QKSieve: Query–Key Balanced Mixed-Bit Retrieval for Efficient Long-Context Attention

Anonymous authors
Paper under double-blind review

Abstract

Autoregressive decoding repeatedly scans the complete key–value (KV) history. Query-aware sparse attention helps only when selection is cheaper than dense QK scoring and faithful to joint Query–Key geometry. We present QKSieve, a training-free, GPU-resident exact-KV retrieval method. Request-local Query and Key second moments define biorthogonal coordinates that preserve every unquantized dot product and order dimensions by joint score energy. A per-layer, per-head allocator assigns 0/1/2/4/8 bits to eight 16-D bands under a 240-bit token/head budget by minimizing a separable logit-MSE surrogate. The packed index targets $B(N) = \min\{N, 1280, \max(256, \lceil .06N \rceil)\}$ positions, after which the original 16-bit K/V receive exact sparse attention. No learned router, exact-QK reranking, task rule, or dense fallback is used. Robust adds a current-Query rank-16 INT4 estimate of the omitted softmax partition and Value numerator, bringing total auxiliary storage to 7.47%. On complete LongBench, Robust retains 99.98%, 99.73%, and 100.00% of Full macro score on Llama-3.1-8B, Qwen3-4B, and Mistral-7B while targeting 6.85–7.43% of history. Across 13 RULER tasks from 4K to 128K it retains 100.33% on average with 4.97% measured active attention. In a controlled native-MHA RTX 3090 implementation, the complete Robust attention path is 2.09–4.12× faster at 32K–128K, and real steady decode is 1.32–3.98× faster. We separately report index construction and persistent-cache break-even rather than hiding fixed cost in steady-state timing. Analysis connects representation error to ranking margins, omitted probability mass, and attention-output error.

1 Introduction

Long-context decoding repeatedly rereads the key–value (KV) history for every generated token. Eviction methods reduce this traffic by permanently removing states (Zhang et al., 2023; Li et al., 2024; Cai et al., 2024; Feng et al., 2024); query-aware methods instead keep the history addressable and retrieve a new subset for each Query (Ribar et al., 2023; Tang et al., 2024; Singhania et al., 2024). The latter is useful only when selection is substantially cheaper than dense QK scoring and preserves the tokens that determine the attention output. Existing low-bit indices make scanning practical (Wang et al., 2025), but a representation optimized to reconstruct Keys need not preserve the Query-dependent logits used for retrieval. We therefore ask: under a small index budget, which representation preserves attention rankings while remaining cheap enough to scan at every decode step?

QKSieve treats selection as a joint Query–Key rate–distortion problem. For each layer and KV head, request-local Query and Key second moments define biorthogonal coordinates that preserve every unquantized dot product and order dimensions by joint score energy. The 128 coordinates are divided into eight 16-D bands; a finite optimizer assigns each band 0/1/2/4/8 bits under a 240-bit token/head budget by minimizing Query-weighted logit distortion. This packed Key index costs 5.86% of resident FP16 K+V storage. A sampled-quantile CUDA scan selects a length-adaptive budget, and the original FP16 K/V at those positions receive exact sparse attention. The method uses no learned router, exact-QK reranking, task rule, or dense fallback.

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.

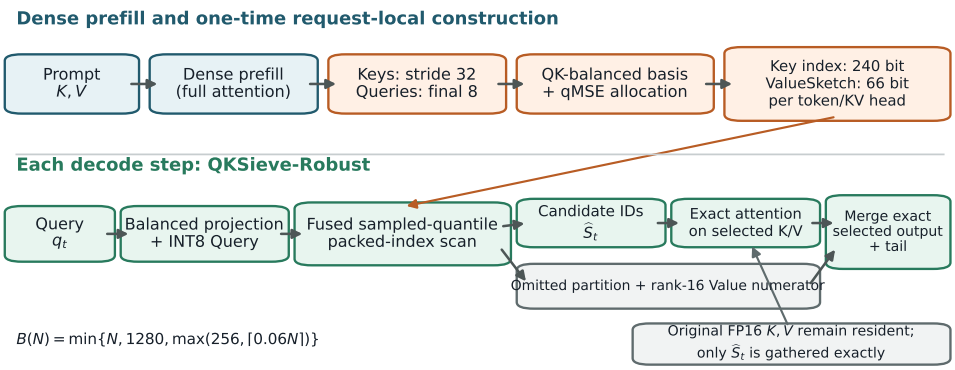


Figure 1: QKSieve-Robust. Prefill constructs request-local QK-balanced Key and Value indices. Decode fuses packed selection with omitted-tail statistics, applies exact attention to selected FP16 K/V, and merges the two outputs.

Ranking alone cannot repair diffuse omitted Value mass. QKSieve-Robust therefore estimates both the omitted softmax partition and its Value numerator from current-Query proxy scores and a request-local rank-16 INT4 Value sketch. The correction raises auxiliary storage to 7.47% without reading additional exact K/V; QKSieve-Quant omits it and exposes its quality–latency cost.

On the complete 16-task LongBench suite, Robust retains 99.98%, 99.73%, and 100.00% of Full macro score on Llama-3.1-8B, Qwen3-4B, and Mistral-7B while targeting 6.85–7.43% of history. Across 13 RULER tasks from 4K to 128K it retains 100.33% with 4.97% measured active attention. In controlled native MHA on RTX 3090, complete Robust attention is 2.09–4.12× faster and steady decode is 1.32–3.98× faster at 32K–128K. We report index construction and persistent-cache break-even separately. Native GQA remains a boundary: a no-tail shared-candidate prototype accelerates complete attention but yields only 1.09× whole-model decode at 32K.

Our contributions are:

- a mixed-bit realization of biorthogonal QK geometry that exactly preserves scores before quantization and minimizes an explicit, Query-weighted distortion under a byte budget;
- a packed sampled-quantile selector and an omitted-Value estimator that connect low-bit retrieval to exact sparse attention without reranking or dense fallback; and
- bounds connecting representation error, ranking margins, omitted mass, and attention-output error, supported by complete LongBench, RULER, and directly timed system evaluations with explicit applicability boundaries.

2 Related Work

KV reduction. H_2O, SnapKV, PyramidKV, and Ada-KV reduce physical cache size by evicting tokens or allocating retention budgets across layers and heads (Zhang et al., 2023; Li et al., 2024; Cai et al., 2024; Feng et al., 2024). CompressKV uses offline semantic-head and layer-utility estimates for prompt eviction (Lin et al., 2026); Minima-KV retains all pages in mixed formats (Kozyrev & Maiboroda, 2026); KIVI quantizes retained KV (Liu et al., 2024b). QKSieve instead keeps every exact KV pair and adds a retrieval index, targeting decode-time attention rather than cache capacity.

Query-aware sparse attention. Quest prunes pages using Key statistics (Tang et al., 2024); SparQ retrieves with salient Query dimensions and corrects omitted probability mass with a mean Value (Ribar et al., 2023); Loki, ShadowKV, LRQK, SALS, RocketKV, and Thin Keys exploit low-dimensional Key or QK structure (Singhania et al., 2024; Sun et al., 2024;

Li et al., 2025; Mu et al., 2025; Behnam et al., 2025; Yao et al., 2026). RetrievalAttention and RetroInfer couple retrieval with offloaded exact KV, while HiSparse provides exact, indexer-agnostic host/GPU placement (Liu et al., 2024a; Chen et al., 2026; Xie et al., 2026). These systems address placement; our main contract keeps exact KV resident and measures how token IDs are selected plus the subsequent sparse attention.

xKV jointly factorizes grouped-layer K/V caches into an aligned shared SVD subspace; xKV-SR and xKV-SR selectively reconstruct sparse K/V at decode time (Chang et al., 2026). This is a complementary compression contract: xKV reduces and reconstructs the cached tensors across layers, whereas QKSieve keeps exact K/V independent, constructs a request-local joint QK score coordinate system, and retrieves tokens for each current Query. We therefore report xKV's public-release speed only in its own backend and do not treat its compressed-cache bytes as QKSieve index bytes.

