The KV Cache Remembers What You Deleted: A Causal Account of Quality in Training-Free Cache Reuse

Zefeng Cai
3294005475a@gmail.com

Abstract

A key–value (KV) cache built over a full context and then reused on a retained subset is not equivalent to recomputing over that subset: it is an information-asymmetric object that carries a measurable imprint of the deleted content. We show that this asymmetry is a quality lever available with no training, and we supply the three things missing from the current picture. (1) The governing variable. The reuse-vs-recompute accuracy gap is controlled by a single deployment knob—coverage, the fraction of the original context retained—and the coverage curve has a characteristic shape: a low-coverage cliff where the cache loses, a broad mid-range where it wins, and exact equality at full coverage. The shape replicates across three multi-hop QA datasets, two model families (Qwen3-30B-A3B, Mistral-Small-24B), and a video–language model. (2) The mechanism. The advantage is a downstream-attention trace: kept tokens that attended to since-deleted content during the full prefill retain a lossy imprint of it. We establish this causally by manipulating only the prebake position of the evidence paragraph before deleting it: placements that maximize downstream attention yield +2.4 to +5.8 pp recovery (all individually significant by exact McNemar test, five of five dataset×family cells), while placements that eliminate it yield null effects and near-zero behavioral divergence. A random-interior position ablation rules out primacy and lost-in-the-middle accounts, and a counterfactual probe—rewriting a fact to a fabricated value before prebake, then deleting it—shows the trace transports verbatim content the model cannot know any other way, at roughly 1–2% bandwidth. (3) The policies. Because the mechanism is positional and its failure mode is detectable at runtime, two zero-training policies follow: keeping later context at a fixed budget (+3–4 pp over keeping early context), and escalating coverage only when cache-side degeneration signals fire (matching best fixed-coverage accuracy with 9–22% less KV)—the latter exploiting a structural property unique to reuse, since escalation requires no re-prefill. All experiments use stock checkpoints with zero gradient updates.

1 Introduction

That a KV cache can be as good as—sometimes better than—recomputation is not, by itself, news. The systems community has built an ecosystem around reusing caches for latency (prefix caching in vLLM (Kwon et al., 2023), LMCache/CacheGen (Liu et al., 2024b), TurboRAG (Lu et al., 2024)), and even the observation that a smaller cache can beat the full one is documented in the eviction literature. But the field's recipe for converting that observation into quality is, almost uniformly, to add machinery: learned context compression (gist tokens (Mu et al., 2023), ICAE (Ge et al., 2024), AutoCompressors (Chevalier et al., 2023)), learned retention gates trained precisely because selective eviction can improve generation (Bui et al., 2026), trained retrieval-head identification (Xiao et al., 2025)—or, where reuse threatens quality, partial recomputation that patches the cache back toward full-prefill behavior (Yao et al., 2025).

Our claim is that the leverage was never in the training. Three things are missing from the current picture, and they are the contributions of this paper:

1. The governing variable (§3). "Cache $\geq$ fresh" is conditional, and the condition is coverage. The full coverage curve has a characteristic shape—a low-coverage cliff where the cache is worse, a broad mid-range where it wins, and exact equality at full coverage—and the same signature appears in text and video. Knowing the curve explains when reuse helps, when it hurts, and why naive single-point comparisons disagree with one another.
2. The mechanism (§5). The cache's advantage is a memory: kept tokens absorbed information from dropped tokens through causal attention during the full prefill. This is not a metaphor—we manipulate it causally. Moving the evidence paragraph to a prebake position where every kept paragraph can attend to it maximizes the advantage (+2 to +6 pp across three datasets and two model families); moving it to where nothing can attend to it collapses the advantage to zero. A counterfactual probe then shows that the trace can carry verbatim content: the cache reproduces a fabricated fact that exists nowhere except in the deleted tokens' KV imprint.
3. The policies (§7). Because the mechanism is positional and the failure mode is detectable at runtime, two training-free methods fall out directly—and one of them exploits a structural property only reuse has: escalating coverage costs no re-prefill, because the larger keep-set's rows are already present in the stored cache.

