WHAT TRANSFERS ACROSS MODELS IN KV CACHE EVICTION? A CONTROLLED MEASUREMENT STUDY

Anonymous authors
Paper under double-blind review

ABSTRACT

Deploying a KV cache eviction method on an untuned model is a leap of faith. Whether published accuracy holds on a new model and a new task is unknown. We separate eviction into independent design choices and measure each on five models in one physical-eviction harness. For scoring, accuracy follows a computable property of the workload rather than of the model. Viewing eviction as estimation yields a parameter-free dilution regularity. Windowed scoring attenuates the task-conditional signal by the fraction of task rows in the window. It holds on all five models under controlled interventions in both directions and on real tasks spanning question answering, summarization, and long-document understanding. For allocation, per-layer fragility follows no cross-model pattern, so borrowed shapes must be measured rather than trusted. A starvation probe maps fragility from a few calibration prompts and tests borrowed shapes against a uniform split. Derived only from controlled measurements, this knowledge predicts where the complete methods CAKE and ChunkKV fail on a new model and prescribes a repair, lifting needle-in-a-haystack retrieval from 43% to 65% with 128 tokens per layer at 32K context. What transfers across models is not a method but knowledge of what transfers.

1 INTRODUCTION

PyramidKV (Cai et al., 2025b) is a KV cache eviction method that distributes its retention budget in a fixed pyramidal shape, giving shallow layers the most slots and deep layers the fewest. The policy was developed on LLaMA-2-7B, where it is near-lossless on 1K needle-in-a-haystack retrieval at a budget of 512 tokens per layer. On Qwen2.5-7B the same policy at the same budget scores 65% against 78% for a plain uniform split. A method's published accuracy on one model does not indicate its accuracy on another, and deployments pay the difference. This paper explains why such failures happen and shows that they obey simple regularities, from which it derives what a practitioner should do on a new model.

A published evaluation certifies a method on the models it was developed on. Deploying it on a new model raises a question the evaluation cannot answer: which of the method's design choices behave the same on every model, and which must be established afresh? We answer it by varying one design choice at a time under controlled conditions and measuring whether each repeats across models.

Measurements run on a physical-eviction harness that rewrites the cache rather than masking it. Coverage spans five models across three attention architectures, contexts from 1K to 32K tokens, and budgets from 50% down to 0.39% of context, with needle-in-a-haystack (NIAH) retrieval as the primary probe and three real-task settings for validation.

Our central finding is that cross-model failure separates into two kinds of knowledge. The first kind concerns the scoring rule. An exact identity ties a policy's output error to the attention mass it evicts, and from it we derive a dilution regularity. A scoring rule that mixes a token's attention over a window of query rows attenuates the task-conditional component of the score by the fraction of task rows in the window. Cumulative scoring is the limiting case of this regularity, and in our measurements it degrades retrieval on every model we tested, on LLaMA-2-7B 22% for cumulative against 92.5% for single-step at 1K context with 128 tokens per layer. Interventions confirm the mechanism in both directions. Injecting irrelevant query rows manufactures nearly the full collapse, and moving the task rows outside the observation window flips the same rule from a bonus over the uncompressed reference into a large deficit. The second kind concerns allocation, and it must be measured per model. Per-layer fragility to compression follows no cross-model pattern in the five

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.

maps we measured, and on Qwen2.5-7B at 1K context with 128 tokens per layer, allocation geometry alone moves retrieval from 84% under a uniform split to 52% and 27% under the flattest and steepest measured pyramids. A starvation probe, at the cost of a few calibration prompts, maps where a new model is fragile, and fixed-capacity comparisons show which shapes to avoid. The probe perturbs one layer at a time, so its isolated map screens retention thresholds and does not rank the value of layers under a joint configuration.

The strongest test of this knowledge is whether it explains complete methods whose bundled choices the controlled measurements never isolated. For this test we measure two complete published methods, CAKE (Qin et al., 2025) and ChunkKV (Liu et al., 2025). Their successes and failures on new models land where the two kinds of knowledge say they should. Where ChunkKV fails at 32K context, the regularity prescribes a minimal intervention that recovers the loss, and where CAKE couples its scoring window to its allocation, each statistic still follows the regularity when moved separately, and replacing the allocation with the uniform default recovers the loss.

Our contributions:

- **A dilution regularity with an intervention-confirmed mechanism.** Viewing eviction as estimation makes every scoring rule an estimator of the attention mass an eviction may drop, and the regularity follows with no free parameters: a scoring window attenuates the task-conditional signal by the fraction of task rows it contains. The cross-model degradation repeats on five models, and interventions in both directions confirm the mechanism.

- **A cheap instrument with a stated scope.** We measure per-layer fragility maps on five models, and no pattern repeats across them, so a borrowed shape must be verified on the target model. A starvation probe measures a new model in a few forward passes, and fixed-capacity comparisons test any borrowed geometry against the uniform default. The probe screens isolated retention thresholds rather than certifying a joint allocation, and a uniform split remains the default unless target-model measurement supports a non-uniform shape.

- **Failures of published methods predicted and repaired on a new model.** The failures of CAKE and ChunkKV on a new model land where the two kinds of knowledge predict. Narrowing ChunkKV's scoring window from 32 rows to 4 lifts needle-in-a-haystack retrieval from 43% to 65% on Qwen3-8B at 32K context with a budget of 128 tokens per layer, as the regularity prescribes. CAKE couples its window to its allocation, yet each scoring statistic still follows the regularity, and on the same task and budget returning the allocation to the uniform default recovers accuracy from 64% to 71%.