UNIQUE scores KV pages from Key mean and dispersion, while Louver casts threshold retrieval as range search with zero false negatives and ProxyAttn shares representative heads primarily for sparse prefill (Deng et al., 2026; Dehghankar & Asudeh, 2026; Wang et al., 2026). QKSieve instead targets token-level decode retrieval: it spends a fixed byte budget on a request-local score index and measures the subsequent exact-KV consumer. Kascade computes exact top-k at development-set-selected anchor layers and reuses mapped head candidates between anchors (Deshmukh et al., 2025). QKSieve does not reuse realized attention across layers: every layer/head is scored from its current Query in a request-local, quantized joint QK metric.

Low-bit selectors. FIER scans 1-bit token-level Keys and applies exact attention to selected KV (Wang et al., 2025). RaBitQCache and Self-Indexing KVCache use binary codes (Li et al., 2026; Yang et al., 2026b); SVDq assigns mixed precision in an SVD space (Hong et al., 2025); BinaryPC learns binary principal components and shares candidates within GQA groups (Yu et al., 2026). Block-GTQ allocates precision to fixed RoPE blocks in a compressed K/V cache (Liang et al., 2026). QKSieve is distinct in three coupled choices: full-rank Query and Key transforms are jointly constructed and exactly biorthogonal, bits are allocated per layer/head using Query-weighted logit distortion, and the low-bit representation indexes separate exact K/V. Equal-rate and same-active-token controls therefore isolate the selector rather than conflate it with KV quantization or placement.

ParisKV addresses long-generation drift with collision-based candidates and a quantized reranker, including an offloaded million-token regime (Qi et al., 2026). Its drift target motivates our explicit fixed-basis Query-drift audit; unlike ParisKV, our main contract neither clusters Keys nor claims CPU-offloaded retrieval.

SAKI, released while this manuscript was in preparation, derives the same covariance-weighted optimal rank-r bilinear score map for a floating-point low-rank Key index (Zhang, 2026). At $r = d$, its asymmetric factors and our unquantized QK-balanced factors share the same algebraic endpoint; we therefore do not claim the weighted low-rank solution itself as exclusive novelty. QKSieve instead studies how that geometry should be physically quantized and allocated under a mixed-bit byte budget, selected with a packed CUDA path, coupled to an omitted-Value consumer, and validated on downstream generation and complete-system timing rather than recall alone.

Residual approximation and evaluation. MagicPIG and vAttention use sampling when deterministic top-k poorly estimates diffuse attention, with vAttention providing a user-specified statistical guarantee (Chen et al., 2025; Desai et al., 2026). RESA estimates omitted attention from a prefill-derived low-rank prior (Yang et al., 2026a). Residual-Mass Accounting builds fixed-size summary states from learned positive feature maps, subtracts the retrieved support, and merges the residual numerator and denominator with an exact branch, while using exhaustive top-k as an oracle selector (Hoshi et al., 2026). QKSieve-Robust instead uses current-Query proxy scores to estimate both the omitted softmax partition and coefficients in a request-local Value basis, with fixed shrinkage and no learned feature map; the deployable mixed-bit selector remains the primary contribution. We do not claim a probabilistic output-error guarantee or residual completion as an exclusive idea.

Concurrent CertKV separates rank inclusion, retained softmax mass, omitted Value contribution, and downstream utility, then charges one tail-summary slot per head (Yang & Li, 2026). We therefore do not claim that omitted Values matter as a new observation. QKSieve differs operationally: its primary contribution is the jointly balanced, mixed-bit QK selector; its optional residual is a blockwise rank-16 ValueSketch merged with exact candidate softmax rather than a single synthetic KV slot. LongBench evaluates heterogeneous generation and RULER controlled retrieval across lengths (Bai et al., 2024; Hsieh et al., 2024). We keep downstream quality, index bytes, active KV, attention latency, whole-model decode, and cold versus persistent request costs separate; numbers from incompatible models, kernels, or KV placement are not treated as direct speed comparisons.

3 QKSieve

3.1 Problem and QK-balanced coordinates

For Query head h mapped to KV head $g(h)$, dense attention is

$$s_{h,i} = q_h^\top k_{g(h),i}/\sqrt{d}, \quad p_h = \mathrm{softmax}(s_h), \quad o_h = \sum_i p_{h,i} v_{g(h),i}. \tag{1}$$

We seek a Query-dependent set $\widehat{S}_h$ and recompute Equation (1) exactly on original K/V restricted to that set. Every historical token remains addressable.

For each layer and KV head, we sample post-RoPE prompt Keys at stride 32 and the final eight prompt Query positions from all mapped Query heads. Their uncentered second moments are

$$\widehat{C}_k = m_k^{-1} \sum_i k_i k_i^\top, \qquad \widehat{C}_q = m_q^{-1} \sum_j q_j q_j^\top. \tag{2}$$

We shrink the smaller Query sample,

$$S_q = (1-\lambda)\widehat{C}_q + \lambda\, \mathrm{tr}(\widehat{C}_q) I/d, \qquad \lambda = .75, \tag{3}$$

and consistently floor eigenvalues at $10^{-8}\lambda_{\max} + 10^{-12}$ to obtain positive-definite C_k and $\widetilde{C}_q$:

$$C_k = \mathcal{F}(\widehat{C}_k), \qquad \widetilde{C}_q = \mathcal{F}(S_q). \tag{4}$$

No task label or trained parameter enters these estimates.

Let

$$\widetilde{C}_q^{1/2} C_k^{1/2} = U \Sigma V^\top \tag{5}$$

and define

$$A = \widetilde{C}_q^{-1/2} U \Sigma^{1/2}, \qquad D = C_k^{-1/2} V \Sigma^{1/2}. \tag{6}$$

For $q' = A^\top q$ and $k' = D^\top k$,

$$AD^\top = I, \qquad q'^\top k' = q^\top k, \tag{7}$$

while

$$A^\top \widetilde{C}_q A = D^\top C_k D = \Sigma. \tag{8}$$

Thus the full transform has zero score error, and Σ orders coordinates by joint Query–Key score energy rather than Key reconstruction energy.

3.2 Query-weighted mixed-bit index

For $d = 128$, eight contiguous 16-D bands receive $b_g \in \{0, 1, 2, 4, 8\}$ bits per coefficient. An active band stores one FP16 scale per token, equivalent to one metadata bit per coefficient, giving

$$R(b) = 16 \sum_{g=1}^{8} \big(b_g + \mathbf{1}[b_g > 0]\big) \le 240 \text{ bits}. \tag{9}$$

Zero-bit bands are physically absent. The selected symmetric quantizers use max-absolute scale for 2/4/8 bits and sign times mean magnitude at 1 bit.

For band error $e_{i,g}(b) = k'_{i,g} - Q_b(k'_{i,g})$, define empirical Query and error moments $C'_{q,g}$ and $C_{e,g}(b)$. The Cartesian-sample score distortion is exactly

$$D_g(b) = \frac{1}{m_q m_k} \sum_{j,i} (q'^{\top}_{j,g} e_{i,g}(b))^2 = \mathrm{tr}(C'_{q,g} C_{e,g}(b)). \tag{10}$$

The per-layer/head allocation is

$$b^\star = \arg\min_b \sum_g D_g(b_g) \quad \text{s.t.} \quad R(b) \le 240. \tag{11}$$

We enumerate the small feasible set exactly and freeze the result after prefill. The 30-byte Key index adds $30/512 = 5.86\%$ to 16-bit K+V for $d = 128$. Key-MSE, fixed 4-4-2-1, Key-PCA, and random rotations are ablations; the frozen method uses Equation (11).

3.3 Packed selection and exact attention

For history length N, every sparse profile targets

$$B(N) = \min\{N, 1280, \max(256, \lceil .06N \rceil)\}. \tag{12}$$

Each decode Query is transformed and INT8-quantized. The optimized CUDA path samples

$$m(N) = \min\{N, 512, \max(256, \lceil 16/(B(N)/N) \rceil)\} \tag{13}$$

regularly stratified index entries, estimates the target quantile, and scans the packed index once while writing scores above threshold into a bounded buffer. This avoids an N-length score tensor and generic top-k. The reference path instead materializes proxy scores and applies deterministic top-$B(N)$; it isolates representation error from threshold estimation.