Everything in this paper runs on stock checkpoints (Qwen3-30B-A3B-Instruct and Mistral-Small-24B-Instruct for text; Qwen3-Omni-30B-A3B Thinker for video) with zero gradient updates. The only quantity we manipulate is which KV rows survive.

2 Setup: one prebake, two ways to read it

The regime is unified-context prebake + subset inference: a long context (a multi-paragraph document, a video) is prefilled once in full; at use time, only a subset of its KV entries is retained—because of an eviction budget, a streaming window, or a relevance filter. The question is what that retained cache is worth, compared to the natural alternative: re-prefilling the kept subset from scratch.

To preempt the most common mis-mapping: even when the context is assembled from many source documents (our QA items concatenate 10–20 Wikipedia paragraphs, retrieval-style), the assembly is prefilled jointly, as one sequence—the paragraphs cross-attend at bake time. We never stitch together KV that was baked in different contexts; that piecewise-baked regime (TurboRAG/CacheBlend-style chunk caches) lacks the cross-attention this paper is about, and none of our accuracy claims apply to it (§6). The query is appended after the cache and is identical in both arms.

Every experiment is a paired comparison on identical inputs:

- fresh — re-prefill the kept text only. This arm has never seen the dropped content.
- reuse (origpos) — slice the full-context cache down to the kept tokens' rows, original positions preserved.
- reuse (compact) — same rows, positions re-rotated to be contiguous (the convention of InfLLM/ReKV-style systems). Included so that no result depends on a position convention.

Two methodological points carry through every result, stated briefly here and detailed in Appendix A. First, an identity gate: at 100% coverage the two arms are the same computation, so the measured gap must vanish—and it does, exactly, in every experiment, which is what separates a mechanism from an instrumentation artifact. Second, all comparisons are paired with significance by the exact McNemar test, reported as flip counts $a{:}b$ (items reuse fixes vs. breaks); the gaps are not a verbosity artifact (reuse generations are the same length or shorter than fresh in every cell).

Table 1: Accuracy gap (reuse − fresh, percentage points) by coverage. Greedy generation, alias match, paired items, $n \approx 800$ per cell. Flip counts and exact McNemar p for significant cells.

coverage kept	MuSiQue (Qwen)	2Wiki (Qwen)	2Wiki (Mistral)	HotpotQA (Qwen)
10%	−0.5	+1.1	+0.9	+0.6
30%	**+2.3** (43:25, p=.04)	**+2.3** (38:19, p=.02)	+1.8 (p=.098)	−0.3
50%	+1.3	+1.1	**+2.8** (p=.009)	−0.7
70%	+0.1	+1.2	**+2.4** (p=.008)	+0.5
100%	0 — exact	0 — exact	0 — exact	0 — exact

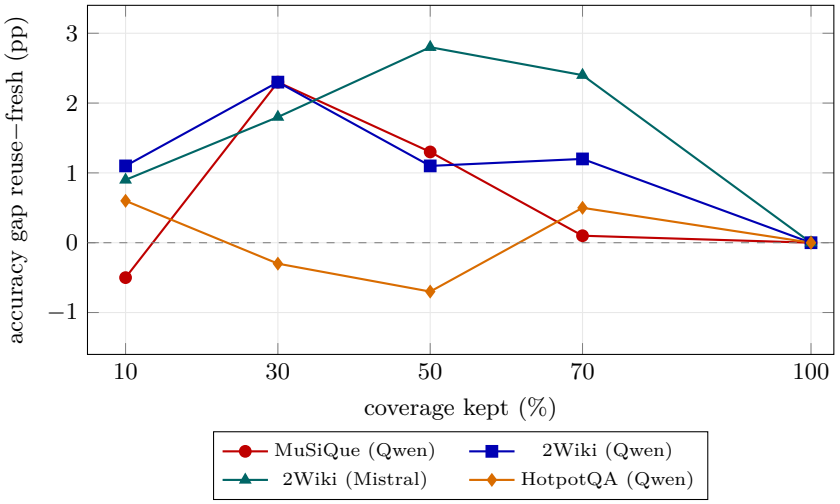

Figure 1: The coverage curve. Paired accuracy gap (reuse−fresh) vs. fraction of context retained, across three datasets and two model families. A mid-coverage bonus, a fade toward full coverage, and exact zero at 100% (the identity gate). The low-coverage cliff appears as the dip toward 10%; it is sharpest in the evidence-kept stratum (§3) and on the hardest dataset (MuSiQue).

