QUANTIZE, EVICT, OR BOTH? WHEN MIXED KV-CACHE ALLOCATION PAYS, AND WHY ITS DISTORTION PROXY MISLEADS

Anonymous authors
Paper under double-blind review

ABSTRACT

A key-cache budget can be spent by quantizing every token, by evicting tokens, or by mixing zero and finite precisions across tokens; values stay exact. Hybrid systems exist. We ask when this mixed interior beats both endpoints, and whether an answer measured on attention-output distortion carries over to end tasks. If eviction is treated as a zero-rate tier, a finite-bit tier is dominated when its logit-noise cost exceeds the cost of zero rate. The share of heads whose 2-bit tier is dominated orders the interior's exactly recomputed output-error advantage across six LLMs and 25 model–context cells (Spearman $\rho = -0.978$; -0.94 controlling for model). Where the interior stops paying, however, depends on the architecture, and longer contexts move a model toward eviction. Over 4,096 decode tokens a head's best policy family is stable, but its token allocation goes stale within a few steps. On retrieval tasks we compare against three dense quantizers (TurboQuant, KIVI, KVQuant) and six eviction methods (H2O, SnapKV, DropKV, Ada-KV, LaProx, OBCache). Routed per KV head, the allocator that is optimal under a local surrogate of output distortion beats every eviction method (mean score 0.77 vs. 0.60), partly by sending 1–25% of heads to dense quantization; its mixed interior alone scores 0.47. Yet dense quantization at the same key code bits does better on all three end-task models, and the best quantizer depends on the model. TurboQuant wins on Llama-3.1-8B (0.99 vs. 0.65 for the allocator) and Mistral-7B (0.95 vs. 0.70). On Qwen3-8B TurboQuant loses to the allocator (8k–32k: 0.87 vs. 0.96), and per-channel KIVI or KVQuant win (0.99). Output distortion ranks methods only weakly and overstates the cost of rotated quantization noise. The allocator mainly fails by truncating multi-token answers; smoothing its token scores over neighboring positions largely removes this failure (Llama, 8k–32k: about 40% to 10% of answers). At both levels, the choice of baselines and of when the cache is compressed moves verdicts more than prompt sampling does.

1 INTRODUCTION

At long context, the KV cache can limit batch size and decode throughput. A cache budget admits at least three actions: quantize every token to one low precision (Liu et al., 2024; Hooper et al., 2024; Zandieh et al., 2025), keep selected tokens and evict the rest (Zhang et al., 2023; Liu et al., 2023; Li et al., 2024; Xiao et al., 2024), or mix zero and finite precisions across tokens. We call the last option *mixed allocation*, or the *interior* between the two endpoints, and study it on the key cache, with values kept exact. The interior is no longer hypothetical: recent systems combine pruning with multiple precision levels (Zhang et al., 2025b; Wang et al., 2026). Benchmarks also find that dense quantization is more robust than token dropping across tasks (Yuan et al., 2024). Three questions remain open: whether the extra interior choices help for a given model and context, whether a simpler endpoint already suffices, and whether the proxy behind such allocators predicts what users observe.

This paper is a measurement study of those questions; it does not propose a better compressor. We build SIEVE, the allocator that is optimal under a local surrogate of attention-output distortion, and use it as an instrument. It shows how large the interior's advantage can be under that objective, where the advantage disappears, and whether it survives the step to task accuracy. One model already shows why the answer depends on context. From 8k to 128k tokens, the share of Llama-3.3-70B's heads

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.

Table 1: Summary of findings, their evidence, and what they do *not* show.

	Observation	Evidence	Boundary
F1	Tier extinction (D_2) orders the interior's output-error advantage	25 cells, $\rho = -0.978$; Fig. 1a	crossing is per architecture, not universal
F2	Longer context moves a fixed model toward the eviction corner	6 trajectories, 4k–256k; Fig. 1b	mostly absolute length, small RoPE-cap residual
F3	Policy family is slow; token allocation is fast	6 cells, 4,096 tokens; Fig. 2b	output error only
F4	Routed per KV head, the output-error optimum beats every eviction method, but a model-matched dense quantizer beats it	36 task cells (8 methods) + 28 at 8k–32k and 10 on Mistral (12 methods); Table 2, Fig. 3a	interior alone does not beat eviction; best quantizer differs by model
F5	Output distortion ranks budgets and contexts well and methods weakly	within-cell $\rho = -0.29$ (8k–32k, 12 methods); Fig. 3b	overstates the cost of rotated noise
F6	Token-granular allocation truncates multi-token answers; positional smoothing largely repairs it	36% vs. 1–11% (128k); 39–44% $\rightarrow$ 9–11% (8k–32k); Fig. 3c	repair shown at 8k–32k on Llama; partial on Mistral
F7	The comparison set and information contract move verdicts more than prompt sampling does	4 contrasts; Table 3	retrieval tasks; sampling from 4 cells

whose 2-bit tier is dominated by eviction (D_2, §2) rises from 5.5% to 41.4%. Over the same range, the interior's hindsight portfolio gain (G_+, §3) falls from $3.17\times$ to $1.40\times$, and the share of heads where the interior beats both endpoints by $2\times$ (the *band*) falls from 87.4% to 21.3%. A verdict at one context need not hold at another.

Findings. We organize the paper around seven findings (Table 1). F1–F3 concern attention-output distortion, where the interior is well defined and its error can be measured exactly (§4). F4–F6 test whether these regimes carry over to retrieval accuracy on three end-task models (Llama-3.1-8B, Qwen3-8B, Mistral-7B), against three dense quantizers (TurboQuant, KIVI, KVQuant (Zandieh et al., 2025; Liu et al., 2024; Hooper et al., 2024)) and six eviction methods (H2O, SnapKV, DropKV, Ada-KV, LaProx, OBCache (Zhang et al., 2023; Li et al., 2024; Zhang et al., 2026a; Feng et al., 2025; Mai & Kim, 2026; Gu et al., 2026); §5). The answer is mixed, and the way it fails is informative. F7 spans both levels: the choice of baselines and information contract, such as when the cache is compressed, moves verdicts more than prompt sampling does.

Contributions.

1. **A common action space and a measurable coordinate.** We place within-head key-token precision and eviction under one local output-distortion surrogate. We separate the exact set-eviction error from its zero-rate approximation, and derive a tier-extinction coordinate D_2 (§2).
2. **A regime map for output distortion.** Using exact recomputation, and giving both sides of each comparison the same information, we show where the interior pays, how context moves a model across regimes, and the two timescales on which a controller would have to act (F1–F3).
3. **An end-task audit of the map.** Against quantization and eviction baselines at matched key code bits, we identify where output-distortion optimality carries over to task accuracy, where it does not, the mechanism of its main failure (F4–F6), and what moves verdicts (F7).

Organization. Section 2 derives the framework and D_2, Section 3 the protocol, and Sections 4–5 the findings; Sections 6–8 give implications, related work, and limitations.

2 JOINT TOKEN-RATE ALLOCATION AS A COMMON FOUNDATION

Scope and local quantization surrogate. For one query q, keys k_i, and values v_i, let $s_i = q^\top k_i/\sqrt{d}$, $a = \mathrm{softmax}(s)$, and $o = \sum_i a_i v_i$. Quantizing key i perturbs its logit by δ_i, and the first-order output change is $\Delta o \approx \sum_i a_i(v_i - o)\delta_i$. If residual perturbations are zero mean and cross-token covariance terms vanish,

$$\mathbb{E}\|\Delta o\|^2 \approx \sum_i w_i d_{b_i}, \qquad w_i = a_i^2\|v_i - o\|^2, \quad d_b = \mathrm{Var}(\delta_b). \tag{1}$$

Softmax cancels only a shift common to all logits; the surrogate ignores token-dependent bias and covariance. We therefore use Eq. 1 only to *choose* allocations, never to score them.

Exact eviction versus a zero-rate surrogate. For an evicted set E with mass $A_E = \sum_{i\in E} a_i$, renormalizing the surviving attention gives exactly

$$\|o_{\setminus E} - o\|^2 = \frac{\left\|\sum_{i\in E} a_i(v_i - o)\right\|^2}{(1 - A_E)^2}. \tag{2}$$

The denominator and the cross-token inner products make set eviction nonseparable. For a single token j the error magnitude is $a_j\|v_j - o\|/(1 - a_j)$, as in CAOTE (Goel et al., 2025). To fit zero rate into a token-separable allocator, we assume the evicted mass is small and drop the cross terms, giving $\widetilde{D}_0(E) = \sum_{i\in E} w_i$; this fixes the *absolute* cost of zero rate at $d_0 = 1$. Eviction is thus a zero-rate action *inside the surrogate* only; it is not exactly equivalent to independent per-token noise.

The implemented discrete allocation. For tiers $\mathcal{B} = \{0, 1, 2, 3, 4, 5, 6, 8\}$ and an average key budget B, we solve

$$\min_{b_i\in\mathcal{B}} \sum_i w_i d_{b_i} \quad \text{s.t.} \quad \sum_i b_i \le BL, \qquad b_i^\star(\lambda) = \arg\min_{b\in\mathcal{B}}\{w_i d_b + \lambda b\}, \tag{3}$$

and a bisection on λ meets the budget. Relaxing bits to continuous values with $d_b \propto 4^{-b}$ gives $b_i^\star = \log_2 a_i + \log_2\|v_i - o\| + c$, the familiar reverse-water-filling shape; experiments use Eq. 3.