For selected positions $\widehat{S}_h$, original resident 16-bit K/V compute

$$\widetilde{p}_{h,i} = \frac{e^{s_{h,i}}}{\sum_{j \in \widehat{S}_h} e^{s_{h,j}}}, \qquad \widetilde{o}_h = \sum_{i \in \widehat{S}_h} \widetilde{p}_{h,i} v_{g(h),i}. \tag{14}$$

Approximation therefore changes only the selected set. There is no exact rerank, router, task rule, sink/recent reservation, or Full fallback.

3.4 Robust omitted-Value estimator

Accurate ranking does not recover a coherent aggregate carried by many small omitted weights. Let $S = \widehat{S}_h$, $T = [N] \setminus S$, and $\widehat{\tau}$ be the proxy threshold. During the same scan, Robust records

$$\widehat{Z}_T = \sum_{i \in T} e^{\widehat{s}_i - \widehat{\tau}}, \qquad \widehat{c}_T = \sum_{i \in T} e^{\widehat{s}_i - \widehat{\tau}} \widehat{c}_i. \tag{15}$$

A request-local rank-16 PCA of stratified Values stores $c_i = P_v^\top (v_i - \mu_v)$ as block-256 affine INT4. Its 64 coefficient bits per token plus block metadata make the total Key-plus-Value index 7.47% of 16-bit K+V. With exact selected numerator u_S, partition Z_S, and maximum m, Robust returns

$$\widehat{o}_\alpha = \frac{u_S + \alpha e^{\widehat{\tau} - m} (\widehat{Z}_T \mu_v + P_v \widehat{c}_T)}{Z_S + \alpha e^{\widehat{\tau} - m} \widehat{Z}_T}, \qquad \alpha = .5. \tag{16}$$

The fixed shrinkage was selected on registered weak cases and then frozen for held-out seeds. Quant omits this estimator; Robust is the length-agnostic main consumer and never changes $B(N)$ or reads additional exact KV.

3.5 Construction and registered profiles

After dense prefill exposes prompt-tail Queries, the method builds (A, D), $b^\star$, packed historical Keys, and, for Robust, the Value sketch. Generation appends one exact K/V and one packed entry per token while keeping the request-local basis fixed. Coord-32 FP16 stores only the first two bands and uses complete proxy top-k. Quant and Robust share the frozen 240-bit allocation and sampled-quantile selector. All profiles share the target budget and original exact K/V, but Coord-32 is not an equal-dimension quantization ablation. CUDA paths fuse Query preparation, packed scanning, candidate compaction, and exact sparse attention; Robust additionally fuses tail accumulation and consumption. The current profile contract, proofs, audit rules, and byte accounting are in Appendix A.

4 Analysis

Exact geometry. Substituting Equations (5) and (6) gives

$$AD^\top = I, \qquad A^\top \widetilde{C}_q A = D^\top C_k D = \Sigma, \tag{17}$$

so the unquantized transform changes no QK score. Moreover, among rank-r bilinear approximations, truncating this SVD minimizes expected squared score error:

$$\min_{\mathrm{rank}(W)\le r} \mathbb{E}(q^\top k - q^\top W k)^2 = \sum_{j>r} \sigma_j^2. \tag{18}$$

Concurrent SAKI derives the same covariance-weighted rank-r endpoint for floating-point low-rank indexing (Zhang, 2026); our method uses its full-rank endpoint before mixed-bit quantization. Proof details and the regularization conditions are in Appendix I.

Why Query-weighted allocation. For quantization error $e = k' - \widehat{k}'$, independence across the empirical Cartesian Query–Key sample gives

$$\mathbb{E}(q'^\top e)^2 = \mathrm{tr}(C'_q C_e), \tag{19}$$

The allocator minimizes the sum of its within-band blocks in Equation (10); this equals the full score MSE when either moment is block diagonal in the chosen bands. Otherwise the omitted cross-band term is explicitly bounded and audited in Equation (32). Key-MSE $\mathrm{tr}(C_e)$ is correct only for isotropic C'_q. Under Query drift $\Delta = C'_{q,\mathrm{dec}} - C'_{q,\mathrm{cal}}$,

$$|\,\mathrm{tr}(\Delta C_e)| \le \|\Delta\|_{\mathrm{op}}\,\mathrm{tr}(C_e). \tag{20}$$

This is a stability diagnostic, not an unconditional held-out guarantee; the auditable cross-band condition is provided in Appendix I.

Ranking and output. Let exact and proxy errors be $\delta_i = s_i - \widehat{s}_i$ and $R_\delta = \max_i \delta_i - \min_i \delta_i$. Any exact top-$B$ token whose boundary margin exceeds R_δ remains selected. The appendix converts centered score RMSE into preserved-core and omitted-mass certificates. For the realized selected set S, let $\epsilon = \sum_{i\notin S} p_i$. Exact recomputation on S yields

$$\|p - \widetilde{p}\|_1 = 2\epsilon, \qquad \|o - \widetilde{o}\|_2 \le \epsilon\ \mathrm{diam}\{v_i\}. \tag{21}$$

Thus selector quality is governed by score margins and omitted probability mass, not set recall alone.

What Robust corrects. Write exact selected and omitted conditional Value means as μ_S, μ_T, their selected mass as p, and Robust estimates as $\widehat{p}, \widehat{\mu}_T$. Its exact error decomposition is

$$\widehat{o}_\alpha - o = (\widehat{p} - p)(\mu_S - \mu_T) + (1 - \widehat{p})(\widehat{\mu}_T - \mu_T), \tag{22}$$

hence

$$\|\widehat{o}_\alpha - o\|_2 \le |\widehat{p} - p|\|\mu_S - \mu_T\|_2 + (1 - \widehat{p})\|\widehat{\mu}_T - \mu_T\|_2. \tag{23}$$

For zero-mean tail-estimation error with variance V and $D = \|\mu_S - \mu_T\|_2^2$, a tail mixture weight w_α has risk

$$\mathbb{E}\|\widehat{o}_\alpha - o\|_2^2 = ((1 - p) - w_\alpha)^2 D + w_\alpha^2 V, \tag{24}$$

and improves over selected-only Quant exactly when

$$0 < w_\alpha < 2(1-p)D/(D+V). \tag{25}$$

Robust therefore addresses an omitted-Value failure that better Key ranking alone cannot remove, while its fixed shrinkage still requires held-out tests.

Cost and crossover. Both deployment profiles scan $\Theta(N)$ low-bit entries; exact attention is targeted by Equation (12). Robust also scans 64 Value coefficient bits/token plus block metadata. If f_t, s_t are directly measured Full and QKSieve step costs and I_{net} is unhidden construction, QKSieve breaks even after G tokens iff

$$\sum_{t=1}^{G}(f_t - s_t) > I_{\text{net}}. \tag{26}$$

For Full non-attention cost t_{base} and attention cost t_{attn}, the whole-model ceiling is

$$\text{speedup} \le (t_{\text{base}} + t_{\text{attn}})/t_{\text{base}}. \tag{27}$$

We therefore report cold, warm, and persistent reuse separately.

5 Experiments

We evaluate one frozen implementation contract. Coord-32 FP16 stores the first two balanced 16-D bands and uses complete proxy top-k; QKSieve-Quant stores all eight bands with the 240-bit qMSE allocation and sampled-quantile selection; QKSieve-Robust additionally uses the rank-16, block-256 INT4 ValueSketch with $\alpha = .5$. All target $B(N) = \min\{N, 1280, \max(256, \lceil .06N \rceil)\}$ positions, retain resident FP16 K/V, and use no learned component, exact-QK reranking, or Full fallback. Measured active KV is reported whenever sampled selection differs from its target.

Downstream quality uses each model's native attention layout and task metric. Controlled timing uses 32-Query/32-KV-head MHA with head dimension 128 on one RTX 3090. Complete-attention latency is a three-seed median; whole-model steady decode uses YaRN-Llama-2-7B-128K and reports the final 48 of 64 greedy tokens. Online-64 additionally includes request-local index construction. Every system number directly times the complete path rather than summing independently measured stages.

5.1 Downstream quality

We run all 16 English LongBench tasks (Bai et al., 2024) on Llama-3.1-8B-Instruct, Qwen3-4B-Instruct-2507, and Mistral-7B-Instruct-v0.3, with 3,750 strict Full/Robust pairs per model. Prompts use official middle truncation at 31,500 tokens and expose the question before selection. Following recent sparse-attention reporting (Feng et al., 2024; Tang et al., 2024), Table 1 aggregates six task domains; all tasks appear in the appendix.