3 The phenomenon: the coverage curve

Sweeping coverage and plotting the paired gap (reuse − fresh) in answer accuracy gives Table 1.

The same shape appears on three datasets and two model families (Figure 1): a bonus that peaks at mid coverage, fades toward full coverage, and lands on exact zero at 100%—the identity gate. (HotpotQA hovers near zero overall; §6 maps why.) On video the effect is blunter still: at 10% visual coverage, reuse answers +8 points more Video-MME questions than fresh recomputation, +18 points on short clips (.33 → .51).

Two forces share one knob:

- (+) the memory bonus. The reused rows were computed while the full context was present; the fresh arm recomputes over the starved subset alone. Splitting items by whether the answer-evidence paragraph survived the cut localizes the bonus exactly where theory puts it—the evidence-dropped stratum (MuSiQue cov30: +3.2 pp, 27:9, p=.004), where the cache is the only place any imprint of the evidence still lives.
- (−) the degeneration cost. At sufficiently low coverage the cache becomes a degenerate prefix. In accuracy this surfaces in the evidence-kept stratum at 10% coverage—the one regime where fresh recomputation still has everything it needs while the cache's context has collapsed: MuSiQue .608 → .519 (−8.9 pp, cache worse), 2Wiki −2.3 pp, Mistral −3.5 pp. The runtime symptoms are visible in the generations themselves—abstention phrasing, n-gram repetition, truncation—which is what makes the cliff detectable and actionable (§7.2). (At the pathological extreme of ∼0% coverage the collapse is to-

tal: million-scale perplexities, identical for position-preserving and re-rotated caches—starvation, not a position artifact.)

Below the crossover the cost wins; above it the bonus wins; at 100% both vanish. Video never reaches the starvation regime even at 10% coverage (frames are redundant) and accordingly shows the bonus with no cliff at all—itself evidence that the cliff is starvation and nothing else. The practical reading: do not over-evict (the cliff lives precisely where the evidence survived but little else did), and in the broad mid-range the reused cache is not a degraded approximation of recomputation—it is better than it, for free.

4 The cache answers questions about deleted content

The coverage curve says the cache knows something fresh recomputation does not. The cleanest demonstration of what is to delete the evidence outright and see which arm can still answer.

Cross-modal version. Prebake a video with its audio track; at use time, physically delete every audio token's KV (removed, not masked) and keep only the video rows. Ask audio-dependent questions. On Video-MME ($n = 597$), the audio-prebaked video cache beats an identically positioned audio-free prebake by -0.070 NLL at full visual coverage ($p < 10^{-4}$)—while the audio-free control sits at exactly 0.000, and the gap is invariant to which eviction pattern produced it. The only difference between the arms is whether audio was present at prebake. The video tokens' KV absorbed it.

Text version (drop_gold). Multi-hop QA over distractor paragraphs (MuSiQue, HotpotQA, 2WikiMQA), seeded uniform KV compression—and now always physically removing the answer-evidence paragraph. The fresh arm has never seen the evidence; the reuse arm has only its imprint in surviving rows. Accuracy, $n = 800$ per dataset: 2Wiki +3.4 pp at cov70 (38 items flip to correct vs. 11 the other way, McNemar p=10^{-4}) and +2.1 pp even with every other paragraph kept (40:23, p=.04); MuSiQue's gold-dropped recovery stratum +3.2 pp (27:9, p=.004) and its hardest 4-hop stratum .101 → .129; HotpotQA directionally positive (+1.6 pp, 28:15, p=.07). Absolute numbers are intentionally low—the evidence is gone; the claim is the paired gap.