Two coordinates, not one universal phase boundary. If $d_b \ge d_0$, tier b spends bits without lowering the surrogate cost below that of eviction, so it is dominated. This condition is sufficient for extinction but not necessary. It describes the *sharp* side of the map, where eviction-like allocation wins, which we summarize by

$$D_2 = H^{-1}\sum_{h=1}^{H} \mathbb{I}[d_{h,2} \ge d_0], \tag{4}$$

the fraction of heads in a model–context cell whose 2-bit tier is dominated. The *diffuse* side, where uniform quantization wins, has a different cause: unequal allocation has little leverage when token importance varies less than the tier spacing. That side needs a dispersion coordinate. For attention-only importance, $\mathrm{std}_i(\log_2 a_i) = \tau/\ln 2$ with $\tau = \mathrm{std}_i(s_i)$ (the attention concentration); this is an algebraic identity, not evidence of optimality. D_2 contains no explicit L, but it is measured *at* a context and changes with it.

3 MEASUREMENT PROTOCOL

We measure at two levels. Attention-output distortion is exact and cheap to measure, so it supports a dense regime map over six models and 4k–256k tokens (§4). End-task accuracy is what users observe, but it is expensive, so we audit the map on three models drawn from both ends of it (§5).

Output-distortion level: models, cells, and units. We evaluate Llama-3.1-8B and Llama-3.3-70B (Grattafiori et al., 2024), Mistral-7B (Jiang et al., 2023), Qwen1.5-MoE-A2.7B (Qwen Team, 2024), Qwen3-8B, and Qwen3-30B-A3B (Yang et al., 2025). The main grid has 16 model–context cells between 8k and 128k; its extension has 25 cells spanning 4k–256k (44,416 query-head–configuration aggregates). The six architectures have 10,240 query-head and 2,016 KV-head positions, reused across contexts, so statistical claims treat cells and models, not heads, as independent units. PG-19 text (Rae et al., 2020) supplies the haystack, and every cell passes a full-cache retrieval check.

Compression, budget, and information contract. Keys are quantized with $\text{TurboQuant}_{\text{mse}}$ (Zandieh et al., 2025) (random rotation and Lloyd–Max scalar quantization (Lloyd, 1982)). The headline budget is $B = 3$ bits per key element; tier 0 evicts and tier 8 stores an 8-bit key. Values stay uncompressed. The *information contract* states what each method may see when it scores tokens. Our primary contract is symmetric: the mixed allocator and the eviction endpoint both use only lagged attention, and positions that score has not yet seen stay at the maximum tier. The endpoints, also called *corners*, are uniform 3-bit quantization and accumulated-attention eviction that keeps as many 8-bit keys as the budget allows (*uniform+accum*); a robustness analysis takes the minimum over five corners. Oracle results are upper bounds only.

Exact response and metrics. For every assignment we quantize the real keys, recompute attention, and measure the output error exactly. For head h,

$$g_h = \frac{\min(D_h^{\text{uniform}}, D_h^{\text{evict}})}{D_h^{\text{mixed}}}, \qquad G_+ = \exp\left[H^{-1} \textstyle\sum_h \log \max(g_h, 1)\right]. \tag{5}$$

G_+ is the primary cell response. It is a hindsight portfolio statistic, not the performance of a real router. The share of heads with $g_h \geq 2$ (the band) is a secondary summary with pre-registered thresholds: a cell is GO when its band is at least 35%, STOP below 15%, and NARROW in between.

End-task level. Llama-3.1-8B (8k, 32k, 128k) and Qwen3-8B (8k, 32k) share size and GQA ratio but sit at opposite ends of the map (low vs. high D_2). Four RULER-style tasks (Hsieh et al., 2024) (single-needle, multi-key, and multi-value retrieval, and variable tracking) use 20 held-out prompts per cell. The cache is compressed once, after the context is prefilled and *before* the question arrives, as in prefix caching; end-task allocations, like all baselines, are fixed after prefill. Compressing after the question lets observation-window scores read it, and SnapKV then scores 1.00. Budgets are $B \in \{2, 3\}$ key *code* bits per key element, checked at run time. Values, the last 32 context tokens, and generated tokens stay uncompressed in every method. Side information (norms, scales, outlier indices, width maps) counts as effective bits. We exclude blocks where the uncompressed model scores below 0.9 (only Qwen3-8B multi-value), leaving 36 valid model/context/task/budget cells, each run once (the *full grid*). The *paired rerun* (8k and 32k; 28 valid cells) runs all methods, including KIVI, KVQuant, and H2O, in one process per cell on the same prompts (App. D.4). Mistral-7B (8k, 32k; same GQA ratio, Llama's side of the map) is run the same way; its uncompressed model fails variable tracking at both contexts and multi-value retrieval at 32k, leaving 10 valid cells.

Baselines and the instrument. Quantization: TurboQuant-MSE, KIVI (Liu et al., 2024), and KVQuant (Hooper et al., 2024). Eviction, all keeping $\lfloor BL/8 \rfloor$ 8-bit keys: H2O (Zhang et al., 2023), SnapKV (Li et al., 2024), DropKV (Zhang et al., 2026a), Ada-KV (Feng et al., 2025), LaProx (Mai & Kim, 2026), and OBCache-K (Gu et al., 2026) with Ada-KV allocation. SIEVE applies Eq. 3 per KV head, scoring tokens with the 32 observation-window queries against 4-bit keys (the *4-bit cascade*, §4). A router assigns each KV head to the mixed interior, uniform quantization, or eviction, whichever has the lowest output error on 10 disjoint calibration prompts. The router is therefore *output-error-optimal by construction*; its end-task behavior tests the proxy, not a tuned method. Implementation details and deviations are in Appendix F.

4 OUTPUT-DISTORTION REGIMES

F1: Tier extinction orders the interior's output-error advantage, but the crossing is per architecture. On the 16-cell grid, the symmetric contract keeps roughly half of the advantage seen under an asymmetric one (current-query allocator vs. lagged eviction; Fig. 2a). The symmetric band stays 1.2–12.5 points above the all-oracle band in all 16 cells, five cells fall below 15% (STOP), and G_+ ranges from $1.05\times$ to $3.21\times$. D_2 still orders the symmetric band ($\rho = -0.985$). On the 25-cell extension, D_2 has Spearman $\rho = -0.978$ with both G_+ and the band, and -0.94 with G_+ (-0.915 with the band) after partialling out model identity (Fig. 1a). The crossing, however, is not universal. Pooled monotone crossings sit near $D_2 = 29.0\%$ (GO) and 54.1% (STOP) on the 25 cells, and at 26.8% and 57.4% on the 16-cell grid. Yet Llama-3.1-8B keeps a 20.2% band at $D_2 = 53.7\%$, while Qwen3-8B has 12.2% at $D_2 = 54.3\%$: an eight-point gap across 0.6 points of D_2, larger than prompt uncertainty. D_2 is thus an ordering coordinate, and any operational boundary must be fitted

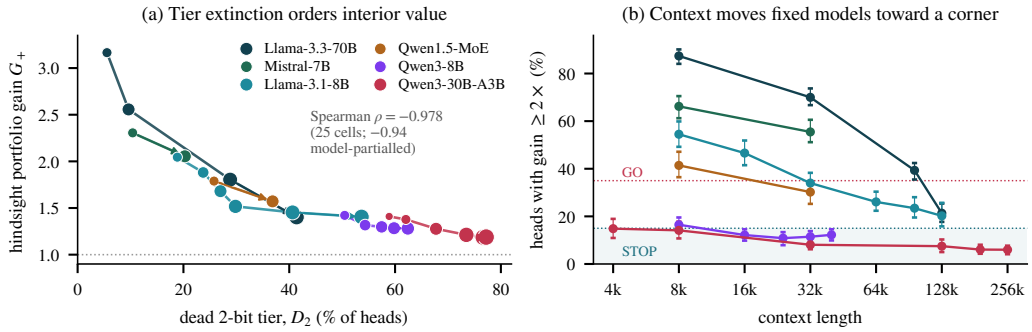


Figure 1: **F1–F2: where the interior lowers output error.** Mixed allocator and eviction endpoint both use lagged information; exact recomputation scores both. **(a)** The tier-extinction coordinate D_2 orders hindsight portfolio gain G_+ across 25 cells; arrows show increasing context. **(b)** Context moves each fixed model across predeclared output-error bands. Error bars are 90% layer-cluster intervals. Thresholds are not universal across architectures.

per architecture. The boundary also depends on the information contract. On the same 16 cells, STOP is crossed at $D_2 = 46.9\%$ when both sides see the current query and at 57.4% when both use lagged attention; this 10.5-point shift exceeds either confidence interval. Under the asymmetric contract STOP is never crossed. D_2 itself is not sensitive in this way: two campaigns with different corner sets and code agree on it to within 0.30 points in all 15 shared cells.

Why D_2 rather than a concentration or context statistic? We compared D_2 with simpler predictors on the same 25 cells, under a protocol fixed before computing them (Table 6): attention concentration τ, the spread of log token sensitivity (a heterogeneity statistic of the kind RateQuant uses (Zuo et al., 2026)), attention entropy, and context length alone. Within a model, every statistic that moves with context orders G_+ almost perfectly (mean $\rho = -0.99$); the difference lies across architectures. Pooled over models, D_2 reaches $\rho = -0.978$, against -0.72 for τ and heterogeneity and -0.48 for context alone. A log-linear fit on five models predicts the sixth to within a factor of 1.12 for D_2, against 1.15 (τ, heterogeneity) and 1.24 (context). D_2 is therefore most useful for comparing architectures, and its held-out advantage over dispersion statistics is modest.