Robust retains 99.98%, 99.73%, and 100.00% of Full macro score on Llama, Qwen, and Mistral, respectively, for a 99.90% three-model mean at 7.22% target KV. This average is not uniform: PassageCount retains 94.53% on Llama and 92.86% on Mistral, and Musique retains 93.17% on Qwen.

We also compare 19 public settings on 800 fixed examples per model. Standard rows retain their registered budgets; a matched control gives FIER, Quest, RaBitQ, and SparQ the exact per-example $B(N)$ of QKSieve. These are quality-only reference selectors, not optimized-runtime claims.

At matched budget, the three-model mean retentions are 99.77% for Robust, 99.86% for FIER, 100.00% for RaBitQ, 99.75% for SparQ, and 92.12% for Quest. Thus task score establishes a near-lossless operating point, not a significant ranking among selectors near 100%. Complete panels, per-task failures, and disjoint post-freeze holdouts appear in Appendices C and D. On 650 paired examples spanning 13 RULER tasks and 4K–128K, the same Robust contract retains 100.33% of Full (95% CI: 99.18–101.75%) at 4.97% measured active attention; the length breakdown is in Appendix E.

Table 1: Complete question-aware LongBench results on three mainstream models, with 3,750 strict Full/Robust pairs per model. Following recent sparse-attention reporting, the main table aggregates six task domains; the 16-task breakdown is in the appendix. Scores are multiplied by 100. Ret. is macro-score retention relative to Full, and Target KV is the sample-mean scheduled KV fraction. Bold marks the better value within each model group.

Model	Method	Single-doc	Multi-doc	Summ.	Few-shot	Synth.	Code	Avg.	Ret.	Target KV
Llama-3.1-8B	Full	44.44	46.21	28.91	69.26	54.17	59.99	49.67	100.00	100.00
	QKSieve Robust	44.20	46.27	29.04	69.22	53.92	60.26	49.66	99.98	7.43
Qwen3-4B	Full	40.91	42.66	25.74	69.13	50.50	59.72	47.23	100.00	100.00
	QKSieve Robust	40.55	42.15	25.85	69.38	50.38	59.58	47.11	99.73	7.39
Mistral-7B	Full	38.56	39.59	28.76	70.71	52.50	59.97	47.36	100.00	100.00
	QKSieve Robust	38.68	39.56	28.81	70.65	52.25	60.09	47.36	100.00	6.85
3-model mean	Full	41.30	42.82	27.80	69.70	52.39	59.89	48.09	100.00	100.00
	QKSieve Robust	41.15	42.66	27.90	69.75	52.18	59.98	48.04	99.90	7.22

Table 2: Strictly paired LongBench quality on three models. Each cell is macro-score retention / active-KV percentage. Every row uses 800 examples across 16 tasks; no result-dependent budget selection is used.

Method	Llama-3.1-8B	Qwen3-4B	Mistral-7B
Full KV	100.00 / 100.00	100.00 / 100.00	100.00 / 100.00
QKSieve Robust	99.92 / 7.13	99.54 / 7.08	99.86 / 6.71
FIER RTN-1 g32 (B=1024)	99.16 / 17.45	100.60 / 17.11	99.58 / 14.89
Quest P16 (B=1024)	96.22 / 17.45	96.20 / 17.11	95.63 / 14.89
SnapKV (B=1024)	96.28 / 17.41	97.85 / 17.08	94.41 / 14.86
PyramidKV (B=1024)	96.09 / 17.41	97.61 / 17.08	94.39 / 14.86
AdaKV-SnapKV (B=1024)	97.04 / 17.41	98.40 / 17.08	95.09 / 14.86
AdaKV-PyramidKV (B=1024)	97.04 / 17.41	98.40 / 17.08	95.09 / 14.86
H2O (20%)	61.23 / 20.00	80.41 / 20.00	78.17 / 20.00
RaBitQ (B=1024)	99.38 / 17.45	100.61 / 17.11	99.57 / 14.89
SparQ-R32 (B=1024)	99.39 / 17.45	99.68 / 17.11	99.47 / 14.89

5.2 Controlled MHA system timing

Robust index construction takes 1.375/1.488/1.839 seconds at 32K/64K/128K, yielding break-even horizons of 69/19/10 generated tokens and Online-64 speedups of .81×/1.31×/2.15×. In a separate persistent-cache audit, warm 32K/64K decode reaches 1.47×/2.50×, four-branch amortized execution reaches 1.22×/2.04×, and cold execution reaches .96×/1.01×. Removing ValueSketch raises steady speed from 1.63× to 2.15× at 32K and from 3.10× to 4.03× at 96K, but lowers pooled PPL retention from 100.52% to 98.58% and from 99.77% to 95.10%, respectively.

5.3 Applicability boundary

The strongest same-path systems result is MHA-specific. On Qwen3-4B GQA at 32K, independent-candidate Robust retains 99.47% PPL quality with 3.95% active KV but reaches only .89× steady and .83× Online-256 decode. A shared-candidate, no-tail Llama prototype reaches 1.09× whole-model decode at 32K and 2.44 × ±.01 complete attention at 64K, but its speed cannot be combined with Robust quality. The bottleneck and both paths are reported in Appendix H. We claim no prefill acceleration; steady decode amortizes request-local construction, and measured short-context slowdowns are retained.

Table 3: Direct RTX 3090 timing under one native-MHA contract. Complete attention includes Query preparation, packed selection, ValueSketch for Robust, and exact sparse attention. Whole-model rows are steady decode.

History	Full	Quant	Robust	Quant speed	Robust speed
Complete attention (ms/step; three-seed median)					
8K	0.1691	0.1327	0.1560	1.27×	1.08×
16K	0.3173	0.1900	0.2250	1.67×	1.41×
32K	0.6163	0.2324	0.2945	2.65×	2.09×
64K	1.2027	0.2646	0.3576	4.55×	3.36×
128K	2.3708	0.3722	0.5757	6.37×	4.12×
Whole-model steady decode (ms/token; fixed audited seed)					
32K	84.18	51.56	63.97	1.63×	1.32×
64K	144.75	52.97	65.25	2.73×	2.22×
128K	268.17	55.41	67.38	4.84×	3.98×

6 Limitations and Conclusion

QKSieve converts request-local joint QK geometry into a training-free mixed-bit retrieval index. Robust retains 99.90% of Full across complete three-model LongBench and 100.33% across 13 RULER tasks; on RTX 3090 MHA, complete attention and steady decode reach 2.09–4.12× and 1.32–3.98× at 32K–128K. It retains exact K/V, adds 7.47% auxiliary storage, and rebuilds the index for new prompts. The strongest same-path systems evidence is MHA-specific; Robust GQA and natural long-generation quality remain open.

AI use statement

Generative AI tools were used to assist with experiment implementation and execution, organization of experimental outputs, and language polishing. The research questions, hypotheses, core method ideas, and scientific decisions were developed by the authors. All AI-assisted code and prose were reviewed against source artifacts and corrected where necessary. The authors take responsibility for the final content, claims, code, and artifacts.

Reproducibility statement

The complete algorithm and frozen operating point are specified in Section 3; proofs and applicability conditions are provided in the appendix. The supplementary artifact records source and data revisions, per-file checksums, launch commands, hardware and software metadata, paired sample identifiers, and bootstrap seeds. It also includes an evidence verifier that rejects results not matching the frozen method contract.

References

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, 2024.

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 International Conference on Machine Learning, pp. 3358–3392, 2025.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. arXiv preprint arXiv:2406.02069, 2024.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Hung-Yueh Chiang, Yash Akhauri, Xilai Dai, Huiqiang Jiang, Yucheng Li, Luis Ceze, Kai-Chiang Wu, and Mohamed S. Abdelfattah. xKV: Cross-layer KV-cache compression via aligned singular vector extraction. In International Conference on Machine Learning, 2026. URL https://arxiv.org/abs/2503.18893.