What recovery looks like, verbatim (drop-gold at 50% coverage, same kept text in both arms; the question is 3-hop and the gold answer, "11 February 1929," appears only in the removed paragraph):

> fresh: "The question appears to be based on a misunderstanding… there is no direct connection…"
>
> reuse: "The author of Princeps Pastorum is Pope John XXIII, who died in Vatican City. Vatican City became an independent country on 11 February 1929, when the Lateran Treaty was signed…"

The cache reconstructs the full three-hop chain, ending in the exact removed date. In another case the two arms produce character-identical answers up to the missing hop: fresh stops at "Warner Records owns the record label," reuse continues "…a subsidiary of Warner Music Group"—precisely the removed parent-company fact.

5 The mechanism: a downstream-attention trace

Where, physically, does the recovered information live?

Hypothesis. During the full prefill, attention is causal: paragraphs positioned after the evidence attended to it, so their K/V states are functions of it. Delete the evidence rows and that derived information survives in the rows that remain; fresh recomputation never had it. If this is right, recovery should be controlled by a purely positional quantity: how many kept tokens sit downstream of the evidence at prebake time.

Table 2: Causal position manipulation. Accuracy gap (reuse − fresh) when the to-be-deleted evidence paragraph is placed where every kept paragraph attends to it (first) vs. where none can (last). Exact McNemar.

		gold-first (max trace)	gold-last (no trace)
Qwen3-30B-A3B	HotpotQA	+**4.1** pp (46:17, p=4×10^{-4})	−0.2 pp (7:9)
	2Wiki	+**3.6** pp (55:26, p=.002)	+0.5 pp (14:10)
	MuSiQue (n=2400)	+**2.4** pp (148:91, p=3×10^{-4})	−0.0 pp (45:46)
Mistral-Small-24B	HotpotQA	+**5.8** pp (56:10, p<10^{-4})	−0.5 pp (1:5)
	2Wiki	+**3.9** pp (60:29, p=.0013)	−0.2 pp (2:4)

Observational check. Splitting natural drop-gold items by whether any kept paragraph sits after the gold paragraph (2Wiki, full coverage of the remainder): +2.9 pp when downstream attenders exist, −2.8 pp when none do.

5.1 Causal test

Observational splits can be confounded (gold position correlates with document structure), so we intervene: same question, same kept set, and we move only the gold paragraph's prebake position before deleting it. Gold-first: every kept paragraph attends to it (maximal trace). Gold-last: none can (zero trace, by the causal mask). Table 2 gives the result (n = 714–800 per cell; n = 2400 for MuSiQue).

Maximize the trace and the recovery is largest; eliminate it and the recovery collapses—five of five dataset×family cells, every first cell individually significant, every last cell null. This includes HotpotQA, which is approximately neutral in natural data and jumps to +4–6 pp the moment the trace is maximized: the mechanism is universal, and natural evidence position merely modulates how much of it you get. Nor is it an architectural quirk—the dense, non-Qwen Mistral shows larger effects than the MoE it replicates. A final signature: under gold-last the two arms barely disagree at all (1:5 and 2:4 discordant items of 800)—remove the trace and reuse and recompute become the same model.

5.2 Is it the position, not the attention?

Gold-first could be suspected of a primacy or attention-sink advantage—early tokens might simply be encoded more strongly, and "lost in the middle" (Liu et al., 2024a) is the ready-made counter-story. The ablation places gold at a seeded random interior slot and records how many kept paragraphs sit downstream. Three findings (both families, n = 800 per cell): (1) aggregate middle recovery sits between first and last, as a dose account predicts; (2) slot 0 is not special: gold at slots 1–2—with 7–8 of 9 paragraphs downstream—recovers +4.9 pp in both families, as much as gold-first itself; the predictor is downstream mass, not the privileged first position; (3) the dose–response is threshold-shaped rather than linear, and a stratification explains why: recovery concentrates on items where the semantically bound supporting paragraph sits downstream of gold (Mistral 2Wiki: +2.5 pp, 18:7, p=.043, vs. −0.6 pp when every supporting paragraph precedes it; Qwen directionally the same). The trace is not a diffuse positional field—it pays when it lands on tokens that are about the dropped content. This is also exactly why keeping later context works as a policy (§7.1).