F2: Context moves a fixed model toward the eviction corner. Every measured trajectory moves toward more tier extinction as context grows (Fig. 1b). Llama-3.3-70B goes from GO at 8k to NARROW at 128k; Qwen3-8B goes from NARROW at 8k to STOP by 16k. The main driver is absolute length: attention concentration τ rises smoothly and convexly in $\log L$, and a pre-registered curve of that shape explains most of the rise. Only at exactly the trained RoPE limit does τ exceed that curve, by about $+0.21$ in Qwen3-8B and Qwen3-30B; there is no excess in Llama-3.1-8B, or at 0.75 of the limit. Both context and architecture matter; the data do not support a universal RoPE-triggered transition. The surrogate is useful but not exact: the median ratio of predicted to recomputed gain lies in $[0.79, 1.34]$ (a fit check, not a held-out prediction).

F3: A head's policy family is stable; its token allocation goes stale within steps. A head's output-error-optimal family (interior, uniform, or eviction) stays the same over 4,096 generated tokens in all six measured cells, but the token allocation inside a mixed head goes stale within a few steps (Fig. 2b). A controller built on this map can therefore route rarely but must re-budget often. Table 5 (App. E) prices two constraints such a controller faces: sharing one allocation per GQA group costs $1.14\times$ at $n_{\text{rep}} = 4$, and the 4-bit cascade recovers 89% of the staleness penalty.

5 Do output-distortion regimes transfer to end tasks?

F4: Routed per head, the output-error optimum beats every eviction method, but a model-matched dense quantizer beats it. Table 2 and Fig. 3a show three regularities. First, routed per KV head, SIEVE beats every eviction method. Over the full grid it scores 0.766 against 0.602 for

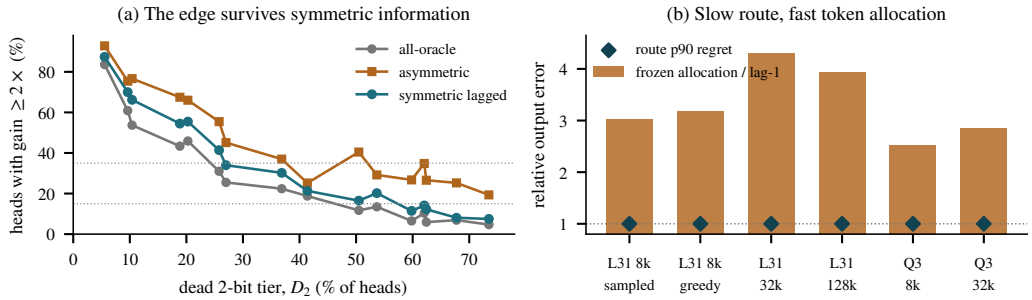


Figure 2: **F1 robustness and F3 timescales. (a)** On the same 16 cells, the lagged-versus-lagged advantage is smaller than the asymmetric numbers (current-query interior against lagged eviction) but remains ordered by D_2. **(b)** In six cells, keeping the calibration-time *policy family* has p90 regret 1.00 through 4,096 generated tokens, whereas freezing the token assignment costs 2.52–4.30× a one-step-old assignment.

Table 2: **End-task audit**: mean RULER score over valid cells (FP = 0.999). *Full grid*: 36 cells, one run each. *8k*: all methods paired in one process (§3). Output error and effective bits ($B = 2$) from the rerun. *Hard*: Llama-3.1-8B, 32k, 32-key, $B = 2$. Bold: best budget-matched; *not matched (1 side bit/element). The router sends 1–25% of KV heads to TurboQuant.

		full grid (36 cells)			8k–32k rerun (28 cells)				
family	method	Llama	Qwen	all	Llama	Qwen	out. err.	eff. bits	hard
quant.	TurboQuant-MSE	**0.985**	0.867	0.946	**0.995**	0.874	0.445	2.12	0.88
	KIVI (g=128)	0.96	0.98	0.97	0.965	0.986	0.298	2.25	0.85
	KIVI (g=32)*	0.99	0.99	0.99	0.991	0.998	0.203	3.00	0.88
	KVQuant	0.98	0.98	0.98	0.981	**0.987**	0.161	2.33	0.88
eviction	H2O	0.36	0.34	0.35	0.355	0.340	0.345	2.04	0.25
	SnapKV	0.479	0.237	0.398	0.385	0.222	0.290	2.04	0.21
	DropKV	0.459	0.334	0.418	0.345	0.327	0.310	2.04	0.18
	Ada-KV	0.612	0.285	0.503	0.550	0.259	0.289	2.04	0.25
	LaProx	0.640	0.457	0.579	0.583	0.453	0.294	2.04	0.32
	OBCache-K + Ada-KV	0.719	0.369	0.602	0.624	0.322	0.298	2.04	0.39
mixed	SIEVE interior only	0.479	0.438	0.465	0.458	0.463	0.150	2.07	0.28
	SIEVE (router)	0.654	**0.989**	0.766	0.692	0.962	**0.135**	2.08	0.32

OBCache-K + Ada-KV (+0.16, 90% CI [+0.13, +0.20] by prompt bootstrap). In the paired rerun it beats every eviction method on average, including H2O (0.808 vs. at most 0.527), with less than half their output error. Part of this gain comes from routing 1–25% of KV heads to dense (TurboQuant) quantization: the interior alone scores 0.465, below OBCache-K + Ada-KV, LaProx, and Ada-KV. On Qwen3-8B, routing 9–18% of heads to dense quantization lifts the score from 0.438 (interior) to 0.989, above TurboQuant alone (0.867); the paired rerun agrees (0.962 vs. 0.874). Second, on Llama-3.1-8B, keeping every token at low precision wins. TurboQuant scores 0.985, and at $B = 2$ every eviction or mixed method trails it by at least 0.35 (best: OBCache-K + Ada-KV, 0.626 vs. 0.976; SIEVE, 0.429). Third, on Qwen3-8B at $B = 2$, TurboQuant's per-token rotated keys break down on variable tracking (0.46 at 8k, 0.40 at 32k). This reversal belongs to TurboQuant, not to dense quantization. In the paired rerun the per-channel quantizers recover Qwen at both 8k and 32k: KIVI-128 scores 0.986 and KVQuant 0.987, above SIEVE (0.962), with only 0.17–0.25 more effective bits. KIVI-32 scores 0.998 but spends 3.00 effective bits against SIEVE's 2.08, so we do not treat it as budget-matched. Mistral-7B follows Llama (10 valid cells, 8k and 32k): TurboQuant scores 0.951, KVQuant 0.861, and KIVI-128 0.825, against 0.699 for SIEVE (interior alone: 0.070), which still leads every eviction method by a wide margin (best: 0.139). So on all three models a budget-matched dense quantizer beats every eviction and mixed policy, as a broad benchmark found for quantization versus token dropping (Yuan et al., 2024), but which one depends on the model. The pre-registered correlation between the output-error band and the router's accuracy gain is −0.70,

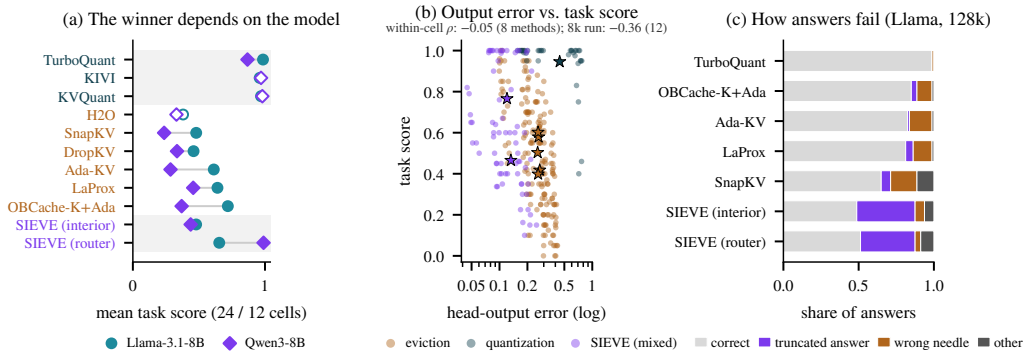


Figure 3: **F4–F6: what transfers to task accuracy. (a)** Mean task score per method and model over the full grid; TurboQuant wins on Llama-3.1-8B; on Qwen3-8B SIEVE beats TurboQuant but not the per-channel quantizers. **(b)** Head-output error vs. task score for every full-grid (cell, method); stars are method means. Error tracks accuracy across budgets and contexts; across methods it is weak, and TurboQuant (dark, top right) combines the highest error with the highest accuracy. **(c)** Single- and multi-key answers at Llama-3.1-8B, 128k, $B \in \{2, 3\}$: SIEVE fails by *truncating* the correct number, eviction methods by returning a wrong needle.

opposite in sign to the prediction. All 10 full-grid cells where SIEVE trails the best eviction method by more than the run-to-run spread (0.15) are on Llama, 7 of them at 128k (Table 8).

F5: Output distortion ranks budgets and contexts well and methods weakly. SIEVE minimizes the distortion it targets and has the lowest head-output error (0.120 on the full grid, 0.135 in the paired rerun), yet dense quantizers beat it. Fig. 3b separates two uses of the proxy. Across budgets and contexts, error and accuracy move together ($\rho = -0.69$ within the eviction family on the full grid). Across methods at a fixed budget, the relation is weak and depends on which methods are compared. The mean within-cell rank correlation is -0.05 over the eight full-grid methods and -0.29 over the twelve methods of the paired rerun (-0.39 among the eight eviction and mixed policies, -0.10 among the four dense quantizers); on Mistral-7B it is -0.40 (twelve methods). TurboQuant is the clearest exception: its rotated, spread-out noise gives the highest output error (0.44) yet the highest Llama accuracy, whereas per-channel quantizers have both low error and high accuracy. Output distortion therefore helps size budgets, but it overstates the cost of rotated quantization noise and should not be used to choose between families. The router's calibration costs some accuracy but is not the main cause: routing each head on its *measured* test-prompt error raises the score only from 0.808 to 0.886 (Llama: 0.692 to 0.815). The allocator is optimal for its own surrogate objective; what fails is the transfer from that objective to retrieval.