Yaoqi Chen, Jinkai Zhang, Baotong Lu, Qianxi Zhang, Chengruidong Zhang, Jing Liu, Jingjia Luo, Di Liu, Huiqiang Jiang, Qi Chen, Bailu Ding, Xiao Yan, Jiawei Jiang, Chen Chen, Mingxing Zhang, Cheng Li, Yuqing Yang, Fan Yang, and Mao Yang. RetroInfer: A vector storage engine for scalable long-context LLM inference. Proceedings of the VLDB Endowment, 19(5):1016–1031, 2026. doi: 10.14778/3796195.3796212. URL https://arxiv.org/abs/2505.02922.

Zhuoming Chen, Ranajoy Sadhukhan, Zihao Ye, Yang Zhou, Jianyu Zhang, Niklas Nolte, Yuandong Tian, Matthijs Douze, Leon Bottou, Zhihao Jia, and Beidi Chen. MagicPIG: LSH sampling for efficient LLM generation. In International Conference on Learning Representations, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/6d50d824ae819d5a961c1d8edc15e833-Abstract-Conference.html.

Mohsen Dehghankar and Abolfazl Asudeh. Sparse attention as a range searching problem: Towards an inference-efficient index for KV cache. arXiv preprint arXiv:2605.06763, 2026. URL https://arxiv.org/abs/2605.06763.

Keqi Deng, Shaoshi Ling, Ruchao Fan, and Jinyu Li. UNIQUE: Universal top-k sparse attention for training-free inference and sparsity-aware training. arXiv preprint arXiv:2605.27740, 2026. URL https://arxiv.org/abs/2605.27740.

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, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/55cb562b1f5af71f6707f3ff3c7941e6-Abstract-Conference.html.

Dhruv Deshmukh, Saurabh Goyal, Nipun Kwatra, and Ramachandran Ramjee. Kascade: A practical sparse attention method for long-context LLM inference. arXiv preprint arXiv:2512.16391, 2025. URL https://arxiv.org/abs/2512.16391.

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. arXiv preprint arXiv:2407.11550, 2024.

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.

Yasuto Hoshi, Daisuke Miyashita, and Jun Deguchi. Residual-mass accounting for partial-KV decoding. arXiv preprint arXiv:2604.05438, 2026. URL https://arxiv.org/abs/2604.05438.

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, 2024.

Sergii Kozyrev and Davyd Maiboroda. Minima-KV: Retention-preserving KV cache compression with mixed-format paged attention. arXiv preprint arXiv:2608.23834, 2026. URL https://arxiv.org/abs/2608.23834.

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, 2025.

Wenhao Li, Jinhao Dong, Hailin Zhang, Wenhang Shi, Wei Lu, and Xiaoyong Du. RaBitQ-Cache: Rotated binary quantization for KVCache in long context LLM inference. In International Conference on Machine Learning, 2026.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In Advances in Neural Information Processing Systems, 2024.

Fengfeng Liang, Yuechen Zhang, and Jiaya Jia. Rope-aware bit allocation for KV-cache quantization. arXiv preprint arXiv:2606.24033, 2026.

Xiaolin Lin, Jingcun Wang, Olga Kondrateva, Yiyu Shi, Bing Li, and Grace Li Zhang. CompressKV: Semantic-retrieval-guided KV-cache compression for resource-efficient long-context LLM inference. arXiv preprint arXiv:2606.24467, 2026. URL https://arxiv.org/abs/2606.24467.

Di Liu, Meng Chen, Baotong Lu, Huiqiang Jiang, Zhenhua Han, Qianxi Zhang, Qi Chen, Chengruidong Zhang, Bailu Ding, Kai Zhang, Chen Chen, Fan Yang, Yuqing Yang, and Lili Qiu. RetrievalAttention: Accelerating long-context LLM inference via vector retrieval. arXiv preprint arXiv:2409.10516, 2024a.

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 International Conference on Machine Learning, 2024b.

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, 2025.

Yanlin Qi, Xinhang Chen, Huiqiang Jiang, Qitong Wang, Botao Peng, and Themis Palpanas. ParisKV: Fast and drift-robust KV-cache retrieval for long-context LLMs. In International Conference on Machine Learning, 2026. URL https://openreview.net/forum?id=wxD4wTYQXt.

Luka Ribar, Ivan Chelombiev, Luke Hudlass-Galley, Charlie Blake, Carlo Luschi, and Douglas Orr. SparQ attention: Bandwidth-efficient LLM inference. arXiv preprint arXiv:2312.04985, 2023.

Prajwal Singhania, Siddharth Singh, Shwai He, Soheil Feizi, and Abhinav Bhatele. Loki: Low-rank keys for efficient sparse attention. arXiv preprint arXiv:2406.02542, 2024.

Hanshi Sun, Li-Wen Chang, Wenlei Bao, Size Zheng, Ningxin Zheng, Xin Liu, Harry Dong, Yuejie Chi, and Beidi Chen. ShadowKV: KV cache in shadows for high-throughput long-context LLM inference. arXiv preprint arXiv:2410.21465, 2024.

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

Dongwei Wang, Zijie Liu, Song Wang, Yuxin Ren, Jianing Deng, Jingtong Hu, Tianlong Chen, and Huanrui Yang. FIER: Fine-grained and efficient KV cache retrieval for long-context LLM inference. In Findings of the Association for Computational Linguistics: EMNLP, 2025.

Yixuan Wang, Huang He, Siqi Bao, Hua Wu, Haifeng Wang, Qingfu Zhu, and Wanxiang Che. ProxyAttn: Guided sparse attention via representative heads. In International Conference on Learning Representations, 2026.

Zhiqiang Xie, Zhangheng Huang, Tingwei Huang, Ziyi Xu, Ruiyang Ma, and Christos Kozyrakis. HiSparse: Scaling sparse-attention decoding with hierarchical KV cache management. arXiv preprint arXiv:2608.07009, 2026. URL https://arxiv.org/abs/2608.07009.

Chiwun Yang and Xiaoyu Li. Beyond sparse weights: When is attention compressible? arXiv preprint arXiv:2608.21541, 2026. URL https://arxiv.org/abs/2608.21541.

Weihao Yang, Hao Huang, Ningke Li, Shihao Wang, Darong Yang, Yanqi Pan, Wen Xia, Shiyi Li, and Xiangyu Zou. RESA: Bringing back what sparse attention ignores with residual estimation. In International Conference on Learning Representations, 2026a.

Xu Yang, Jiapeng Zhang, Dongyang Zhao, Guo Chen, and Zhuo Tang. Self-indexing KV-Cache: Predicting sparse attention from compressed keys. In AAAI Conference on Artificial Intelligence, volume 40, pp. 27675–27683, 2026b.

Hengshuai Yao, Xing Chen, Ahmed Murtadha, and Guan Wang. Thin keys, full values: Reducing KV cache via low-dimensional attention selection. arXiv preprint arXiv:2603.04427, 2026.

Daohai Yu, Zhanpeng Zeng, Keyu Chen, Wenhao Li, Zhifeng Shen, Luxi Lin, Ruizhi Qiao, Xing Sun, and Rongrong Ji. Training-free hashing-based attention via binary principal components. In International Conference on Machine Learning, 2026. URL https://openreview.net/forum?id=4spHlgHY9x.

Lin Zhang. SAKI: Score-aware low-rank key indexing with random-matrix noise correction for KV retrieval. arXiv preprint arXiv:2608.03228, 2026. URL https://arxiv.org/abs/2608.03228.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. In Advances in Neural Information Processing Systems, 2023.

A Current Frozen Contract

This appendix deliberately excludes artifacts produced by pre-freeze methods. The admissible profiles are listed in Table 4.

Table 4: Current experimental profiles. "Target" is strict for Coord-32 and a sampled-quantile target for Quant and Robust.

Profile	Key dims	Key bits	Value rate	Target	Tail correction
Coord-32 FP16	32	512	0	$B(N)$	none
QKSieve-Quant	128	240	0	$B(N)$	none
QKSieve-Robust	128	240	66	$B(N)$	rank-16 INT4

The Value rate includes 64 code bits and two amortized block-scale bits per token/KV head. All profiles use request-local QK-balanced coordinates and resident original 16-bit K/V. Selected positions receive exact QK–softmax–AV. None uses a router, task identifier, recent/sink reservation, exact-QK reranking, or Full fallback. Robust uses block size 256 and $\alpha = .5$.