5.3 Does the trace store content, or just prime retrieval?

The sharpest objection to "the cache remembers what you deleted": perhaps the trace merely nudges the model toward facts it already knows parametrically—Vatican City's 1929 is in the pretraining data, after all. The probe: rewrite the gold paragraph's answer year to a fabricated one before the prebake (eligibility-filtered so the year appears nowhere else in the context), then drop the paragraph and ask.

Three controls anchor the readout (2Wiki, n = 461). With the counterfactual paragraph visible, the model echoes the fabricated year 93% of the time—it trusts context over

parameters—and the two arms are exactly identical (the identity gate holds for this pipeline too). With the paragraph dropped: the cached arm reproduces the fabricated year where fresh never does—6:0 one-sided flips at maximal trace (exact p=.031), dose-consistent (2:0 at natural position)—while the true-year channel shows no significant difference (27:23). The priming story loses; the content story wins.

The texture of the transport is telling: the cache often reproduces the fabricated year correctly but errs on the day or month ("April 15, 1946" where the document said "November 10, 1946")—a partial-fidelity imprint, with net verbatim bandwidth around 1–2% per readable item (a year-token lower bound). MuSiQue's null on this probe is exactly what that bandwidth, multiplied by its much lower task ceiling, predicts: the probe's sensitivity is ceiling × bandwidth. One transcript even cites the ghost—"this is supported by the text provided"—of a paragraph that no longer exists.

5.4 Characterizing the trace

Two further measurements sharpen the picture:

- The readout is early. In the cross-modal setting, swapping only the first 8 of 48 layers to the cached state recovers 68% of the full gap, with a plateau by layer 24. The trace is consumed in early-to-mid layers—consistent with contextual binding, not a deep-layer answer cache.
- The trace is a soft prior, not verbatim storage. Reading 91 recovery transcripts, the dominant mode is disambiguation: fresh wavers between two same-named entities ("…wait, no, that's a mix-up…") and picks the wrong one; reuse tips to the right one and proceeds. The model voices the recovered fact as its own world knowledge—it cannot cite a paragraph that is not there. In the gain population the recovered facts are correct; this is not a hallucination channel. The effect size (+2–4 pp, never a verbatim dump) matches: a lossy, associative imprint.

6 Where it breaks: boundary conditions

A mechanism is only useful with its sign conditions. The full three-dataset accuracy matrix (uniform compression, $n \approx 800$ each):

- 2Wiki: reuse ≥ fresh at every coverage (+1.1–2.3 pp overall)—the clean win.
- MuSiQue: reuse ≥ fresh (+1.3–2.3 pp at cov30–50; the gold-dropped stratum is +3.2 pp, 27:9).
- HotpotQA: approximately neutral overall, with one fresh-favored cell: gold kept, 50% coverage, −1.8 pp (19:12—directional, p=.28). Case-level reading: distractor overanchoring. The trace amplifies whatever was salient at prebake—including a kept, topically adjacent distractor. In a representative transcript the gold actor's paragraph is removed and a same-titled 1991 film's paragraph is kept: fresh bridges parametrically to the right person; reuse confidently answers with the kept film's cast. The cell is gone by 70% coverage, and even on HotpotQA the drop-gold recovery stays positive.

The unifying statement: the trace amplifies what was salient in the full context. It helps when the dropped content is the signal the task needs and the task resists shortcuts (MuSiQue, 2Wiki); it costs a couple of points when kept distractors are strong, coverage is low, and the task is shortcut-prone (HotpotQA 2-hop). The sign is (need for the dropped evidence) × (distractor adjacency); it is not a hop-count story.