F6: Token-granular allocation truncates multi-token answers; positional smoothing largely repairs it. SIEVE's weakest setting is Llama-3.1-8B at 128k, where it trails OBCache-K + Ada-KV by 0.32 at $B = 2$ (0.442 vs. 0.762). Its wrong answers have a distinctive form (Fig. 3c). Asked for 4929054, it answers "492"; for 9465170, "946"; for 6022964, "602296". Llama tokenizes digits in groups of three, so the allocator keeps the key and the first token of the value but loses or garbles the rest. We call an answer *truncated* when it keeps the correct first three digits but is not the correct number. Truncation accounts for 36% of SIEVE's single- and multi-key answers at 128k (39% for the interior alone; proper prefixes alone, 23% and 28%), against 1–11% for every eviction method; those instead fail by returning a different needle (10–18%). Truncation is not specific to 128k: pooled over Llama's three contexts, 40% of the interior's answers are truncated, although eviction methods also truncate 10–26% at 8k and 32k, so the gap is widest at 128k. The router does not rescue these heads, because by its objective the interior is the right choice. At 128k it sends 99% of Llama's KV heads to the interior, whose output error (0.094) is half that of any eviction method. Losing the second token of one needle's value barely changes an error averaged over heads and decoding steps, but it changes the answer completely.

Why do the eviction methods avoid truncation? Every one of them that beats SIEVE here smooths its token scores over neighboring positions before selecting (max- or average-pooling over 7 positions). A kept token thus protects its neighbors, and a needle survives as a phrase. Pooling has its own cost:

Table 3: **F7:** how much each choice moves a verdict, compared with prompt sampling.

Choice varied	Effect	Verdicts changed
Prompt block (sampling)	band s.d. 0.4–3.0 pts, 4 cells	none
Eviction corner: accum vs. best of 5	band -5.1 to -8.6 pts	3 of 4 borderline cells
Information: current query vs. lagged	STOP crossing $46.9\% \rightarrow 57.4\%$ D_2	boundary location
Question visible at compression	SnapKV 0.11–0.65 $\rightarrow$ 1.00 in all 5 cells (same prompts); H2O changes by ≤ 0.08 (8k, 32k)	every eviction ranking
Dense baseline: TurboQuant vs. per-channel	Qwen3-8B 8k–32k: 0.87 $\rightarrow$ 0.99 (KIVI-128, KVQuant)	SIEVE vs. dense on Qwen

with several targets it can merge evidence that belongs to different ones (Zhang et al., 2026c). SIEVE, in contrast, allocates each token independently from unpooled attention. We pre-registered a test of this explanation: the same water-filling allocator on pooled attention supports it if it recovers at least half of the 128k, $B = 2$ gap to OBCache-K + Ada-KV, and refutes it if it recovers less than one fifth. The paired rerun already points in that direction. On Llama-3.1-8B, pooling cuts truncated single- and multi-key answers from 39% to 9% at 8k and from 44% to 11% at 32k. It raises the interior's mean score from 0.458 to 0.756, above OBCache-K + Ada-KV (0.624) and the calibrated router (0.692); at $B = 2$ it moves the interior past OBCache at both contexts (8k: 0.299 to 0.663 vs. 0.446; 32k: 0.343 to 0.627 vs. 0.570). Pooling helps less on Qwen3-8B (0.463 to 0.581; router 0.962). On Mistral-7B the interior truncates as often as on Llama (39%), but pooling cuts truncation only to 24% (score 0.070 to 0.172); there the router, which sends 16% of KV heads to dense quantization, does far better (0.699). The pre-registered 128k cell remains the decisive test.

F7: The comparison set and information contract move verdicts more than prompt sampling does. At both levels, the largest changes in a verdict came from what a method was compared against, and under which information contract, not from which prompts were drawn (Table 3). On the regime map, a new block of six prompts moves the band by a standard deviation of only 0.4–3.0 points, so six prompts per cell suffice there. Each of the other four choices in Table 3 moves a result by more than that and changes at least one verdict: the eviction corner, the information each side sees (F1), whether the question is visible when the cache is compressed, and which dense quantizer represents the endpoint (F4).

6 IMPLICATIONS

Five practical conclusions follow, each bounded by the retrieval tasks and models measured so far.

1. **Report dense low-bit quantization, with more than one quantizer, next to any eviction or mixed method.** On all three end-task models a dense quantizer beats every eviction and mixed policy we test at 2–3 key code bits, but not always the same one: TurboQuant on Llama-3.1-8B and Mistral-7B, per-channel KIVI and KVQuant on Qwen3-8B, where TurboQuant loses to SIEVE.

2. **Use output distortion to size budgets, not to choose methods.** It orders settings well (F1, F5) and methods weakly, and overstates the cost of rotated quantization noise (F5). Routing between method families needs a task-aware signal.

3. **Allocate spans, not tokens.** Token-separable objectives cannot see that a multi-token answer is only useful whole (F6). Smoothing the allocator's scores over neighboring positions, which leaves Eq. 3 intact, already cuts truncation from about 40% to 10% on Llama at 8k and 32k.

4. **Name the comparison set and information contract with every verdict.** They move results more than the prompt sample does (F7).

5. **Calibrate per architecture and per context, and re-budget often.** Crossings are architecture-specific and move with context (F1, F2); allocations go stale within steps (F3).

7 Related work

Dense KV quantization. Dense quantizers store every token at low precision: per-channel or per-token integer schemes (Liu et al., 2024; Hooper et al., 2024), low-rank and sparse residual correction (Kang et al., 2024), rotations and sketches (Zandieh et al., 2024; 2025), and importance-aware mixed precision (Yang et al., 2024; He et al., 2024). RateQuant allocates finite precision across K/V heads from fitted distortion curves (Zuo et al., 2026). Block-scaled 4-bit formats now make such caches native to hardware: the MX formats (Open Compute Project, 2023) and NVFP4, whose 16-element blocks carry FP8 scales (Alvarez et al., 2025).

Eviction and sparse attention. Evictors keep recent and sink tokens (Xiao et al., 2024), accumulate attention (Zhang et al., 2023; Liu et al., 2023), use the last query (Oren et al., 2024) or a prefill observation window (Li et al., 2024), and split budgets across layers and heads (Cai et al., 2024; Feng et al., 2025). Output-aware scores estimate the attention-output perturbation of removing a token: exactly for a singleton (Goel et al., 2025), under perturbation constraints (Feng et al., 2026), by decoupled residual-output perturbation (Zhang et al., 2026a), by Optimal-Brain-Damage saliency (Gu et al., 2026), through the value and output projection with a model-wide budget (Mai & Kim, 2026), or by layer-wise reconstruction with spatial-temporal smoothing (An et al., 2026). Query-aware sparsity keeps the whole cache and selects pages per query (Tang et al., 2024), and reasoning-oriented methods target long generated traces (Cai et al., 2025; Mao et al., 2026). Our truncation finding (F6) suggests one reason the strongest ones work: most smooth their scores over neighboring positions. We claim not output-aware eviction but the separable approximation that puts zero rate inside a finite-rate allocator.

Hybrid storage and rate allocation. DiffKV routes tokens among high precision, low precision, and pruning with an on-GPU manager (Zhang et al., 2025b), and HqeKV optimizes among full precision, several quantized levels, and eviction using joint K–V importance (Wang et al., 2026). Concurrent work RDKV derives joint eviction–quantization rate allocation, assigning zero-to-finite rates to value tokens and key channels with packed decode (Zhang et al., 2026b), and concurrent attention-aware transform coding applies output-aware reverse water-filling to transformed K/V channels (Laus et al., 2026). Our allocation unit is the within-head key token, and our question is not whether water-filling can be derived but when its interior pays and whether the answer survives the step to task accuracy. Head, channel, and token axes are potentially composable (Table 11, App. G).

Diagnostics, evaluation contracts, and systems. A broad benchmark already found that KV quantization is reliable across task types while token dropping excels only on some, that compressing during prefill hurts retrieval, and that a model can emit a passkey's first digits before compression takes effect (Yuan et al., 2024). Zhang et al. (2026c) hold an eviction selector's contract fixed (attention tensor, observation window, budget, projection rule) and vary one stage at a time. Their study covers eviction alone at 5–10% retention and finds that output-aware scoring helps in most cells where it has room to. Ours adds dense and mixed-precision policies and finds that output error fails at a different point: choosing *between* families at a fixed budget (F5). Our truncation failure (F6) is an instance of their projection stage, produced by a mixed-precision allocator rather than a selector, and F7 carries their fixed-contract principle one level up, to the baselines and information contract under which a verdict is stated.

8 Limitations and conclusion

Compression is key-only with exact values, and budgets match key code bits rather than total KV bytes. The regime crossing is architecture-specific, and the diffuse (uniform) side of the map has less support than tier extinction. Within that scope, mixed allocation is not uniformly useful. Tier extinction orders when the interior lowers output error, context moves a model toward a corner, and allocations go stale faster than routes. On end tasks, the output-error optimum, routed per head, beats every evictor we test, partly by keeping some heads dense. As prior benchmarks found, a model-matched dense quantizer does better still, and output distortion cannot identify it. Truncation points to the unit of allocation, which positional smoothing begins to repair. Verdicts depend more on the comparison contract than on sampling.