B Detailed LongBench Results

Table 5: Complete 16-task LongBench breakdown on three models. Full and Robust use the same 3,750 samples for each model; scores are multiplied by 100. Bold marks the better value within each model group.

		Single-doc QA			Multi-doc QA			Summarization			Few-shot			Synthetic		Code		
Model	Method	NrtvQA	Qasper	MF-en	HotpotQA	2WikiMQA	Musique	GovReport	QMSum	MultiNews	TREC	TriviaQA	SAMSum	PCount	PRe	LCC	RB-P	Avg.
Llama-3.1-8B	Full	29.83	46.55	56.93	58.10	48.96	31.57	34.51	25.23	26.98	72.50	91.65	43.64	8.83	99.50	63.21	56.77	49.67
	QKSieve Robust	29.38	46.46	56.77	58.10	49.05	31.64	34.63	25.43	27.07	72.50	91.48	43.67	8.35	99.50	63.40	57.12	49.66
Qwen3-4B	Full	27.08	44.71	50.94	58.89	43.53	25.56	30.67	22.46	24.10	74.50	87.47	45.41	1.00	100.00	63.37	56.06	47.23
	QKSieve Robust	26.43	44.53	50.70	58.95	43.70	23.82	30.83	22.65	24.06	74.50	88.14	45.50	0.75	100.00	63.65	55.51	47.11
Mistral-7B	Full	27.01	38.15	50.53	52.09	38.67	28.02	34.19	25.63	26.45	76.00	88.59	47.53	7.00	98.00	59.49	60.45	47.36
	QKSieve Robust	27.47	38.02	50.55	51.80	38.61	28.26	33.99	25.65	26.79	76.00	88.64	47.32	6.50	98.00	59.60	60.58	47.36

C Complete Public-Baseline Panels

The following tables use 800 strictly paired examples per model. The first panel preserves each public method's registered budget. The second isolates selector quality by giving four retrieval baselines exactly QKSieve's frozen per-example $B(N)$. Each cell reports Full-relative macro retention / measured active-KV percentage.

Table 6: Complete fixed-budget LongBench comparison. Each cell is macro-score retention / active-KV percentage relative to paired Full KV.

Method	Llama-3.1-8B	Qwen3-4B	Mistral-7B
Full KV	100.00 / 100.00	100.00 / 100.00	100.00 / 100.00
QKSieve Robust	99.92 / 7.13	99.54 / 7.08	99.86 / 6.71
FIER RTN-1 g32 (B=1024)	99.16 / 17.45	100.60 / 17.11	99.58 / 14.89
Quest P16 (B=1024)	96.22 / 17.45	96.20 / 17.11	95.63 / 14.89
Quest P16 (B=2048)	98.29 / 32.30	97.83 / 31.72	96.52 / 28.30
AdaKV-SnapKV (B=512)	93.25 / 8.88	94.73 / 8.71	91.64 / 7.52
AdaKV-SnapKV (B=1024)	97.04 / 17.41	98.40 / 17.08	95.09 / 14.86
H2O (B=1024)	58.85 / 17.41	72.87 / 17.08	71.31 / 14.86
H2O (20%)	61.23 / 20.00	80.41 / 20.00	78.17 / 20.00
SnapKV (B=512)	93.63 / 8.88	94.82 / 8.71	90.10 / 7.52
SnapKV (B=1024)	96.28 / 17.41	97.85 / 17.08	94.41 / 14.86
PyramidKV (B=512)	93.58 / 8.88	94.74 / 8.71	90.11 / 7.52
PyramidKV (B=1024)	96.09 / 17.41	97.61 / 17.08	94.39 / 14.86
AdaKV-PyramidKV (B=512)	93.25 / 8.88	94.73 / 8.71	91.64 / 7.52
AdaKV-PyramidKV (B=1024)	97.04 / 17.41	98.40 / 17.08	95.09 / 14.86
RaBitQ (B=1024)	99.38 / 17.45	100.61 / 17.11	99.57 / 14.89
SparQ-R32 (B=1024)	99.39 / 17.45	99.68 / 17.11	99.47 / 14.89

Table 7: Matched-budget LongBench comparison. Public selectors use exactly the same frozen per-sample B(N) as QKSieve. Each cell is quality retention / active-KV percentage.

Method	Llama-3.1-8B	Qwen3-4B	Mistral-7B
Full KV	100.00 / 100.00	100.00 / 100.00	100.00 / 100.00
QKSieve Robust	99.92 / 7.13	99.54 / 7.08	99.86 / 6.71
FIER, matched $B(N)$	99.05 / 7.13	100.83 / 7.08	99.69 / 6.71
Quest, matched $B(N)$	92.51 / 7.13	89.25 / 7.08	94.61 / 6.71
RaBitQ, matched $B(N)$	99.48 / 7.13	100.49 / 7.08	100.01 / 6.71
SparQ-R32, matched $B(N)$	99.67 / 7.13	100.08 / 7.08	99.49 / 6.71

Table 8: Per-task QKSieve-Robust quality retention (%) relative to strictly paired Full KV. Values expose task-specific failures hidden by macro averages.

Task	Llama-3.1-8B	Qwen3-4B	Mistral-7B
2wikimqa	100.39	100.00	100.00
gov_report	101.94	100.11	98.43
hotpotqa	100.00	100.01	100.00
lcc	98.03	101.16	100.06
multi_news	101.24	99.55	104.48
multifieldqa_en	100.45	98.91	97.35
musique	100.21	85.39	102.21
narrativeqa	96.43	98.23	105.99
passage_count	100.00	100.00	100.00
passage_retrieval_en	100.00	100.00	100.00
qasper	98.44	97.28	98.45
qmsum	98.99	103.45	100.53
repobench-p	101.48	97.17	99.51
samsum	100.64	100.02	97.35
trec	100.00	100.00	100.00
triviaqa	100.00	101.85	100.00

D Post-Freeze Holdouts

We evaluate examples 51–60 of every LongBench task after freezing the method; these 160 pairs are disjoint from the 50-example public-comparison subset. On Llama, Robust scores .49073 versus Full's .48873, or 100.41% retention at 7.08% active KV (task-bootstrap 95% CI: 99.43–101.72%). Matched-budget FIER, RaBitQ, SparQ, and Quest retain 100.92%, 100.47%, 100.09%, and 93.83%. On Qwen, Robust scores .45997 versus .45762, or 100.51% at 7.03% active KV (95% CI: 100.04–101.30%); the corresponding baselines retain 99.44%, 100.30%, 99.84%, and 83.54%. The ranking reversal cautions against ranking near-100% selectors from one small subset; the stable observation is transfer of the frozen operating point without retuning.

E RULER Length Breakdown

Every task–length cell uses strictly paired Full and Robust examples, with no Full fallback. Harness timing is excluded because methods can terminate after different numbers of generated tokens.

Table 9: Formal RULER results for frozen QKSieve-Robust over 13 tasks, 4K–128K, and 650 strict Full/Robust pairs. Scores and retention are multiplied by 100; intervals use 10,000 paired bootstrap resamples over task–length cells. Mean measured active attention is 4.97%, with zero Full fallback.

Length	Full	QKSieve	Retention	Paired 95% CI
4K	88.31	87.90	99.54	[98.85, 100.00]
8K	88.76	88.71	99.94	[99.28, 100.79]
16K	87.81	88.90	101.24	[99.06, 105.35]
32K	83.42	83.17	99.69	[99.02, 100.00]
64K	80.92	80.92	100.00	[100.00, 100.00]
128K	75.72	77.03	101.73	[95.21, 111.35]
Overall	84.16	84.44	100.33	[99.18, 101.75]

F Teacher-Forced Long-Generation Drift

We freeze the production eight-Query transform from a 32K prompt and process 4,096 real continuation tokens from each of six 20 Newsgroups topics. The audit records five layers and all Query heads at 13 registered positions, yielding 62,400 retrieval conditions. Exact K/V and the 240-bit Key index are unchanged; every Query selects 1,280 prompt positions. Confidence intervals cluster by topic rather than treating correlated layers and heads as independent.