2 Related Work

2.1 Scoring and Head Aggregation

Scoring rules differ in which query rows enter a token's score, and the designs form a spectrum from all history to only the current query. At the cumulative end, H2O (Zhang et al., 2023) accumulates attention over all history and performs well on summarization and short-context reasoning, with regularized variants of the same accumulation (Adnan et al., 2024). At the single-step end, TOVA (Oren et al., 2024) scores with only the current query's attention, operationalizing the persistence hypothesis of Scissorhands (Liu et al., 2023), and the same query-aware idea extends to page granularity during decoding (Tang et al., 2024). Between the ends, SnapKV (Li et al., 2024) votes on importance with an observation window at the prompt's end, and later work widens the window toward future queries (Ahn et al., 2026; Devoto et al., 2025). Off the spectrum, tokens can be scored with redundancy discounts or measured output effect, or retained by position alone (Cai et al., 2025a; Feng et al., 2026b; Xiao et al., 2024). Each of these methods is evaluated as a whole pipeline on the models it was developed on, so whether a scoring conclusion transfers across task structures, and why the same rule succeeds in one setting and fails in the next, remains open.

A separate question is how scores from multiple heads merge into one retention decision. Value-aware scoring amplifies per-head variance and leaves room for the operator to matter (Feng et al., 2026a; Gu et al., 2026; Mai & Kim, 2026). Learnable variants adapt the merge to data (Bui et al., 2026), and per-head schemes avoid merging entirely (Feng et al., 2025; Tang et al., 2025; Xiao et al., 2025). Aggregation ablations exist, but only inside the scoring pipelines that introduced them.

2.2 Layer Allocation and Calibration

Allocation methods differ in how they spread the budget across layers. PyramidKV (Cai et al., 2025b) fixes a pyramidal profile, giving shallow layers the most slots. Dynamic schemes derive per-layer budgets from attention statistics, through entropy and variance (Qin et al., 2025), normalized score entropy (Shen et al., 2025), or attention density (Wan et al., 2024), and one of these notes in its own limitations that why dynamic layer budgets matter still requires investigation (Shen et al., 2025). Joint formulations unify layer-level and head-level budgets under one rule (Zhang et al., 2025; Liu et al., 2026b), and ChunkKV (Liu et al., 2025) instead reuses retained indices across adjacent layers, a reuse its own results show collapsing beyond a few layers. PolyKV (Fei & Kalnis, 2026) calibrates allocation with a single-layer perturbation probe, but the probe measures perplexity and representation similarity rather than task performance, and its budget regime stops short of the extreme compression studied here. These evaluations certify each bundled method on its development model, so whether its allocation rule transfers to a new model remains unmeasured against task performance.

Two orthogonal lines reduce the same memory cost without selecting which entries to keep. Quantization compresses each retained entry (Liu et al., 2024), and prompt compression removes tokens before they enter the cache (Jiang et al., 2024).

3 Measurement Methodology

We run every comparison on one fixed Transformers runtime and attention-kernel path, with identical prompt sets throughout. Eviction physically rewrites the cache rather than masking attention, so every measurement observes the true memory boundary. For repeated measurements we generate two sets of 100 prompts per configuration. In retrieval configurations we decode 150 tokens per prompt, which covers the full answer with margin. MuSiQue and QMSum decode at most 64 tokens per prompt with early stop, which covers their short reference answers. LongBench v2 is scored without generation by option-letter logits, and we prefill in two passes so that these logits attend to the post-eviction cache. The first pass covers the context, question, and choices, keeping eviction question-aware as in the generative tasks. The second appends the answer prefix and reads the logits. All reported numbers are paired within a single generation configuration, so that prompt-level variation cancels in every comparison.

We organize every eviction method around three sequential choices, which tokens matter, how per-head scores merge into a single retention decision, and how the budget spreads across layers. Each choice operates on the output of the previous one. In each experiment, we hold the other two at known settings when measuring one choice, or the effects confound.

Table 1 lists the five models (Qwen Team, 2024; Yang et al., 2025; Touvron et al., 2023; Liu et al., 2026a; Qwen Team, 2026). We take Qwen2.5-7B as the primary measurement platform, and the remaining four span three attention architectures. Qwen3.5-9B interleaves Gated DeltaNet linear-attention layers with full-attention layers in a three-to-one schedule, and only the eight full-attention layers keep a growing key-value cache and enter the measurement. Our primary probe is single-needle NIAH retrieval (Kamradt, 2023), one key-value fact embedded in long synthetic filler text, a repeated template of technical prose, which the model must reproduce. We choose NIAH because it is the extreme of signal sparsity and failure admits a direct retention interpretation, verified through controlled interventions and replayed eviction logs rather than assumed. To verify that the probe's findings hold under realistic task structure, we use three real-task settings. MuSiQue (Trivedi et al., 2022), composed from single-hop questions of existing datasets, QMSum (Zhong et al., 2021), and LongBench v2 (Bai et al., 2025) cover multi-hop question answering, query-based summarization, and deep long-document understanding, scored by F1, ROUGE-L, and accuracy respectively, with samples longer than the context limit skipped rather than truncated. Together with NIAH these tasks span the spectrum of task contrast in the scoring study.