The standing scope condition: all of this requires the unified-context prebake. In vanilla corpus-RAG, where chunks are baked independently, there is no full-context prefill, hence no trace, hence no asymmetry—we make no accuracy claim for generic retrieval-RAG. The regime that does qualify is common: multi-query sessions over one long document, streaming eviction, and agentic loops that repeatedly subset one long history.

Table 3: Keep-early vs. keep-late at a fixed budget. Gaps are reuse − fresh within each arm.

	keep-early gap	keep-late gap	diff-in-diff
2Wiki cov30	−0.9 pp	**+2.3** pp (41:23, p=.03)	+3.2 pp
2Wiki cov50	+0.7 pp	**+4.6** pp (59:22, $p<10^{-4}$)	+3.9 pp
HotpotQA cov30	−0.1 pp	**+2.0** pp (46:30, p=.085)	+2.1 pp
HotpotQA cov50	−0.5 pp	+0.4 pp	+0.9 pp
MuSiQue cov30/50	+0.9 / +0.2 pp	+0.6 / +1.0 pp	≈ 0

7 From explanation to method: two training-free policies

A mechanism that can be stated positionally can be acted on positionally. Both methods below are zero-training, and the second is uniquely cheap because of reuse.

7.1 Keep later context at a fixed budget

A direct corollary of causality: later tokens carry the trace of earlier dropped content; earlier tokens carry nothing of later content. At a fixed KV budget, therefore, prefer the last k paragraphs over the first k. Controlled test (same budget, same questions, $n = 800$; Table 3).

Fresh accuracy itself is unchanged early-vs-late (.573 vs. .576)—so this is not "late content is more useful"; the benefit is cache-specific, exactly as the mechanism predicts. The discordant-pair signature seals it: under keep-early, reuse and fresh disagree on almost nothing (19/18 items of 800); under keep-late they diverge (64/81). The trace in late-kept tokens is the difference between the arms. Implementation cost: one line in an eviction policy. Scoreboard, honestly: strong on 2Wiki, weak-positive on HotpotQA at the tighter budget, ≈ 0 on 20-paragraph MuSiQue—keep-late never hurts (it is the keep-early arm that goes negative) and pays where the trace is strong, i.e. where the kept tail can attend to most of what was dropped—the same bridge-routing condition §5.2 established.

7.2 Degeneration-gated adaptive coverage

The cliff (§3) is the one regime where reuse loses—and it is detectable from the cache side alone, with no fresh control: abstention phrasing, n-gram repetition, and missing-EOS truncation in the low-coverage generation predict wrong answers everywhere (MuSiQue cov30: $P(\text{wrong} \mid \text{signal}) = .80$ vs. .57 without). Policy: answer at low coverage; escalate coverage only while a signal fires.

Simulated on the real sweep generations (a proof of concept, stated as such): the gate matches the accuracy of the best fixed coverage with 9–22% less KV on all three datasets, and on 2Wiki it strictly dominates fixed cov50 (+1.5 pp accuracy and −9% budget). The oracle ceiling is striking—.922 at 42% of the full budget, versus .865 for always-full—so the signal design has headroom.

Why this is a reuse method and not a generic trick: keep-sets are nested, and the stored full-document cache already contains every row. Escalation is therefore fetch more rows and re-decode—no re-prefill. Under fresh computation, every escalation is a full re-prefill of a longer context. Adaptive coverage is uniquely cheap in exactly the regime this paper is about.

8 Discussion: the rolling agent context

The deployment regime where these results matter most is not one bake + many queries—it is the rolling agent context: user–agent and agent–agent turns accumulate, the window fills, and something must go. Today's frameworks truncate the oldest turns and re-prefill the survivors, or selectively evict tokens by importance. Our results reorder those options.