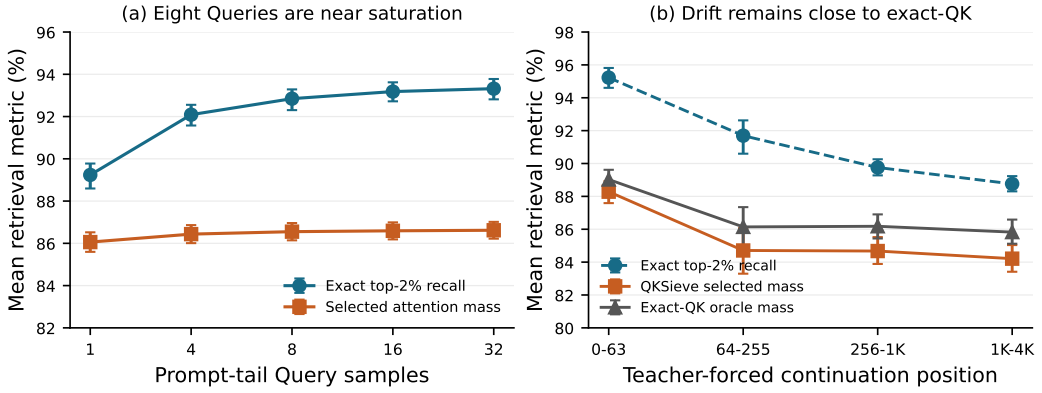


Figure 2: Prompt-tail calibration and teacher-forced Query drift. Error bars are 95% topic-cluster bootstrap intervals over six topics (10,000 draws, fixed seed). The exact-QK oracle uses the same 1,280-token budget.

Eight prompt Queries are close to the 32-Query empirical reference: aggregate top-2% recall is 92.85% versus 93.32%, while selected attention mass is 86.55% versus 86.62%. At continuation positions 1K–4K, recall falls to 88.76% (95% CI 88.31–89.23%), but selected mass remains 84.21% (83.41–85.04%) versus 85.83% (85.12–86.59%) for exact-QK selection. The paired oracle gap is 1.62 points (1.45–1.80). Although relative raw/projected covariance drift reaches .704/1.042, frozen allocation agrees with the local oracle on 82.66% of bands and incurs 1.072× qMSE regret. Thus drift is measurable but most late-position mass loss is already present under exact-QK top-1,280.

This is a mechanism audit, not a generation-quality claim: later Queries are teacher-forced, all retrieval is evaluated against fixed prompt Keys, and Keys appended during continuation are excluded.

G Matched Public-Selector Controls

G.1 Paired downstream quality screen

Table 10: Strictly paired public-baseline LongBench screen on three models. Each row contains 10 examples from each of 16 tasks; scores are multiplied by 100. Ret. is macro-score retention relative to the same-model Full row, with task-cluster paired-bootstrap 95% intervals in parentheses (10,000 draws). B and KV are sample means. This table compares downstream quality, not KVPress harness latency.

Model	Method	Score	Ret. [95% CI]	B	KV
Llama-3.1-8B	Full	49.06	100.00 [100.00, 100.00]	10649	100.00%
	QKSieve Robust	49.10	100.08 [99.26, 101.09]	636	7.35%
	FIER RTN-1 g32, B=1024	48.62	99.10 [96.50, 101.05]	1019	17.63%
	Quest-P16, B=1024	48.52	98.90 [94.97, 103.40]	1019	17.63%
	Quest-P16, B=2048	47.83	97.49 [93.42, 102.37]	2007	32.60%
	AdaKV-SnapKV, B=512	46.03	93.82 [87.98, 98.31]	512	9.24%
	AdaKV-SnapKV, B=1024	48.77	99.41 [94.88, 103.80]	1019	17.59%
	H2O, B=1024	27.73	56.51 [36.65, 76.26]	1019	17.59%
	H2O, 20%	29.77	60.68 [37.27, 83.54]	2131	20.00%
Qwen3-4B	Full	45.84	100.00 [100.00, 100.00]	10932	100.00%
	QKSieve Robust	46.49	101.42 [99.65, 103.72]	652	7.30%
	FIER RTN-1 g32, B=1024	48.11	104.96 [100.92, 110.62]	1019	17.23%
	Quest-P16, B=1024	46.52	101.48 [96.80, 106.85]	1019	17.23%
	Quest-P16, B=2048	47.41	103.43 [100.89, 106.62]	2009	31.98%
	AdaKV-SnapKV, B=512	43.49	94.88 [90.39, 97.84]	512	9.04%
	AdaKV-SnapKV, B=1024	45.33	98.90 [95.74, 102.09]	1019	17.19%
	H2O, B=1024	32.39	70.66 [53.36, 87.35]	1019	17.19%
	H2O, 20%	36.35	79.31 [64.34, 93.93]	2188	20.00%
Mistral-7B	Full	49.94	100.00 [100.00, 100.00]	12279	100.00%
	QKSieve Robust	49.99	100.11 [99.08, 101.15]	721	6.88%
	FIER RTN-1 g32, B=1024	50.02	100.16 [97.07, 103.07]	1020	15.06%
	Quest-P16, B=1024	48.56	97.24 [93.84, 100.15]	1020	15.06%
	Quest-P16, B=2048	49.33	98.77 [95.45, 100.74]	2025	28.53%
	AdaKV-SnapKV, B=512	46.04	92.19 [83.51, 99.35]	512	7.82%
	AdaKV-SnapKV, B=1024	47.84	95.80 [89.84, 100.14]	1020	15.03%
	H2O, B=1024	35.45	70.99 [51.64, 87.77]	1020	15.03%
	H2O, 20%	39.24	78.58 [60.60, 95.62]	2457	20.00%

FIER and Quest use audited equation/reference selectors inside the common generation consumer. AdaKV uses KVPress 0.5.4 with an exact per-Query-head quality mask where its released fake-Key separator is undefined; H2O uses an online heavy-hitter/recent-cache implementation. These rows compare task quality only. Their harness latency and memory are not used as optimized system measurements.

Table 11: Matched selector screening on 160 strict Llama-3.1-8B LongBench pairs (10 per task). Sparse rows share the active-token schedule and original KV consumer. Baselines are audited equation/reference implementations, not claimed optimized-system timings.

Method	Index bits/KV head	Active KV	Full-relative quality
Full	0	100.000%	100.000%
QKSieve reference	240	7.440%	99.789%
FIER RTN-1 g32	256	7.440%	100.010%
Quest-P16, page-rounded	256	7.562%	92.217%
RaBitQ RTN-1	224	7.440%	100.012%
SparQ-R32, complete formula	0	7.440%	99.950%
UNIQUE-P8, equation reference	258	7.494%	93.193%

The QKSieve row uses complete proxy top-k and therefore isolates selector representation from sampled-quantile and ValueSketch effects; it is not the Robust downstream row. Quest and UNIQUE load complete pages, hence their slightly different active fractions. The sample

size is adequate to reject broken operating points but not to establish small superiority margins near 100%; the preceding three-model table is the separate generation-level comparison with each method's released or audited quality path.

On an orthogonal Qwen3-4B retrieval audit spanning 18,432 held-out layer/head/Query conditions at 32K, QKSieve's 240-bit reference attains 75.70% exact top-2% recall and 87.30% attention mass, versus 47.90% and 81.61% for the 256-bit FIER reference. This mechanism result explains a better equal-rate selector but is not a downstream task score.

G.2 Matched MHA latency against FIER

Table 12: Direct FP16 complete-attention latency on one RTX 3090. All rows use batch one, 32 Query/32 KV heads, $d = 128$, resident FP16 K/V, the same target schedule, and the same split-8 exact sparse consumer. Times are three-seed medians after 10 warmup and 40 measured iterations. FIER is our audited RTN-1 g32 CUDA reference, not a timing claim for the authors' runtime.

History	FIER ms	Quant ms	Robust ms	Quant/FIER	Robust/FIER
8K	0.1816	0.1327	0.1560	1.37×	1.16×
16K	0.2951	0.1900	0.2250	1.55×	1.31×
32K	0.4647	0.2324	0.2945	2.00×	1.58×
64K	0.7329	0.2646	0.3576	2.77×	2.05×
128K	1.3322	0.3722	0.5757	3.58×	2.31×

At 32K, Quant and Robust are respectively 2.00× and 1.58× faster than this matched FIER reference; the Robust advantage rises to 2.31× at 128K. This table measures the complete attention call, including each selector and Robust's ValueSketch, but excludes model layers outside attention and one-time index construction. We therefore use it as a same-harness kernel comparison, not as a replacement for FIER's published end-to-end results.