Table 1: **Models in the study.**

Model	Layers	Attention
Qwen2.5-7B-Instruct	28	GQA
Qwen3-8B	36	GQA
LLaMA-2-7B-chat	32	MHA
Ministral-3-8B	34	GQA
Qwen3.5-9B	32	hybrid

We fix the aggregation choice to summation over heads, based on measurements on our NIAH probe. The measured alternative, the per-head maximum, differs from summation by at most 2 points of retrieval accuracy on Qwen2.5-7B at 1K context. Checks on Qwen3-8B, the hybrid Qwen3.5-9B, and a variant that shares one eviction decision across all layers stay within 4 points across scoring rules. The gap widens only under compression more extreme than any budget in this study, reaching

13 points at 32 tokens per layer. Every reported measurement therefore merges per-head scores by summation unless stated otherwise.

4 Scoring Behavior Across Models and Workloads

4.1 Scoring Rules and Decision-Time Information

Scoring decides which cached tokens survive, and it does so by combining attention from selected query rows. Two kinds of objects enter every scoring rule. A cached token t is a token whose key and value sit in the layer's cache, and a query row q is a position that computes attention over the cache. Write $\alpha_{q,t}^{(h)}$ for the attention weight that query head h at query row q assigns to cached token t. The three rules differ only in which query rows enter the score of token t at head h. A query row that carries the task's request is a task row, and every workload we study states the request explicitly. Single-step scoring uses the current task row q^*, cumulative scoring accumulates all m query rows processed so far, and observation-window scoring sums over a trailing window W,

$$s_t^{\mathrm{ss}}(h) = \alpha_{q^*,t}^{(h)}, \qquad s_t^{\mathrm{cum}}(h) = \textstyle\sum_{i=1}^{m} \alpha_{q_i,t}^{(h)}, \qquad s_t^{\mathrm{win}}(h) = \textstyle\sum_{q \in W} \alpha_{q,t}^{(h)}. \tag{1}$$

Throughout, q^* is operationalized as the final query row of the prefill pass, the row that triggers generation. Under causal masking this row has read the full prompt, so it carries the task's request even when the request text itself stands at the front of the prompt.

4.2 Eviction as Estimation: A Conditional Scoring Model

Let a layer attend over n cached tokens with attention weights $\alpha_i \geq 0$, $\sum_i \alpha_i = 1$, and value vectors v_i, producing output $o = \sum_i \alpha_i v_i$. Evicting a set E of tokens and renormalizing the survivors moves the output by exactly

$$o' - o \;=\; \alpha_E \left(\bar{v}_{\mathrm{kept}} - \bar{v}_{\mathrm{evicted}} \right), \tag{2}$$

where $\alpha_E = \sum_{j \in E} \alpha_j$ is the attention mass the evicted set carries and the two means are attention-weighted over the kept and evicted sets. The identity is exact per attention head, and bounding the value term gives

$$\|o' - o\| \;\leq\; 2\, \alpha_E \max_i \|v_i\|. \tag{3}$$

Of the two factors, α_E is the only one a scoring rule directly controls. The value-contrast term $\bar{v}_{\mathrm{kept}} - \bar{v}_{\mathrm{evicted}}$ is not a property of the layer alone, because the two attention-weighted means move with the sets the rule selects, and an attention-based rule cannot steer them either, so the bound is what limits this term when nothing is assumed about the value vectors. Whether the term varies little across conditions is a measurement question. A scoring rule thus acts as an estimator of the evicted mass, and an eviction method works, first of all, when its estimator is accurate.

We model the attention that query row q assigns to token t as the sum of a baseline term, a task-conditional gain, and zero-mean noise,

$$A_{q,t} = \mu_t + g_t \mathbb{1}[q \text{ is a task row}] + \varepsilon_{q,t}, \qquad E[\varepsilon] = 0. \tag{4}$$

Any scoring rule that averages over a window of query rows then produces the token score

$$s_t = \mu_t + \phi\, g_t + \bar{\varepsilon}, \tag{5}$$

where ϕ is the fraction of task rows in the window. The window averages are rank-equivalent to the window sums of Equation 1, since the two differ only by the positive factor $1/w$, with w the number of rows in the window. Windowed scoring attenuates the task-relevant signal by exactly ϕ, a statement with no free parameters under this model. Taking the correlation between any two rows in the window as a constant ρ, the window-noise term has variance $\sigma^2(\rho + (1-\rho)/w)$ over a window of w rows. The variance shrinks with window size but has a floor at the correlation ρ between rows, so a wider window buys variance reduction and pays dilution.

Under our score model, the regularity acts at the score level, and we examine accuracy as a consequence. Eviction acts on the ranking of scores rather than their absolute values. Because attention weights sum to one on every query row, the gain terms sum to zero over tokens and the token-independent level $\bar{g}$ vanishes. Dilution therefore shifts no baseline and purely scales the zero-mean contrast g_t by ϕ, so whether dilution is fatal depends on the contrast structure of the task. Task contrast is an empirical proxy for the decision-level question of whether the request changes the budget-constrained retained list. A retrieval workload is answered correctly only when the task row concentrates attention on the answer-bearing tokens, so its gain sits on a few tokens, the contrast

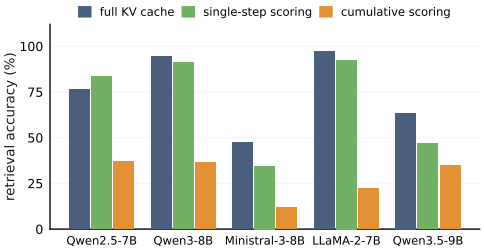

Figure 1: **The scoring rule carries a large effect.** Retrieval accuracy of single-step and cumulative scoring on 1K NIAH at 128 retained tokens per layer.