AI USE STATEMENT

In this work, we used generative AI tools (an AI coding and writing assistant) for the following tasks that the ICLR policy requires us to disclose. *Method implementation:* implementing and testing the KIVI and KVQuant-style baseline quantizers, the end-task result readers, the predictor comparison, and the figure and table scripts. *Experiment-design feedback:* proposing parts of the end-task campaign, namely the harder 32-key operating point, the question-visible control, the third end-task model, the pooled-score and oracle diagnostic arms, the predictor-comparison protocol, and several of their pre-registered decision rules, all of which the authors reviewed before any run. *Hypothesis proposal and results interpretation:* proposing the positional-pooling explanation of answer truncation (F6), the cross-cutting observation on comparison contracts (F7), and interpretations of the end-task results. *Qualitative data analysis:* classifying the models' wrong answers as truncated, wrong-needle, or other by rule, from generated text. We also used AI assistance, as the policy recommends disclosing, for editing code and job scripts, drafting and restructuring the paper (including its title and section structure), literature search and summarization, bibliography entries, and figure creation. We have not used generative AI tools to generate, alter, or impute measured data: every reported measurement is computed by the experiment code from model runs. AI tools were not used in earlier phases of the project. The key problem this paper studies, the theoretical framework in Section 2, and the regime-map experiments were not developed by AI. We have reviewed all AI-assisted work. AI-written code was checked with unit tests, including a check of the RoPE inversion against a real model's cache, and with end-to-end runs before any GPU campaign; reported statistics were recomputed from the raw result files by the same readers that generate the tables; literature summaries were checked against the cited papers, and bibliography entries against arXiv or publisher pages. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work measures compression of the key-value cache of publicly released language models on synthetic retrieval tasks built from public-domain books (PG-19). It involves no human subjects, no personal or sensitive data, and no new model training. The experiments used GPU time on a shared academic cluster (Appendix D.4). To guard research integrity, each experiment's decision rules were written before its results were read, and negative outcomes are reported alongside positive ones.

REPRODUCIBILITY STATEMENT

Section 2 and Appendices B–C give the allocation problem, its approximations, and the discrete optimizer. Section 3 and Appendix D specify models, context lengths, budgets, information contracts, and the response statistics; Appendix F specifies the end-task protocol, prompt blocks, validity filter, budget and side-information accounting, and every baseline with its deviations from the original paper. The anonymized supplementary code contains the measurement and end-task runners, the baseline implementations with their unit tests, the pre-registered plans and decision rules for each experiment, the readers that produce every table and figure from result files, and the job scripts with the exact arguments used. All models are public checkpoints, the corpus is public, and prompt indices and random seeds are fixed and recorded with each result. Appendix D.4 lists the hardware, software versions, and per-job resources.

REFERENCES

Eduardo Alvarez, Omri Almog, Eric Chung, Simon Layton, Dusan Stosic, Ronny Krashinsky, and Kyle Aubrey. Introducing NVFP4 for efficient and accurate low-precision inference. NVIDIA Technical Blog, 2025.

Yongqi An, Chang Lu, Kuan Zhu, Tao Yu, Chaoyang Zhao, Hong Wu, Ming Tang, and Jinqiao Wang. ReST-KV: Robust KV cache eviction with layer-wise output reconstruction and spatial-temporal smoothing. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/8be9c134bb193d8bd3827d4df8488228-Abstract-Conference.html.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2025. arXiv:2412.15204.

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

Zefan Cai, Wen Xiao, Hanshi Sun, Cheng Luo, Yikai Zhang, et al. R-KV: Redundancy-aware KV cache compression for reasoning models. In *Advances in Neural Information Processing Systems*, 2025. arXiv:2505.24133.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

William H. R. Equitz and Thomas M. Cover. Successive refinement of information. *IEEE Transactions on Information Theory*, 37(2):269–275, 1991.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S. Kevin Zhou. Ada-KV: Optimizing KV cache eviction by adaptive budget allocation for efficient LLM inference. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3773.

Yuan Feng, Junlin Lv, Haoyu Guo, Yukun Cao, S. Kevin Zhou, and Xike Xie. CriticalKV: Optimizing KV cache eviction from an output perturbation perspective. In *International Conference on Machine Learning*, 2026. arXiv:2502.03805.

Raghavv Goel, Junyoung Park, Mukul Gagrani, Dalton Jones, Matthew Morse, Harper Langston, Mingu Lee, and Chris Lott. CAOTE: KV cache selection for LLMs via attention output error-based token eviction. *arXiv preprint arXiv:2504.14051*, 2025. URL https://arxiv.org/abs/2504.14051.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yuzhe Gu, Xiyu Liang, Jiaojiao Zhao, and Enmao Diao. OBCache: Optimal brain KV cache pruning for efficient long-context LLM inference. In *International Conference on Machine Learning*, 2026. arXiv:2510.07651.

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

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

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

Albert Q. Jiang et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Hao Kang, Qingru Zhang, Souvik Kundu, Geonhwa Jeong, Zaoxing Liu, Tushar Krishna, and Tuo Zhao. GEAR: An efficient KV cache compression recipe for near-lossless generative inference of LLM. *arXiv preprint arXiv:2403.05527*, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Hannah Laus, Claudio Mayrink Verdun, Hao Wang, Flavio du Pin Calmon, and Felix Krahmer. KV cache compression through the lens of transform coding. *arXiv preprint arXiv:2608.14191*, 2026. URL https://arxiv.org/abs/2608.14191.

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.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time. In *Advances in Neural Information Processing Systems*, 2023.

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*, 2024.

Stuart Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2): 129–137, 1982.

Tho Mai and Joo-Young Kim. Reformulating KV cache eviction problem for long-context LLM inference. *arXiv preprint arXiv:2605.07234*, 2026.

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

NVIDIA. NVIDIA TensorRT model optimizer. https://github.com/NVIDIA/TensorRT-Model-Optimizer, 2024a.

NVIDIA. NVIDIA TransformerEngine. https://github.com/NVIDIA/TransformerEngine, 2024b.

NVIDIA. 3 ways NVFP4 accelerates AI training and inference. NVIDIA Technical Blog, 2025.

Open Compute Project. OCP microscaling formats (MX) specification. Specification v1.0, 2023.

Matanel Oren, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz. Transformers are multi-state RNNs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2024. arXiv:2401.06104.

Qwen Team. Qwen1.5-MoE: Matching 7b model performance with 1/3 activated parameters, 2024. URL https://qwenlm.github.io/blog/qwen-moe/.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*, 2020.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and accurate attention with asynchrony and low-precision. *arXiv preprint arXiv:2407.08608*, 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. arXiv:2406.10774.

He Wang, Yu Gu, Fangfang Li, Zhigang Wang, Zhenghao Liu, Ning Wang, Xiaohua Li, and Ge Yu. HqeKV: Towards hybrid quantization and eviction for KV cache in long-context LLM inference. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 4138–4153. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.201. URL https://aclanthology.org/2026.findings-acl.201/.

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

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

June Yong Yang, Byeongwook Kim, Jeongin Bae, Beomseok Kwon, Gunho Park, Eunho Yang, Se Jung Kwon, and Dongsoo Lee. No token left behind: Reliable KV cache compression via importance-aware mixed precision quantization. *arXiv preprint arXiv:2402.18096*, 2024.

Jiayi Yuan, Hongyi Liu, Shaochen Zhong, Yu-Neng Chuang, Songchen Li, Guanchu Wang, Duy Le, Hongye Jin, Vipin Chaudhary, Zhaozhuo Xu, Zirui Liu, and Xia Hu. KV cache compression, but what must we give in return? A comprehensive benchmark of long context capable approaches. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024. arXiv:2407.01527.

Amir Zandieh, Majid Daliri, and Insu Han. QJL: 1-bit quantized JL transform for KV cache quantization with zero overhead. *arXiv preprint arXiv:2406.03482*, 2024.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. TurboQuant: Online vector quantization with near-optimal distortion rate. *arXiv preprint arXiv:2504.19874*, 2025.

Aozhong Zhang, Selcuk Gurses, Yanxia Deng, Naigang Wang, Charlie Liu, Davis Wertheimer, Derrick Liu, Xin Li, Zi Yang, Felix Ye, and Penghang Yin. DropKV: Decoupling residual-output perturbation for near-optimal KV-cache eviction. In *ICML Workshop*, 2026a. Code: github.com/AozhongZhang/DropKV.

Jintao Zhang, Haofeng Huang, Pengle Zhang, Jia Wei, Jun Zhu, and Jianfei Chen. SageAttention2: Efficient attention with thorough outlier smoothing and per-thread INT4 quantization. *arXiv preprint arXiv:2411.10958*, 2024a.

Jintao Zhang, Jia Wei, Haofeng Huang, Pengle Zhang, Jun Zhu, and Jianfei Chen. SageAttention: Accurate 8-bit attention for plug-and-play inference acceleration. *arXiv preprint arXiv:2410.02367*, 2024b.

Jintao Zhang, Jia Wei, Pengle Zhang, Xiaoming Xu, Haofeng Huang, Haoxu Wang, Kai Jiang, Jianfei Chen, and Jun Zhu. SageAttention3: Microscaling FP4 attention for inference and an exploration of 8-bit training. *arXiv preprint arXiv:2505.11594*, 2025a.

Junkai Zhang, Hang Guo, Luca Benini, and Yawei Li. RDKV: Rate-distortion bit allocation for joint eviction and quantization of the KV cache. *arXiv preprint arXiv:2605.08317*, 2026b. URL https://arxiv.org/abs/2605.08317.

Ruijie Zhang, Haozhe Liang, Da Chang, Li Hu, Fanqi Kong, Huaxiao Yin, and Yu Li. When does value-aware KV eviction help? A fixed-contract diagnostic for non-monotone cache compression. *arXiv preprint arXiv:2605.08234*, 2026c.