G.3 Official public-backend timing

Table 13: Isolated-node RTX 3090 reproductions using tracked official benchmark paths. Every speedup divides by Full from the same release; raw milliseconds are not compared across backends. All rows use batch one.

Release path	Scope	History	Budget	Speedup
xKV-SR	Attention	16K	1,280	.432×
xKV-SR	Attention	32K	1,280	.680×
xKV-SR	Attention	64K	1,280	1.128×
xKV-SR	Four decoder layers	64K	1,280	1.002×
Quest	Complete CausalLM	8K	1,024	1.160×
Quest	Complete CausalLM	16K	1,024	1.340×
RocketKV	Compiled decode	16K	512	1.203×
RocketKV	Compiled decode	31.7K	512	1.766×

xKV uses commit 05d91ec, 32 warmup and 128 measured iterations over three repeats; its advertised large-batch throughput regime is outside this single-request latency control. Quest commit 01c1623 uses 32 warmup and 128 measured decode tokens over three repeats. RocketKV commit 63637c8 generates 256 tokens in official compiled gpt-fast; its 31.7K B1024 prefill exceeds 24 GB and is omitted. A separate physical- cache control on Llama-3.1-8B reaches 1.17–1.54× at 32K across SnapKV/PyramidKV and B512/B1024. These results show why retained-token ratio alone is not a system-speed claim.

H Native-GQA Boundary Details

H.1 Frozen Robust on the same path

We first retain the complete frozen method on Qwen3-4B: four Query heads share one physical Key index but keep independent candidates and Query-conditioned ValueSketch coefficients. Quality uses four deterministic LongBench-derived streams with 64 consecutive tokens each. Speed uses three 256-token runs; each Full–Robust pair executes serially on the same RTX 3090, and the table reports the paired median. Every Full run records 9,216 calls to compact-GQA SDPA with 32 Query and eight KV heads.

Table 14: Frozen Robust at 32K under a native-GQA Full baseline. Online-256 includes request-local index construction. Auxiliary storage is relative to resident FP16 K+V.

Path	PPL ret.	Active KV	Auxiliary	Steady	Online-256
Native-GQA Full	100.000%	100.000%	0	42.574 ms	42.698 ms
Robust-GQA4	99.474%	3.946%	7.399%	47.675 ms	51.252 ms

The steady and Online-256 speedups are therefore .893× and .833×; the predeclared 1.05× admission threshold fails. An independent CUDA profile attributes 19.845 ms/token to ValueSketch, versus 4.446 to exact sparse attention, 3.911 to retrieval, 2.984 to Key append, and 2.446 to Query preparation. Thus the quality mechanism transfers to GQA, but the current Value-tail consumer does not provide a 32K speedup.

H.2 Shared-candidate no-tail prototype

For each of the eight KV heads in Llama-3.1-8B, one fused scan standardizes the proxy scores of its four Query heads, takes their per-token maximum, and emits one shared candidate set. A shared exact-attention consumer reads each selected K/V once for the four Queries. The target is $\min\{N, 1920, 64\lceil \max(256, .06N)/64 \rceil\}$; this prototype uses no Value-tail correction or dense fallback.

Table 15: Single-RTX-3090 native-GQA whole-model decode. Full uses PyTorch SDPA with enable_gqa=True and never materializes repeated K/V. Both paths are fixed-shape FP16 CUDA Graphs at batch size one. PPL retention uses 256 teacher-forced tokens from four fixed LongBench-derived streams.

History	Target	Active	Full	QKSieve	Speed	PPL ret.
8K	512	6.26%	22.38 ms	22.84 ms	0.98×	100.07%
16K	1,024	6.26%	24.24 ms	23.98 ms	1.01×	99.59%
32K	1,920	5.80%	26.55 ms	24.43 ms	1.09×	101.80%

The complete attention Graph includes Query projection and quantization, the shared selector, and exact sparse attention. Over three seeds it takes .1310 ms versus .1886 ms for native-GQA Full at 32K ($1.44 \times \pm.03$), and .1358 ms versus .3313 ms at 64K ($2.44 \times \pm.01$). Proxy top-k recall is 92.51% and 91.49%, respectively. Longer whole-model rows require model sharding on 24GB GPUs and are not inferred from these attention measurements.

I Proof Details

I.1 Biorthogonal score preservation

Let $C_q, C_k \succ 0$ and take $C_q^{1/2} C_k^{1/2} = U \Sigma V^\top$. Define

$$A = C_q^{-1/2} U \Sigma^{1/2}, \qquad B = C_k^{-1/2} V \Sigma^{1/2}. \tag{28}$$

Then $AB^\top = I$. Hence, with $q' = qA$ and $k' = kB$,

$$q'k'^\top = qAB^\top k^\top = qk^\top. \tag{29}$$

Moreover, $\mathbb{E}[q'^\top q'] = \mathbb{E}[k'^\top k'] = \Sigma$ after matching row/ column conventions. The jth coordinate therefore contributes the same expected Query and Key energy σ_j and the unquantized transform itself introduces no score approximation.

I.2 Quantized score error

Write $\widehat{k}' = k' + e$. Conditional on e,

$$\mathbb{E}_q[(q'e^\top)^2 \mid e] = e\, C_{q'} e^\top. \tag{30}$$

In balanced coordinates $C_{q'} = \Sigma$. If quantization acts independently on disjoint 16-D bands g, the expected score distortion is

$$D(b) = \sum_g D_g(b_g), \qquad D_g(b_g) = \mathbb{E}[e_g \Sigma_g e_g^\top]. \tag{31}$$

The frozen dynamic program enumerates $b_g \in \{0, 1, 2, 4, 8\}$ under the packed 240-bit rate. Separability makes this the exact minimizer of the stated surrogate over the finite allocation set; it is not a guarantee of minimum downstream task loss.

Without block diagonality, write the omitted cross-band contribution as $R_{\text{cross}} = \sum_{g \neq h} \operatorname{tr}(C'_{q,gh} C_{e,hg})$. Frobenius Cauchy–Schwarz gives the directly auditable bound

$$|R_{\text{cross}}| \leq \sum_{g \neq h} \|C'_{q,gh}\|_F \, \|C_{e,hg}\|_F. \tag{32}$$

The additive allocator is exact for its surrogate when this term is zero; a large measured right-hand side is evidence that independent band allocation is insufficient for that request.

I.3 Ranking stability

Let exact scores be s_i, proxy errors satisfy $|\widehat{s}_i - s_i| \leq \epsilon$, and let $\gamma = s_{(k)} - s_{(k+1)}$. If $\gamma > 2\epsilon$, every exact top-k element remains above every non-top-k element under the proxy, so the two sets are identical. More generally, only tokens whose exact score lies within 2ϵ of the boundary can cross it. This explains why logit MSE is useful but insufficient: long histories can create many near-boundary tokens even when average error is small.

I.4 Omitted Value mass

For attention weights p_i and selected set S, let $m = \sum_{i \notin S} p_i$. With $\|v_i\| \leq V_{\max}$,

$$\left\| \sum_i p_i v_i - \sum_{i \in S} p_i v_i \right\| \leq m V_{\max}. \tag{33}$$

Renormalizing selected weights changes the constant but not the dependence on omitted mass. Robust estimates both the omitted partition and its projected Value numerator, rather than assuming the omitted numerator is zero. The rank-16 INT4 sketch is biased after quantization and scaling by α; the bound motivates the correction but does not assert unbiased downstream output.

J Concurrent Work

SAKI was posted to arXiv on August 4, 2026, while this manuscript was in preparation (Zhang, 2026). ICLR 2027 guidance does not require comparison to arXiv-only work, but encourages citation of known recent work.[1] We therefore cite it explicitly. Both works derive the covariance-weighted rank-r bilinear endpoint; QKSieve instead claims the full-rank mixed-bit allocation, packed retrieval path, omitted-Value correction, and downstream and complete-system evaluation.

[1] https://iclr.cc/Conferences/2027/ReviewerGuidelines