruct, 20480 words for Mistral-7B-Instruct-v0.2, and 81920 words for Llama3.1-8B-Instruct and Qwen2.5-7B-Instruct. For compactness, models reaching perfect retrieval are summarized by the lowest budget achieving 100% accuracy, while Qwen2.5-7B-Instruct is shown separately.

Context	Model	Budget	FullCache	TopK	RocketKV	ES-RocketKV
RULER-8K	Mistral-7B-Instruct-v0.2	128	90.65	89.52	**81.33**	81.16
		256	90.65	90.34	**85.09**	84.95
		512	90.65	90.45	**88.53**	88.50
		1024	90.65	90.80	**89.65**	89.63
		2048	90.65	90.61	89.70	**90.07**
	Llama3.1-8B-Instruct	128	95.31	92.71	**85.62**	85.13
		256	95.31	94.15	**89.72**	89.57
		512	95.31	94.87	**91.88**	91.62
		1024	95.31	95.17	**94.02**	93.64
		2048	95.31	95.27	**94.67**	94.45
	Qwen2.5-7B-Instruct	128	93.36	73.71	**70.45**	69.17
		256	93.36	75.19	**74.41**	73.09
		512	93.36	75.89	**74.06**	73.70
		1024	93.36	75.91	74.75	**75.31**
		2048	93.36	75.80	74.17	**75.21**
RULER-16K	Mistral-7B-Instruct-v0.2	128	86.23	83.48	75.18	**75.89**
		256	86.23	85.23	78.27	**80.38**
		512	86.23	85.65	82.72	**83.05**
		1024	86.23	86.26	**84.44**	84.34
		2048	86.23	86.49	85.33	**85.49**
	Llama3.1-8B-Instruct	128	93.22	90.31	**85.09**	84.70
		256	93.22	91.56	**87.75**	87.29
		512	93.22	92.75	89.52	**89.61**
		1024	93.22	93.02	**91.44**	91.38
		2048	93.22	93.16	92.11	**92.22**
	Qwen2.5-7B-Instruct	128	92.03	68.16	**63.75**	63.57
		256	92.03	69.91	**67.27**	66.78
		512	92.03	70.16	68.73	**69.67**
		1024	92.03	70.74	**70.54**	70.05
		2048	92.03	71.68	70.66	**71.14**
RULER-32K	Mistral-7B-Instruct-v0.2	128	79.74	74.26	**65.49**	65.33
		256	79.74	76.67	72.27	**72.35**
		512	79.74	79.26	74.47	**74.56**
		1024	79.74	79.87	75.88	**77.08**
		2048	79.74	80.61	**77.99**	77.85
	Llama3.1-8B-Instruct	128	88.43	86.17	82.73	**82.98**
		256	88.43	87.23	84.59	**84.95**
		512	88.43	88.27	85.61	**85.77**
		1024	88.43	88.83	**86.32**	86.09
		2048	88.43	88.36	**87.56**	87.34
	Qwen2.5-7B-Instruct	128	89.37	64.01	**55.71**	55.18
		256	89.37	64.96	**60.91**	60.80
		512	89.37	64.43	62.22	**62.82**
		1024	89.37	64.35	**64.89**	64.05
		2048	89.37	65.54	**64.02**	62.97

Table 9: Average matching accuracy (%) on RULER across context lengths.

Model	Budget	RocketKV	ES-in	ES-mid	ES-Mix
Llama3-8B-Instruct	64	39.89	39.54	39.97	**40.02**
	128	41.68	41.62	41.78	**41.92**
	256	42.43	42.45	**42.72**	42.65
	512	43.07	43.02	43.03	**43.15**
	1024	43.18	43.16	**43.21**	43.14
	2048	43.32	43.36	43.29	**43.38**
	4096	43.23	43.27	43.25	**43.29**
Llama3.1-8B-Instruct	64	45.35	44.92	45.24	**45.36**
	128	**47.57**	47.48	47.47	47.46
	256	48.36	48.28	48.35	**48.41**
	512	49.17	49.21	49.08	**49.26**
	1024	49.41	**49.62**	49.50	49.47
	2048	49.59	**49.78**	49.63	49.55
	4096	49.71	**49.82**	49.58	49.58
Mistral-7B-Instruct-v0.2	64	**36.70**	35.62	36.25	36.25
	128	39.24	39.11	**39.32**	39.24
	256	**41.03**	40.73	40.98	40.94
	512	41.96	**42.39**	42.33	42.18
	1024	42.71	42.57	42.59	**42.75**
	2048	42.88	42.82	**42.99**	42.86
	4096	43.05	**43.20**	43.14	43.13

Table 10: Average LongBench scores of RocketKV and EarlySelectKV routing variants.

Context	Model	Budget	RocketKV	ES-In	ES-Mid	ES-Mix
RULER-8K	Mistral-7B-Instruct-v0.2	128	**81.33**	80.72	81.16	80.73
		256	85.09	84.76	84.95	**85.18**
		512	**88.53**	87.88	88.50	88.31
		1024	**89.65**	89.49	89.63	89.58
		2048	89.70	89.96	**90.07**	89.77
	Llama3.1-8B-Instruct	128	**85.62**	84.54	85.13	85.38
		256	**89.72**	88.67	89.57	89.71
		512	**91.88**	91.06	91.62	91.59
		1024	**94.02**	93.23	93.64	93.79
		2048	**94.67**	94.39	94.45	94.49
RULER-16K	Mistral-7B-Instruct-v0.2	128	75.18	75.22	**75.89**	75.87
		256	78.27	77.87	**80.38**	80.26
		512	82.72	81.36	**83.05**	82.92
		1024	**84.44**	83.66	84.34	84.05
		2048	85.33	85.27	**85.49**	85.40
	Llama3.1-8B-Instruct	128	**85.09**	84.42	84.70	84.94
		256	**87.75**	87.14	87.29	87.57
		512	89.52	89.34	**89.61**	89.57
		1024	**91.44**	91.28	91.38	91.28
		2048	92.11	91.96	92.22	**92.32**
RULER-32K	Mistral-7B-Instruct-v0.2	128	**65.49**	64.19	65.33	65.04
		256	72.27	71.85	72.35	**72.57**
		512	74.47	74.49	**74.56**	74.54
		1024	75.88	76.22	**77.08**	76.76
		2048	**77.99**	77.29	77.85	77.50
	Llama3.1-8B-Instruct	128	82.73	81.71	**82.98**	82.87
		256	84.59	84.58	**84.95**	84.90
		512	85.61	85.56	85.77	**85.96**
		1024	86.32	86.39	86.09	**86.54**
		2048	**87.56**	87.29	87.34	87.36

Table 11: Average matching accuracy (%) of RocketKV and EarlySelectKV routing variants on RULER.