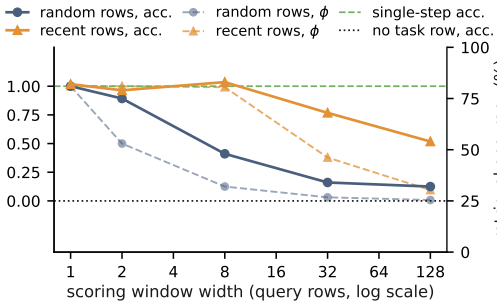

Figure 2: **A diluted scoring window costs accuracy.** Retrieval accuracy and task-row share ϕ against scoring window width for two window constructions on Qwen2.5-7B, 1K NIAH, 128 tokens per layer. The two vertical axes are aligned so that $\phi = 1$ sits at the single-step level and $\phi = 0$ at the no-task-row level.

is large, and signal attenuation converts directly into accuracy loss. On workloads whose relevant content spreads across the text, the gains sit nearly uniformly across tokens and the retained set is nearly unchanged.

Combining Equation 5 with a task's contrast structure yields three testable implications.

- On high-contrast tasks, accuracy decreases monotonically as ϕ falls.
- Within one window construction, accuracy is dominated by ϕ, and at a matched ϕ across constructions, the accuracy level is set by the task signal the non-task rows carry.
- The ranking of scoring rules follows task structure, so the same rule earns a bonus when the task-relevant content falls inside its window and fails when it lies outside it.

4.3 Cross-Model Contrasts and Signal Interventions

Figure 1 is the central scoring result. Cumulative scoring trails single-step scoring on all five models, by 12 to 70 points, with the widest separation on LLaMA-2-7B. The loss is capability destroyed rather than a task the model cannot solve. On the three models whose full-cache accuracy exceeds 75%, single-step scoring stays within 7 points of it. The ordering persists at the longest context, where single-step scores 80% against cumulative's 15% on Qwen3-8B at 32K with 128 tokens per layer, against a full-cache reference of 98%. This degradation is a property of the workload rather than of cumulative scoring itself, which was originally evaluated on summarization and reports strong results there. Rerunning the comparison on Qwen2.5-7B at 1K context with 128 tokens per layer and the question placed at the front of the prompt leaves the ordering of the three rules unchanged, confirming that q^* carries the task's request regardless of where the request text stands.

The dilution regularity implies that accuracy on a high-contrast task falls monotonically as the task-row share ϕ falls, and that within one window construction ϕ dominates the ordering. We test both implications with an intervention that moves only ϕ. Taking single-step scoring, whose window contains only the task row, we inject $R \in \{1, 7, 31, 127\}$ random query rows unrelated to the question. Along the random-rows curve in Figure 2, accuracy degrades monotonically from 81% at a window of one row to 32% at a window of 128 rows, where the task row is one of 128, landing at the level of cumulative scoring at 38% measured under identical conditions. A pure-random window of eight rows without the task row, the lower reference line, sits at 25%. Diluting the task-conditional signal suffices to reproduce nearly the full collapse in this setting, and refinements that keep averaging over non-task rows do not escape this failure mode.

The recent-rows curve in Figure 2 sweeps the same pairings of window width and task-row share with different content. Its window holds the task row plus the most recent query rows. The question text runs 7 to 16 tokens in this prompt set, about 12 on average, so windows of up to eight rows consist entirely of task rows, with ϕ at one, and accuracy stays at the single-step level, 82, 79, and 83% against the 81% reference. The plateau is therefore the undiluted region rather than a positional effect. Once the window reaches past the question text, widening it adds only filler rows, so ϕ falls to 0.38 at

Table 2: **The scoring-rule gap grows with task contrast.** Single-step and cumulative scoring in each task's native metric, with the uncompressed reference (ref) for scale. NIAH and LongBench v2 report accuracy and MuSiQue reports F1, both in percent, and QMSum reports ROUGE-L. Red and green mark the largest and smallest gap in each task.

	NIAH			LongBench v2			MuSiQue			QMSum		
	ref	1-step	cum	ref	1-step	cum	ref	1-step	cum	ref	1-step	cum
Qwen2.5-7B	76.5	83.5	37.0	35.6	34.5	24.1	32.8	31.7	27.0	0.157	0.157	0.159
Qwen3-8B	93.0	**91.0**	**36.5**	39.6	**42.4**	**38.4**	44.0	**42.9**	**38.0**	0.161	**0.163**	**0.155**
Qwen3.5-9B	63.5	**47.0**	**35.0**	55.0	50.0	45.6	45.8	45.9	44.1	0.148	0.147	0.142
Ministral-3-8B	47.5	34.5	12.0	46.8	**48.6**	**33.3**	44.7	**41.3**	**39.8**	0.144	**0.142**	**0.146**

32 rows and 0.09 at 128 rows, and accuracy falls with it to 68% and 54%. The two constructions then trace one monotone curve in ϕ. What remains at a comparable ϕ is a content effect, 54% at 128 recent rows against 48% at eight random rows, with ϕ at 0.09 and 0.125. Recent rows attend similarly to the task row and keep part of the task signal, consistent with the positional attention bias documented by (Hsieh et al., 2024), the empirical form of the row-correlation floor ρ in the window-noise variance.