Yanqi Zhang, Yuwei Hu, Runyuan Zhao, John C. S. Lui, and Haibo Chen. DiffKV: Differentiated memory management for large language models with parallel KV compaction. In *Proceedings of the ACM SIGOPS 31st Symposium on Operating Systems Principles*, pp. 431–445. ACM, 2025b. doi: 10.1145/3731569.3764810.

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. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, 2023.

Fei Zuo, Zikang Zhou, Hao Cong, Xiaoyan Xi, and Ho Fai Leung. RateQuant: Optimal mixed-precision KV cache quantization via rate-distortion theory. *arXiv preprint arXiv:2605.06675*, 2026. URL https://arxiv.org/abs/2605.06675.

A SCOPE OF THE COMMON RATE-ALLOCATION VIEW

This appendix expands the derivation and measurement contract used in the main paper. The common action space is deliberately narrow: it allocates *key-token* rates within an attention head, leaves values exact, and measures attention-output error. Prior work already combines quantization and eviction in deployable systems. Our claim is instead that a common local surrogate makes it possible to ask a different empirical question: when does the discrete interior improve over uniform quantization and eviction under matched information and key-bit budgets?

Throughout, *zero rate* means a zero-bit action in the separable planning surrogate. It does not mean that set eviction is exactly identical to independent additive logit noise. Section B makes this distinction explicit.

B OUTPUT DISTORTION AND THE ZERO-RATE APPROXIMATION

B.1 ATTENTION AND FINITE-BIT KEY PERTURBATIONS

For a query $q \in \mathbb{R}^d$, keys $k_i \in \mathbb{R}^d$, and values $v_i \in \mathbb{R}^{d_v}$, define

$$s_i = q^\top k_i/\sqrt{d}, \qquad a_i = \frac{\exp(s_i)}{\sum_j \exp(s_j)}, \qquad o = \sum_i a_i v_i. \tag{6}$$

Quantizing key i changes its logit by δ_i. Because

$$\frac{\partial o}{\partial s_i} = a_i(v_i - o),$$

the first-order output perturbation is

$$\Delta o \approx \sum_i a_i(v_i - o)\delta_i. \tag{7}$$

Consequently,

$$\begin{aligned}\mathbb{E}\|\Delta o\|^2 \approx \sum_i a_i^2 \|v_i - o\|^2 \operatorname{Var}(\delta_i) \\ + \sum_{i\neq j} a_i a_j (v_i - o)^\top (v_j - o) \operatorname{Cov}(\delta_i, \delta_j).\end{aligned} \tag{8}$$

Equation equation 1 follows only when the residual perturbations are zero mean and the cross-token covariance contribution vanishes. Softmax invariance removes a *common* shift, $\mathrm{softmax}(s + c\mathbf{1}) = \mathrm{softmax}(s)$; it does not remove arbitrary token-dependent bias.

B.2 EXACT SET EVICTION

Let E be an evicted token set and $A_E = \sum_{i\in E} a_i$. Renormalizing the remaining attention weights yields

$$o_{\setminus E} = \frac{o - \sum_{i\in E} a_i v_i}{1 - A_E}, \tag{9}$$

$$o_{\setminus E} - o = -\frac{\sum_{i\in E} a_i (v_i - o)}{1 - A_E}, \tag{10}$$

$$\|o_{\setminus E} - o\|^2 = \frac{\left\|\sum_{i\in E} a_i (v_i - o)\right\|^2}{(1 - A_E)^2}. \tag{11}$$

The denominator and cross-token inner products prevent an exact token-separable representation. For one evicted token j, the exact error magnitude is

$$\frac{a_j}{1 - a_j}\|v_j - o\|,$$

which is the CAOTE singleton score (Goel et al., 2025). ReST-KV further models attention redistribution and output reconstruction (An et al., 2026); our local planner does not capture those effects.

B.3 The planning approximation

When the evicted mass is small and cross terms are neglected, Eq. equation 11 motivates

$$\widetilde{D}_0(E) = \sum_{i \in E} w_i, \qquad w_i = a_i^2 \|v_i - o\|^2.$$

We choose units in which the absolute zero-rate distortion is $d_0 = 1$ and write the finite-rate costs as $d_b = \mathrm{Var}(\delta_b)$ in the same logit units. Thus

$$\widetilde{D}(\mathbf{b}) = \sum_i w_i d_{b_i}, \qquad b_i \in \mathcal{B}, \quad 0 \in \mathcal{B}.$$

This convention keeps relative quantizer factors separate from the absolute zero-rate cost. It is a local approximation used to choose an allocation; every reported response is obtained by recomputing attention with the actual quantized/evicted cache.

C Discrete rate allocation and regime coordinates

C.1 The implemented optimizer

At average key budget B, the discrete problem is

$$\min_{b_i \in \mathcal{B}} \sum_i w_i d_{b_i} \quad \text{subject to} \quad \sum_i b_i \le BL. \tag{12}$$

Introducing multiplier $\lambda \ge 0$ separates the token decisions:

$$b_i^\star(\lambda) = \arg\min_{b \in \mathcal{B}} \{w_i d_b + \lambda b\}. \tag{13}$$

We search λ to satisfy the budget and use the discrete tiers in Eq. equation 13; the continuous expression is explanatory, not the experimental optimizer.

For the relaxation $d_b = \kappa 4^{-b}$,

$$b_i^\star = \frac{1}{2} \log_2 w_i + c = \log_2 a_i + \log_2 \|v_i - o\| + c.$$

Adding the value term $\log_2 \|v_i - o\|$ changed an *aggregate ladder-width statistic* by at most 0.035 bits in our measurements. This is not a per-token bound; the end-task allocator therefore uses the full weight $w_i = a_i^2 \|v_i - o\|^2$.

C.2 Tier extinction

If $d_b \ge d_0$, tier b consumes positive bits without improving the local surrogate relative to zero rate and is dominated. This is a sufficient extinction condition. The condition is not necessary: a tier with $d_b < d_0$ can still be dominated if it lies above the lower convex envelope of the other rate–distortion points. For the 2-bit tier we report

$$D_2 = \frac{1}{H} \sum_{h=1}^{H} \mathbb{I}[d_{h,2} \ge d_0].$$

D_2 is measured at a particular model and context. Although L is absent from its formula, the head cost distribution changes with L, so D_2 is not sequence-length invariant and a new deployment context requires calibration.

C.3 WHY THE REGIME MAP NEEDS TWO COORDINATES

The sharp side and diffuse side arise for different reasons:

1. On the sharp side, large absolute finite-bit noise removes low-bit tiers; D_2 directly probes this boundary.
2. On the diffuse side, unequal allocation has little leverage when token importances are nearly equal; an importance-dispersion statistic is needed.

For attention-only importance, $\mathrm{std}_i(\log_2 a_i) = \tau/\ln 2$ with $\tau = \mathrm{std}_i(s_i)$, because softmax subtracts the same log normalizer from every logit. This identity alone neither proves that τ is a complete regime variable nor establishes a uniform-quantization optimum. The paper therefore treats the mixed-to-eviction side as the well-supported boundary; Table 6 compares D_2 with dispersion predictors.

D MEASUREMENT PROTOCOL AND ACCOUNTING

D.1 GRID AND INDEPENDENT UNITS

The main (symmetric) grid contains 16 model–context cells from six models and 29,312 query-head–configuration aggregates. The boundary extension contains 25 cells spanning 4k–256k and 44,416 such aggregates. Across the six models, 10,240 query-head positions and 2,016 KV-head positions are counted once; contexts reuse those positions. Model/cell-level analyses, rather than raw heads, are therefore the independent statistical units.

PG-19 passages supply natural context. Each cell is admitted only after the uncompressed model passes the needle-retrieval validity gate. The allocation planner uses its local surrogate, while every reported error comes from actual key quantization or eviction followed by exact attention-output recomputation.

D.2 MATCHED INFORMATION AND BUDGET

The primary comparison uses $B = 3$ bits per key element. Uniform quantization stores every key at 3 bits. The eviction endpoint retains the number of 8-bit keys allowed by the same key-bit budget. In the symmetric contract, the mixed allocator and accumulated-attention eviction both use lagged importance information; an oracle/current-query result is labeled only as an upper bound. The five-corner robustness result changes the reference set and is reported separately.

This accounting omits value-cache bytes, quantization scales, indices, metadata, protected windows, alignment, and packing. Moreover, *8-bit* is the maximum experimental key tier, not full floating-point precision. The experiments therefore establish matched key-bit output error, not matched total memory or throughput.

D.3 RESPONSES

For each head,

$$g_h = \frac{\min(D_h^{\mathrm{uniform}}, D_h^{\mathrm{evict}})}{D_h^{\mathrm{mixed}}}, \qquad G_+ = \exp\left(\frac{1}{H}\sum_h \log\max(g_h, 1)\right).$$

G_+ is a hindsight portfolio statistic: it clips losses after observing all three errors. It is not the performance of a deployed router. The primary cell-level response is continuous G_+; the fraction with $g_h \geq 2$ is a secondary thresholded band statistic. Every band result names its corner and information contracts.

D.4 COMPUTE RESOURCES AND SOFTWARE

All experiments ran as Slurm jobs on one shared academic GPU cluster. Every job recorded its GPU and library versions, and the cluster has NVIDIA H100 80GB HBM3 GPUs, PyTorch 2.13.0, and

Table 4: Hardware per experiment. Each job ran on one node; wall times are measured from job records where available.

Experiment	GPUs per job	Wall time per job
Regime map, 7–8B and MoE models (§4)	1×H100	minutes to a few hours per cell
Regime map, Llama-3.3-70B and Qwen3-30B-A3B	4×H100 (70B), up to 4×H100 (30B)	up to several hours per cell
End-task audit, one model–context cell of 20 prompts (§5)	1×H100	0.5 h (8k) to 2.7 h (128k)
Router calibration, 10 prompts	1×H100	about 10 minutes
Nested-code overhead (App. F)	1×H100	1.8–2.8 h per cell
Readers, statistics, figures, tests	CPU only	minutes

Hugging Face Transformers 5.16.1 (Python 3.11). Models were loaded in bfloat16 from pinned local checkpoints with the hub offline.