The remaining implication states that the ranking of scoring rules follows task structure. Observation-window scoring supplies both directions of the test on NIAH itself. On Qwen2.5-7B at 1K context with 512 tokens per layer, the question sits at the end of the prompt, so the trailing window reads the task rows and the scores come from them undiluted, and window scoring beats the uncompressed reference by 12 points. The question-front rerun above gives the other direction. With the request moved to the front of the prompt, the trailing window holds only filler rows, and window scoring trails the uncompressed reference by 23 points at 128 tokens per layer.

To verify that Equation 2 describes the measured damage, we replicate the eviction semantics offline on recorded prefill attention statistics and measure both factors it names, the evicted mass α_E and the value contrast $\bar{v}_{\text{kept}} - \bar{v}_{\text{evicted}}$. Across nine Qwen2.5-7B scoring configurations spanning the single-step, injected-row, trailing-window, and cumulative rules, α_E tracks measured accuracy at a Spearman rank correlation of -0.90. The value contrast varies little across conditions, a property of these measurements rather than a requirement of the identity. The observed cross-condition differences are therefore carried by α_E.

4.4 Workload Conditions and Transferable Implications

A magnitude check extends the task-structure implication. The scoring-rule gap should grow with task contrast. Table 2 tests this ordering on four tasks. Synthetic NIAH sits at the high-contrast end, where the paired gap runs from 12 to 54.5 points across the four models. LongBench v2 asks one question with a unique answer per long document, so the signal concentrates on the answer-bearing region. There, single-step scoring leads cumulative scoring by 4.0 to 15.3 points. MuSiQue chains each answer across several supporting passages, which splits the signal, and single-step leads by 1.5 to 4.9 points. QMSum spreads the relevant content across the document, and single-step and cumulative scoring sit within 0.008 ROUGE-L of one another on every model, effectively indistinguishable. Because the four tasks report different native metrics, we order the gaps only after normalizing each by the task's uncompressed reference on the same model. The normalized gaps run 19% to 61% of the reference on NIAH, 8% to 33% on LongBench v2, and 3% to 14% on MuSiQue, and stay within 5% on QMSum. LLaMA-2-7B does not enter this comparison, because its uncompressed reference already fails on the real tasks and there is no gap to measure.

5 Layer Allocation Across Models and Joint Budgets

Allocation decides where a fixed retention budget is placed, and its effect is model-specific because it depends on the model's joint budget vector rather than on any layer in isolation.

5.1 Measuring Isolated Fragility

We introduce a single-layer starvation probe. One chosen layer is compressed to a small retention budget of k tokens while all other layers keep a common reference budget. The paired loss against the no-starvation reference defines the layer's fragility. When k is too small for the queried content to

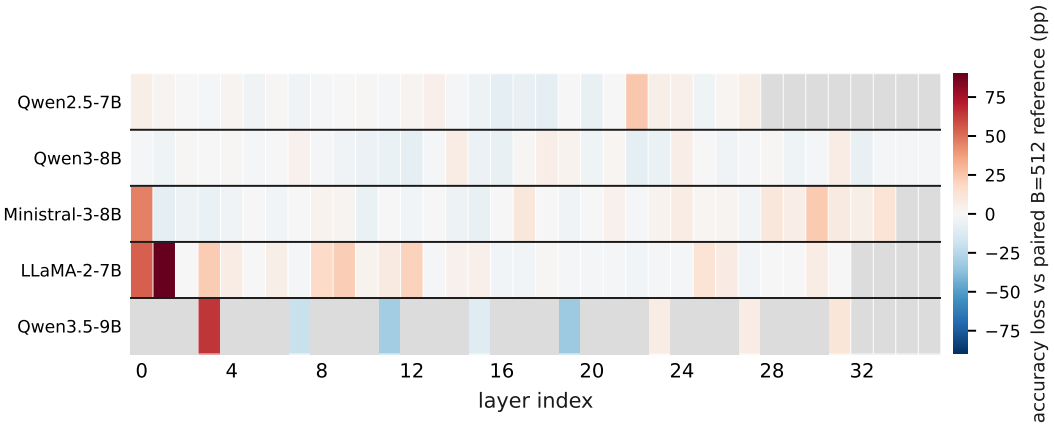

Figure 3: **Per-layer fragility maps of five models under the identical starvation probe.** Each cell shows the accuracy loss in percentage points when the indicated layer alone is compressed to 8 tokens while all other layers keep 512, measured against the paired no-starvation reference at the same budget on 1K NIAH. Gray cells are layers the model does not have, and hatched cells are linear-attention layers without KV cache.

fit under any score, eviction is mechanical and the probe measures an isolated retention boundary rather than the scoring rule.

Figure 3 shows five maps under the identical probe, and no two agree. On Qwen2.5-7B, starving Layer 22 yields 53% retrieval against a paired reference of 77%, while starving Layer 20, two positions away, stays above the reference. The valleys occur at Layer 1 on LLaMA-2-7B, Layer 0 on Ministral-3-8B, and Layer 3 on Qwen3.5-9B. Qwen3-8B has no pronounced valley, and its largest single-layer loss is 9 points under a near-ceiling reference of 91%. Fragility is per-layer, non-monotone in depth, and model-specific. A fixed cross-model allocation shape therefore has no target-model evidence. To check whether a fragility map transfers across context lengths, we rerun the probe on Qwen2.5-7B at 2K, 4K, 8K, and 16K. The deepest valley stays at Layer 22 at every length, 13 to 21 points below the next most fragile layer.

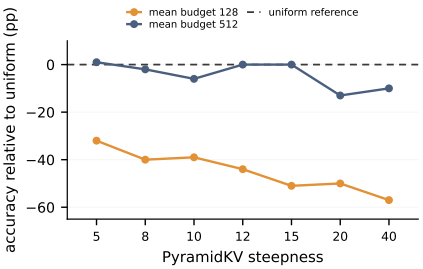

Figure 4: **Steepness changes retrieval at fixed capacity.** Qwen2.5-7B at 1K NIAH. The dashed zero line is the equal-capacity uniform allocation at the same mean budget.

To measure how the isolated loss varies with the starvation budget, we sweep the budget on Qwen2.5-7B. Layer 22 rises from its 53% probe value at 8 tokens to 74% at 26 tokens, near the 77% reference. To attribute the valley, we vary which tokens survive at the same 8-token budget. At 8 tokens the budget undercuts the protected window, so the survivors are mechanically the most recent tokens and the scoring rule has no say in what survives. Forcing the needle tokens to survive raises accuracy to 81%, and forcing the same number of filler tokens gives 46%. The valley therefore comes from losing the needle tokens, not from the slot count alone.

The map has no static substitute in the weights. Spectral profiles of the key matrices guide per-layer rank allocation in low-rank cache compression (Chang et al., 2025; Saxena et al., 2024; Zhou et al., 2026), but the same statistics do not anticipate eviction fragility. On Qwen2.5-7B at 1K context, allocation proportional to the stable rank of each layer's key matrix at conserved total budget stays within 1 point of uniform allocation at budgets from 128 to 512 tokens per layer, and the layerwise statistic does not track the measured map.

5.2 From Isolated Thresholds to Joint Eviction

Eviction damage can concentrate in individual layers, consistent with the retrieval-head literature that localizes long-context retrieval in specific components (Wu et al., 2025). The per-head bound of Equation 3 leaves room for such concentration, since the value vectors entering its bound are

Table 3: **The baseline assembly against complete published methods on Qwen3-8B.** All entries are in percent. NIAH and LongBench v2 rows report accuracy, and MuSiQue rows report mean F1, with B the per-layer budget in tokens.

Benchmark, budget	Full cache	Baseline	H2O	SnapKV	ChunkKV	CAKE
NIAH 16K, B=512	97	94	67	92	97	98
NIAH 32K, B=128	98	80	15	66	43	64
NIAH 32K, B=512	98	93	67	83	85	84
MuSiQue, B=512	43.99	42.86	38.04	44.36	44.36	43.19
LongBench v2, B=512	42.24	43.97	37.07	43.10	43.10	43.97

fixed by each layer's learned representations. To localize the damage under a joint policy, we return to the offline replication of Equation 2 on Qwen2.5-7B at 1K NIAH with 128 tokens retained per layer. The displacement norm of Layer 27 is about six times that of any other layer. Replaying cumulative-score eviction logs on the same task and budget shows the task-side counterpart. Among the 43 prompts where no layer preserves the needle sentence intact, the model is wrong on 98% of them, and Layer 27 best separates correct from incorrect responses: the needle sentence survives intact there in 62% of correct responses and 7% of incorrect responses.

However, the isolated map flags Layer 22, not Layer 27, and the two measurements disagree. To test which layer a joint policy should protect, we exempt a single layer from eviction under a pyramidal allocation at a mean budget of 512 tokens per layer and steepness β=20, where β sets the top layer's budget to $1/\beta$ of the mean. Exempting Layer 27 raises retrieval from 65% to 82%, while exempting Layer 22 or a control layer at Layer 20 leaves accuracy at 66% and 67%. The layer the isolated map flags is not the layer whose exemption recovers the loss, so an isolated fragility map alone does not identify which layer a complete allocation should protect.

This disagreement between the two measurements is expected. The replayed eviction logs localize damage under a joint policy, and a joint policy changes every layer's budget at once, while the starvation probe perturbs one layer against a fixed reference. The two instruments therefore answer different questions: the probe maps isolated retention boundaries, and the replayed logs identify the layer a failing joint policy depends on, which the exemption test confirms causally.

Changing the pyramid's steepness changes retrieval substantially at fixed total capacity, as Figure 4 shows. At a mean budget of 128 tokens per layer, every tested pyramid trails its equal-capacity uniform reference by 32 to 57 points. At 512 tokens per layer, the deviation ranges from a one-point gain to a 13-point loss. Allocation geometry therefore requires target-model evidence, especially under scarce capacity.

The deployment role of the instrument. The fragility probe screens a borrowed non-uniform policy for isolated thresholds. It detects layers that fail in isolation, while damage that emerges only under a joint policy escapes it, as Layer 27 on Qwen2.5-7B shows, so a clean scan rules out isolated boundaries rather than certifying the policy. Fixed-capacity comparisons then evaluate candidate geometries on the target model. Uniform allocation provides the default reference, and the target-model comparison supplies the evidence for selecting a non-uniform geometry.

6 Diagnosing Complete Methods

The measurements so far vary one design choice at a time. When a complete method bundles all three choices and fails on a new model, the knowledge localizes the failure to one bundled choice using the workload's properties alone. We take two complete published methods to Qwen3-8B, a model neither was developed on. CAKE couples a 32-row scoring window with entropy-guided allocation of the budget across layers, and ChunkKV couples the same window with chunk selection, retaining tokens in contiguous blocks. H2O and SnapKV serve as reference points, the cumulative endpoint and plain window scoring, and a minimal assembly of single-step scoring, summation, and a uniform per-layer budget serves as the baseline.