Because the model forward pass is not bit-reproducible across processes even on one GPU, every comparison that pairs methods prompt by prompt computes all methods inside one process on the same captured queries, keys, and values; comparisons across processes are made on cell means only. The spread is measurable: the full grid and the 8k run evaluate the same 8k cells on the same prompts in different processes, and their cell means differ by a median of 0.04 and a 90th percentile of 0.15 over 112 cell–method pairs (TurboQuant: at most 0.05). Per-cell wins and losses (Table 8) therefore use a 0.15 margin. Method means over the 14 cells differ by at most 0.08 (DropKV; SIEVE 0.01, TurboQuant 0.001): enough to reorder close eviction methods, but not SIEVE against eviction or dense quantization.

E ADDITIONAL ROBUSTNESS AND DEPLOYMENT EVIDENCE

Table 5: Measured costs of acting on the output-distortion map. Output error only.

Constraint	Measurement	Implication
Family label over 4,096 tokens	p90 regret 1.00 in 6/6 cells	calibrate the route once per context
Frozen token assignment	2.52–4.30× lag-1 error	re-budget mixed heads every few steps
Shared GQA allocation	$1.000/1.136/1.316\times$ at $n_{\mathrm{rep}} = 1/4/8$	route per KV head at a small cost
4-bit current-query cascade	closes 82–91% (median 89%) of lag penalty	re-score from base-tier keys

The completed measurements support the following bounded conclusions:

- In all 16 main-grid cells, the symmetric lagged-versus-lagged band exceeds the all-oracle band by 1.2–12.5 points, but it is lower than the asymmetric current-query-versus-lagged numbers. Five cells are STOP and five NARROW.
- Across the 25 extension cells, D_2 has Spearman correlation -0.978 with both G_+ and the band; controlling for model, it is -0.94 with G_+ and -0.915 with the band. Similar D_2 values can nevertheless differ by eight band points across architectures.
- Prompt-block standard deviation is 0.4–3.0 band points. Replacing the two headline corners by the five-corner minimum lowers the band by 5.1–8.6 points and can change a categorical verdict.
- In six decode-stability cells, the calibration-time family choice has p90 regret 1.00 through 4,096 generated tokens, whereas a frozen token allocation costs 2.52–$4.30\times$ a one-step-old allocation.
- Enforcing one allocation per shared KV head gives marginal in-band costs 1.000, 1.136, and 1.316 at GQA repetition factors 1, 4, and 8. A 4-bit current-query cascade closes 82–91% (median 89%) of the lagged-score penalty.

Table 6: Predictor comparison on the 25 regime-map cells (§4). Columns: pooled Spearman correlation with G_+; the same partialled on model identity; leave-one-model-out error of a log-linear fit of G_+, as a multiplicative factor; mean Spearman within the held-out model. All predictors are cell medians of per-head medians on the rows that define D_2; the recomputed D_2 matches the reported one to within 0.52 points in every cell. The heterogeneity statistic is the spread of $\frac{1}{2}\log_2 w_i$; for log-normal w it ranks cells as the AM/GM ratio does.

predictor	ρ with G_+	$\rho \mid$ model	held-out model error	within-model ρ
D_2 (ours)	-0.978	-0.942	$1.119\times$	-0.986
spread of log sensitivity (heterogeneity)	-0.722	-0.923	$1.155\times$	-0.986
attention concentration τ	-0.724	-0.922	$1.154\times$	-0.986
attention entropy	-0.288	-0.903	$1.273\times$	-0.943
n_{95}/L	$+0.331$	$+0.021$	$1.271\times$	$+0.286$
context length only	-0.478	-0.872	$1.243\times$	-0.986

Methods note: fitting the crossing. On the 16-cell main grid, the band crosses 35% between the cells at $D_2 = 25.8\%$ and 27.0%, and the isotonic crossing is 26.8%. A logistic fit instead places GO at 34.6%, outside that bracket, because the flat tail above 30% pulls the curve. We therefore report isotonic (monotone) crossings, which cannot leave the bracket of measured cells, and use the logistic only as a flagged cross-check. On the 25-cell extension the GO crossing moves to 29.0%: Llama-3.3-70B at 96k lies above the GO line at $D_2 = 28.8\%$, so models now overlap near the line instead of leaving a gap.

Why the audit uses synthetic retrieval. An audit can only rank compression policies on a task where their answers differ. We screened a natural task, LongBench-v2 (Bai et al., 2025) in forced-choice form, on two development blocks of 52 questions. Even choosing the best policy separately for each question beat the best single fixed policy on only 3 and 2 of the 52 questions, so at these budgets the task barely separates policies, and its pre-registered gate stopped both blocks before the confirmation block. On the RULER tasks of Appendix F, method means differ by more than 0.6, so the audit can discriminate there.

F End-task audit: protocol, baselines, and full results

Protocol. Each prompt's context is prefilled at full precision and scored; the cache is then compressed once, and the question is prefilled *through* the compressed cache before greedy decoding. This is the question-agnostic setting of a prefix cache or multi-turn session. When the question is inside the observation window instead, SnapKV scores 1.00 at every budget, because the window vote points at the needle. Evaluation uses prompts 100–119; the SIEVE router is calibrated on the disjoint prompts 0–9. A block is valid when the uncompressed model scores at least 0.9; this excludes Qwen3-8B multi-value retrieval (0.24 at 8k, 0.04 at 32k). Scores are RULER string matches; multi-value and variable tracking give fractional credit.

Budget accounting. Every method spends B key code bits per context-key element, verified at run time from the applied bit maps. Side information differs by method and is reported in Table 7 rather than folded into B. Eviction and SIEVE keep tokens at 8 bits with the same TurboQuant quantizer, so their side information scales with the kept fraction.

Storage costs outside the code budget. Two further costs are small relative to the differences reported in the main text. *Nested tiers.* A mixed allocator must store tokens at several widths. We measured the rate cost of storing the tier ladder $\{0, 3, 4, 6, 8\}$ as one successively refinable 3+1+2+2 code, in which an 8-bit index truncates to 6, 4, or 3 bits without re-encoding, against separate Lloyd–Max codebooks, at matched attention-output error; successive refinability is classical (Equitz & Cover, 1991), but our scalar codes are not the asymptotic optimum, so we measured it. The code itself costs at most 0.42% rate at any width. End to end at $B = 3$, including the extra bit that 3-bit tokens keep for the 4-bit cascade score, the overhead is $+0.34\%$ averaged over four model–context cells (90% CI $[-0.33, +0.99]$) and $+2.9\%$ in the worst cell. *Protected window.* The last 32 context tokens stay uncompressed in every method: 0.39% of the context at 8k, 0.10% at 32k, and 0.02% at

128k. Because it is identical across methods, it shifts every method's memory equally and does not affect the comparison.

Table 7: Baselines as implemented, and side information per key element ($d = 128$).

method	what is stored, and how it is chosen	side bits
TurboQuant-MSE	every token at B bits; random rotation, Lloyd–Max, per-token norm	0.125
KIVI ($g = 32$ / 128)	every token at B bits; per-channel asymmetric integer, min/max per g tokens, post-RoPE	1.0 / 0.25
KVQuant	every token at B bits; pre-RoPE per-channel non-uniform codebook; 1% outliers and token 0 exact	≈ 0.32
H2O	$\lfloor BL/8 \rfloor$ tokens at 8 bits; raw attention summed over all prefill queries, no pooling; no separate recent-token budget beyond the 32-token window shared by all methods	≈ 0.05
SnapKV	same keep count; 32-query observation window, max-pool 7	≈ 0.05
DropKV, Ada-KV, OBCache-K, LaProx	same keep count; published scores; Ada-KV head allocation for Ada-KV and OBCache-K; LaProx global top-K	≈ 0.05
SIEVE	per-token width in $\{0, 1, 2, 3, 4, 5, 6, 8\}$; Eq. 3 per KV head, 4-bit cascade score; offline per-KV-head router	0.07–0.09

Deviations from the original papers. KIVI keeps its last partial group in full precision; we quantize it as a short group, and the 32-token uncompressed window shared by all methods plays the role of KIVI's residual. KVQuant calibrates outlier thresholds and its non-uniform codebook offline with Fisher-weighted k-means; we fit both online on each prompt's own context without weighting. Fitting on the test context itself likely favors KVQuant. The Qwen conclusion does not rest on it: KIVI-128, which needs no calibration, also beats SIEVE there on its own (0.987 vs. 0.953). H2O uses its raw accumulated score, whose causal position bias (a token is seen by every later query) is H2O's own.

Table 8: SIEVE (router) against each baseline over the 36 valid cells. A win or loss is a cell-mean difference above 0.15, the 90th percentile of the run-to-run spread (App. D.4); the interval resamples prompts within each model and context.

SIEVE (router) vs.	wins	ties	losses	mean Δ	90% CI
TurboQuant-MSE	3	17	16	-0.180	$[-0.208, -0.152]$
SnapKV	23	11	2	$+0.367$	$[+0.338, +0.398]$
DropKV	25	8	3	$+0.348$	$[+0.314, +0.384]$
Ada-KV	21	9	6	$+0.262$	$[+0.236, +0.290]$
LaProx	19	10	7	$+0.187$	$[+0.154, +0.221]$
OBCache-K + Ada-KV	17	9	10	$+0.163$	$[+0.129, +0.197]$
best eviction method per cell	16	10	10	$+0.127$	—

Table 9: Failure case: Llama-3.1-8B at 128k. Error types are over single- and multi-key answers at $B \in \{2, 3\}$ (80 per method). *Truncated*: the prediction keeps the correct first three digits (Llama's first number token) but is not the correct number; proper prefixes alone account for 28% (interior), 23% (router), and at most 9% (eviction methods). *Wrong needle*: another needle's value.

	task score		single/multi-key answers, B=2, 3			evicted	head-output	heads to
method	B=2	B=3	truncated	wrong needle	other	(B=2)	error (B=2)	interior
TurboQuant-MSE	0.942	0.982	0%	1%	0%	0%	0.706	—
SnapKV	0.523	0.652	6%	18%	11%	75%	0.187	—
DropKV	0.480	0.657	11%	16%	10%	75%	0.201	—
Ada-KV	0.704	0.872	1%	15%	1%	75%	0.181	—
LaProx	0.737	0.863	5%	12%	1%	75%	0.187	—
OBCache-K + Ada-KV	0.762	0.902	4%	10%	1%	75%	0.182	—
SIEVE interior only	0.409	0.622	39%	6%	6%	65%	0.096	—
SIEVE (router)	0.442	0.677	36%	4%	9%	64%	0.094	99%

Table 10: Full end-task grid: mean score over 20 prompts per cell. Bold marks the best method in each valid row. ‡: uncompressed score below 0.9, excluded from every aggregate. KIVI, KVQuant, and H2O were measured only in the 8k run (Table 2).

model	ctx	task	B	FP	TurboQ.	SnapKV	DropKV	Ada-KV	LaProx	OBC+Ada	SIEVE-int.	SIEVE
Llama-3.1-8B	8k	multi-key	2	1.00	**1.00**	0.20	0.30	0.25	0.25	0.40	0.35	0.30
Llama-3.1-8B	8k	multi-key	3	1.00	**1.00**	0.35	0.55	0.55	0.55	0.75	0.60	**1.00**
Llama-3.1-8B	8k	multi-value	2	1.00	**1.00**	0.14	0.16	0.19	0.25	0.39	0.15	0.23
Llama-3.1-8B	8k	multi-value	3	1.00	**1.00**	0.34	0.42	0.42	0.46	0.59	0.40	0.99
Llama-3.1-8B	8k	single	2	1.00	**1.00**	0.45	0.55	0.50	0.60	0.70	0.30	0.35
Llama-3.1-8B	8k	single	3	1.00	**1.00**	0.75	0.65	0.75	0.80	0.75	0.60	**1.00**
Llama-3.1-8B	8k	var. track.	2	1.00	**0.98**	0.45	0.49	0.42	0.40	0.54	0.43	0.60
Llama-3.1-8B	8k	var. track.	3	1.00	**1.00**	0.60	0.64	0.62	0.59	0.88	0.67	0.95
Llama-3.1-8B	32k	multi-key	2	1.00	**1.00**	0.30	0.20	0.35	0.45	0.55	0.40	0.45
Llama-3.1-8B	32k	multi-key	3	1.00	**1.00**	0.45	0.35	0.70	0.70	0.70	0.60	**1.00**
Llama-3.1-8B	32k	multi-value	2	1.00	**0.97**	0.12	0.06	0.20	0.26	0.35	0.16	0.23
Llama-3.1-8B	32k	multi-value	3	1.00	**1.00**	0.42	0.23	0.56	0.61	0.70	0.40	0.90
Llama-3.1-8B	32k	single	2	1.00	**1.00**	0.60	0.35	0.65	0.75	0.85	0.55	0.60
Llama-3.1-8B	32k	single	3	1.00	**1.00**	0.70	0.60	0.95	0.95	0.90	0.65	**1.00**
Llama-3.1-8B	32k	var. track.	2	1.00	**0.99**	0.28	0.37	0.46	0.47	0.69	0.46	0.63
Llama-3.1-8B	32k	var. track.	3	1.00	**1.00**	0.65	0.55	0.82	0.87	0.86	0.64	**1.00**
Llama-3.1-8B	128k	multi-key	2	1.00	**1.00**	0.30	0.40	0.50	0.60	0.65	0.35	0.40
Llama-3.1-8B	128k	multi-key	3	1.00	**0.95**	0.45	0.50	0.80	0.70	0.85	0.50	0.55
Llama-3.1-8B	128k	multi-value	2	0.97	**0.94**	0.41	0.35	0.57	0.59	0.64	0.34	0.34
Llama-3.1-8B	128k	multi-value	3	0.97	**0.99**	0.54	0.55	0.79	0.81	0.85	0.55	0.69
Llama-3.1-8B	128k	single	2	1.00	**1.00**	0.90	0.75	**1.00**	0.95	0.95	0.45	0.45
Llama-3.1-8B	128k	single	3	1.00	**1.00**	0.95	0.85	**1.00**	**1.00**	0.95	0.65	0.65
Llama-3.1-8B	128k	var. track.	2	1.00	**0.83**	0.48	0.42	0.74	0.81	0.81	0.50	0.58
Llama-3.1-8B	128k	var. track.	3	1.00	**0.99**	0.67	0.73	0.90	0.94	0.96	0.79	0.82
Qwen3-8B	8k	multi-key	2	1.00	**0.95**	0.00	0.05	0.00	0.05	0.05	0.15	0.90
Qwen3-8B	8k	multi-key	3	1.00	**1.00**	0.10	0.20	0.15	0.30	0.10	0.50	**1.00**
Qwen3-8B	8k	multi-value‡	2	0.24	0.69	0.00	0.05	0.00	0.01	0.01	0.05	0.72
Qwen3-8B	8k	multi-value‡	3	0.24	0.49	0.00	0.07	0.00	0.21	0.10	0.28	0.59
Qwen3-8B	8k	single	2	1.00	0.95	0.10	0.10	0.05	0.25	0.15	0.20	**1.00**
Qwen3-8B	8k	single	3	1.00	**1.00**	0.25	0.35	0.25	0.55	0.40	0.60	**1.00**
Qwen3-8B	8k	var. track.	2	1.00	0.46	0.19	0.28	0.28	0.41	0.43	0.47	**0.99**
Qwen3-8B	8k	var. track.	3	1.00	**1.00**	0.63	0.63	0.62	0.66	0.68	0.81	**1.00**
Qwen3-8B	32k	multi-key	2	1.00	0.75	0.00	0.15	0.05	0.15	0.15	0.10	**1.00**
Qwen3-8B	32k	multi-key	3	1.00	**1.00**	0.10	0.30	0.10	0.30	0.25	0.40	**1.00**
Qwen3-8B	32k	multi-value‡	2	0.04	0.75	0.00	0.00	0.00	0.05	0.00	0.01	0.12
Qwen3-8B	32k	multi-value‡	3	0.04	0.42	0.00	0.04	0.00	0.03	0.00	0.00	0.07
Qwen3-8B	32k	single	2	1.00	0.95	0.20	0.30	0.20	0.40	0.25	0.35	**1.00**
Qwen3-8B	32k	single	3	1.00	**1.00**	0.30	0.55	0.50	0.70	0.55	0.45	**1.00**
Qwen3-8B	32k	var. track.	2	1.00	0.40	0.28	0.40	0.45	0.79	0.61	0.46	**0.99**
Qwen3-8B	32k	var. track.	3	1.00	0.95	0.69	0.70	0.77	0.92	0.81	0.76	**0.99**

G EXTENDED RELATED WORK

Table 11: Closest work and the claim boundary of this paper.

Work	**Allocation/action**	**Difference here**
DiffKV; HqeKV	token high/low precision and pruning	explain and predict when the middle tier pays; no first-hybrid claim
CAOTE; CriticalKV; DropKV; OBCache; LaProx; ReST-KV	output-aware token eviction	finite-rate extension under a stated local surrogate; exact set error retained; evaluated as baselines where available
RateQuant	K/V heads, finite rates	within-head key tokens, explicit zero rate, context traversal
RDKV	value tokens, key channels, zero-to-finite rates	when the token-wise interior beats endpoints; score staleness and regime movement
AATC	transformed K/V channels, including zero	token-axis tier viability; values/channels remain outside our allocation
Fixed-contract diagnostic	attribution of eviction failures	measures how comparison set and information contract move verdicts; truncation mechanism

Low-precision formats and the dense endpoint. Block-scaled 4-bit formats strengthen the dense endpoint. NVFP4 pairs 4-bit E2M1 values with FP8 scales over 16-element blocks (Alvarez et al.,

2025; NVIDIA, 2025), finer than the power-of-two, 32-element scales of MXFP4 (Open Compute Project, 2023), and libraries provide fused FP8/FP4 kernels (NVIDIA, 2024b;a). A 4-bit cache with hardware support is therefore the baseline any mixed or evicting policy must beat.

Low-precision attention kernels. FlashAttention-2 and -3 fuse attention without materializing the score matrix, the latter with FP8 (Dao, 2023; Shah et al., 2024); SageAttention quantizes the attention computation itself to 8 bits, INT4 with outlier smoothing, and microscaled FP4 (Zhang et al., 2024b;a; 2025a). Because these kernels never write attention scores to memory, evictors that read historical scores (H2O's accumulated attention, and the lagged scores our allocator uses) must either fall back to eager attention or estimate importance another way, for example from pre-RoPE geometry (Mao et al., 2026). Paged allocators also reclaim only empty blocks (Kwon et al., 2023), so scattered surviving tokens free little memory unless the cache is compacted (Mao et al., 2026). Our 4-bit cascade avoids this: it re-scores tokens from stored 4-bit keys in a small separate pass over the 32 window queries, so it needs no scores from the fused kernel. Our experiments use an eager attention path and therefore measure accuracy, not these costs.