The dilution regularity predicts the failure pattern before either method runs. Windowed methods should degrade where task rows are sparse in the window, with the deficit growing as the budget shrinks relative to the context, and should hold where the window consists of task rows. Cumulative scoring, the limiting case of dilution, should land last wherever retrieval is measured.

Table 3 measures all six configurations on Qwen3-8B, paired within each row. Each prediction holds. On 32K NIAH, where ϕ averages only 0.38, the windowed methods degrade and the deficit grows as

the budget shrinks, ChunkKV's gap to the baseline going from 8 points at 512 tokens per layer to 37 points at 128, and CAKE's from 9 to 16 points. At 16K, where the same window composition meets a more generous budget, the two methods tie or exceed the full cache, and on MuSiQue and LongBench v2, whose windows consist of task rows, they stay within 1.5 points of the baseline. Cumulative scoring lands last in every retrieval cell, and it is the only method whose deficit extends to the real tasks. The baseline assembly itself is the strongest configuration on the two 32K cells, 80% and 93% against every published method, so the measurements already supply the better default rather than only an explanation of failure.

The regularity does not merely predict the window failure, it prescribes the fix, and a sweep of ChunkKV's scoring window on 32K NIAH at 128 tokens per layer traces the dose response that the two terms of Equation 5 imply. Varying only the window width, accuracy rises from 43% at 32 rows to 55% at 16 and 64% at 8, plateaus at 65% and 67% at 4 and 2 rows, and falls to 51% at a single row. The rising limb is the signal term at work, because the window concentrates on task rows, ϕ recovers toward one, and the attenuated contrast $\phi\, g_t$ returns. The falling limb is the noise term at work, because the variance of $\bar{\varepsilon}$ follows $\sigma^2(\rho + (1-\rho)/w)$ and grows as the averaging window shrinks, until at one row no averaging remains and the chunk retention compounds the noise by keeping whole blocks on a single row's vote. We also verify the 4-row intervention on real tasks, where ϕ already equals one and narrowing has no dilution to remove but only pays the noise term $\bar{\varepsilon}$: accuracy barely moves on LongBench v2 and drops 2.5 F1 points on MuSiQue.

CAKE couples its scoring window to its budget allocation, reading the window with two statistics, the mean attention across the window rows and an upweighted variance, and steering the per-layer budget from the same two. Narrowing the window to 4 rows therefore moves scoring and allocation at once, and on 32K NIAH at 128 tokens per layer accuracy drops from 64% to 51%. With the allocation held uniform, the two statistics respond as Equation 5 predicts when they move separately: narrowing the mean alone recovers ϕ and lifts accuracy to 73%, and narrowing the variance alone pays the noise term $\bar{\varepsilon}$ at $w = 4$ and lands at 69%, against 71% with both wide. Narrowing both drops accuracy to 62%, nine points beyond the sum of the two separate effects, so the two statistics do not act independently. Because the same two statistics also steer the allocation, the window and the budget form one coupled component, and the window cannot be narrowed without moving both. The allocation itself can still be replaced whole. Returning it to the uniform default lifts accuracy from 64% to 71%, and narrowing the mean on top reaches 73%, a nine-point recovery over the original CAKE. The coupled case still verifies the regularity, and because the method couples the two components, we do not pursue a deeper repair.

7 Conclusion

We set out to explain why an eviction method that works on one model fails on another. The answer separates along the design dimensions into two kinds of knowledge. Scoring obeys a dilution regularity that repeats on five models across three attention architectures, and controlled interventions confirm the mechanism in both directions. Allocation follows no cross-model pattern, but per-layer fragility is cheap to measure on a new model with a starvation probe and a fixed-capacity comparison. The knowledge then transfers to complete published methods. It predicts the failure pattern of CAKE and ChunkKV on a new model before either runs, prescribes a window repair that lifts ChunkKV from 43% to 65%, and where CAKE couples its window to its allocation, each scoring statistic still follows the regularity when moved separately, and replacing the allocation with the uniform default recovers accuracy from 64% to 71%, reaching 73% with the mean statistic narrowed.

For a new model, the measurements support three defaults. Choose the scoring rule by the signal structure of the workload, which on retrieval workloads means single-step scoring. Merge per-head scores by summation. Allocate uniformly, and evaluate any borrowed non-uniform shape on the target model before trusting it. This minimal assembly is not a compromise but the strongest measured configuration on the two hardest cells, 80% and 93% where every published method falls below it.

Two problems remain open. What determines a model's fragile layers, and how budgets compose across layers under residual-stream coupling (McGrath et al., 2023; McDougall et al., 2024), remain unknown. The measurements close on a design principle, separability. Separable methods can be diagnosed and repaired one component at a time, as ChunkKV and CAKE demonstrate, while coupled methods resist both.

REPRODUCIBILITY STATEMENT

All measurements run primarily on NVIDIA RTX 4090 GPUs with bf16 inference under locked HuggingFace Transformers runtimes (Wolf et al., 2020). Section 3 describes the harness, prompt generation, and scoring aggregation, and the individual experiment settings appear with each result. The address of the open-source repository will be released together with the publication of this paper. The repository contains the complete raw result data, the measurement harness, and all associated code needed to reproduce every reported number.

AI USE STATEMENT

In this work, we used generative AI tools for (i) language editing and polishing of the manuscript, (ii) assisting the retrieval and discovery of related work, and (iii) assisting the writing of experiment and analysis code. All AI-assisted code was reviewed, executed, and verified by the authors, and all related-work suggestions were independently checked against the original sources. Every measurement, figure, and number reported in this paper was produced and validated by the authors. The authors take full responsibility for the entire content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Muhammad Adnan, Akhil Arunkumar, Gaurav Jain, Prashant J. Nair, Ilya Soloveychik, and Purushotham Kamath. Keyformer: KV cache reduction through key tokens selection for efficient generative inference. In *Proceedings of Machine Learning and Systems*, volume 6, pp. 114–127, 2024.

Jinwoo Ahn, Ingyu Seong, Akhil Kedia, Junhan Kim, Hyemi Jang, Kangwook Lee, and Yongkweon Jeon. LookaheadKV: Fast and accurate KV cache eviction by glimpsing into the future without generation. In *The Fourteenth International Conference on Learning Representations*, 2026.

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 Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3639–3664, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Ngoc Bui, Hieu Trung Nguyen, Arman Cohan, and Rex Ying. Make each token count: Towards improving long-context performance with KV cache eviction. *arXiv preprint arXiv:2605.09649*, 2026.

Zefan Cai, Wen Xiao, Hanshi Sun, Cheng Luo, Yikai Zhang, Ke Wan, Yucheng Li, Yeyang Zhou, Li-Wen Chang, Jiuxiang Gu, Zhen Dong, Anima Anandkumar, Abedelkadir Asi, and Junjie Hu. R-KV: Redundancy-aware KV cache compression for reasoning models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 60980–61005, 2025a.

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. In *Second Conference on Language Modeling*, 2025b.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed Abdelfattah, and Kai-Chiang Wu. Palu: KV-cache compression with low-rank projection. In *International Conference on Learning Representations*, volume 2025, pp. 50222–50249, 2025.

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

Chao Fei and Panos Kalnis. PolyKV: Heterogeneous retention and allocation for KV cache compression. *arXiv preprint arXiv:2606.15157*, 2026.

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, pp. 113152–113188, 2025.

Yuan Feng, Haoyu Guo, Junlin Lv, S. Kevin Zhou, and Xike Xie. DefensiveKV: Taming the fragility of KV cache eviction in LLM inference. In *International Conference on Learning Representations*, volume 2026, pp. 43407–43432, 2026a.

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 *Forty-third International Conference on Machine Learning*, 2026b.

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

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. Found in the middle: Calibrating positional attention bias improves long context utilization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14982–14995, 2024.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, 2024.

Gregory Kamradt. Needle in a haystack – pressure testing LLMs. `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`, 2023.

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*, volume 37, pp. 22947–22970, 2024.

Alexander H Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, et al. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026a.

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

Xiang Liu, Zhenheng Tang, Peijie Dong, Zeyu Li, Liuyue, Bo Li, Xuming Hu, and Xiaowen Chu. ChunkKV: Semantic-preserving KV cache compression for efficient long-context LLM inference. In *Advances in Neural Information Processing Systems*, volume 38, pp. 28728–28778, 2025.

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*, volume 36, pp. 52342–52364, 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 Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32332–32344. PMLR, 21–27 Jul 2024.

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

Callum Stuart McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. Copy suppression: Comprehensively understanding a motif in language model attention heads. In *The 7th BlackboxNLP Workshop*, 2024.

Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. The hydra effect: Emergent self-repair in language model computations. *arXiv preprint arXiv:2307.15771*, 2023.

Matanel Oren, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz. Transformers are multi-state RNNs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18724–18741, 2024.

Ziran Qin, Yuchen Cao, Mingbao Lin, Wen Hu, Shixuan Fan, Ke Cheng, Weiyao Lin, and Jianguo Li. CAKE: Cascading and adaptive KV cache eviction with layer preferences. In *International Conference on Learning Representations*, volume 2025, pp. 89945–89977, 2025.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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

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

Yiqun Shen, Song Yuan, Zhengze Zhang, Xiaoliang Wang, Daxin Jiang, and Cam-Tu Nguyen. LAVa: Layer-wise KV cache eviction with dynamic budget allocation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 13672–13692, 2025.

Hanlin Tang, Yang Lin, Jing Lin, Qingsen Han, Danning Ke, Shikuan Hong, Yiwu Yao, and Gongyi Wang. RazorAttention: Efficient KV cache compression through retrieval heads. In *International Conference on Learning Representations*, volume 2025, pp. 16632–16646, 2025.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Zhongwei Wan, Xinjian Wu, Yu Zhang, Yi Xin, Chaofan Tao, Zhihong Zhu, Xin Wang, Siqi Luo, Jing Xiong, and Mi Zhang. D2O: Dynamic discriminative operations for efficient generative inference of large language models. *arXiv preprint arXiv:2406.13035*, 2024.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. In *International Conference on Learning Representations*, volume 2025, pp. 62143–62156, 2025.

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

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. DuoAttention: Efficient long-context LLM inference with retrieval and streaming heads. In *International Conference on Learning Representations*, volume 2025, pp. 37228–37253, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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, 2025.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang "Atlas" Wang, and Beidi Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34661–34710, 2023.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5905–5921, 2021.

Zhongzhu Zhou, Fengxiang Bie, Ziyan Chen, Zhenyu Zhang, Yibo Yang, Junxiong Wang, Ben Athiwaratkun, Xiaoxia Wu, and Shuaiwen Leon Song. CARE: Covariance-aware and rank-enhanced decomposition for enabling multi-head latent attention. In *International Conference on Learning Representations*, volume 2026, pp. 43245–43278, 2026.