Drop-oldest is the principled policy, and slicing beats re-prefilling. Keep-late is "truncate the oldest turns"—except one keeps the surviving KV rows instead of recomputing them from the surviving text. Those rows were baked while the dropped turns were still present, so they carry the absorbed trace; re-prefilling from text is the one way to actually lose it. The usual correctness instinct—"the context changed, re-prefill to be safe"—selects the arm that is both slower and worse.

Selective mid-context eviction needs a query that does not exist yet. Importance scoring (H2O, SnapKV) is query-driven; in an agent loop the future queries are unknown, and in our video-side decomposition query-free importance selection added approximately nothing. With no query, recency is the only signal available—and it happens to be the trace-optimal one. (This also supplies a mechanistic justification for StreamingLLM's recency window: beyond keeping decoding stable, the window is the trace-richest subset shape.)

Summarize-then-evict is an engineered trace. Generate the summary while the old turns are still in context, append it, then drop them: the summary tokens' KV absorbs the full context at bake time, exactly like any other downstream token. The natural trace and the summary are then the same kind of object at different bandwidths—the free channel carries gist and binding at ∼1–2% verbatim bandwidth (§5.3); the summary carries whatever one spends decode tokens on. The bandwidth measurement doubles as a specification for what the summary must contain: verbatim detail and long-chain structure (what the free channel drops), not gist (what it already keeps).

Limitations and next steps. Everything in this paper is measured on a single bake and a single slice. A rolling context is bake → evict → extend → evict again, and whether absorption compounds or decays across cycles is unmeasured—that is the experiment we run next, alongside trace-aware eviction: a chunk whose information has already been absorbed by surviving downstream tokens is safe to evict regardless of its importance. An absorption/redundancy criterion is computable from prebake attention; we state it as the open direction and will benchmark it in follow-up work. Further limitations: results cover two text model families and one omni-modal model at the ∼24–30B scale; the degeneration gate is a simulation on persisted generations, not a serving-system implementation; and the counterfactual probe establishes existence and a bandwidth estimate, not a full account of what the trace encodes.

9 Related work

We place the closest lines of work last, because none of them measures the axis this paper is about—the conditional, mechanistic quality of a same-context cache reused on a subset—and the contrast is clearest after the results.

Reuse for latency. TurboRAG (Lu et al., 2024), prefix caching (Kwon et al., 2023), and LMCache/CacheGen (Liu et al., 2024b) treat cache reuse as a time-to-first-token problem and aim for quality neutrality. The quality dimension we map is invisible in that framing.

Cache fusion. The cache-fusion line—CacheBlend (Yao et al., 2025) and, most recently, RelayCaching (Geng et al., 2026), which reuses one agent's decode-phase KV in the next agent's prefill—runs in the opposite direction from us: there, the KV was computed under a different or missing context than the one it is used in, so the absorbed-prefix component is contamination, and the methods spend selective recomputation to remove it. Our setting is the mirror image: the cache is baked in the same full context the query is about, and the absorbed component is not a cost to patch but the asset itself—it is precisely what makes the reused cache better than recomputation. One sentence resolves the apparent contradiction: whether KV deviation from recompute is an error or a memory depends on whether the bake-time context is the context the query wants to condition on. Notably, RelayCaching's own measurements independently confirm the physics we exploit: the deviation (the absorbed context) is sparse, concentrated on few token positions, and layer-wise non-uniform—and

one of its two selection criteria scores tokens by attention received from subsequent positions (their "influence-based selection"), the same downstream-attention quantity our causal experiments manipulate.

Learned compression and retention. Gist tokens (Mu et al., 2023), ICAE (Ge et al., 2024), and AutoCompressors (Chevalier et al., 2023) buy smaller caches with training; DuoAttention (Xiao et al., 2025) and Make Each Token Count (Bui et al., 2026) learn what to retain, the latter showing learnable eviction can surpass the full cache. The coverage curve says a large fraction of that quality is available with no training at all, and the paired reuse-vs-fresh design (same kept tokens, only the KV source differs) isolates why—a trained compressor inherits, rather than explains, the trace.

Eviction and streaming. H2O (Zhang et al., 2023) and SnapKV (Li et al., 2024) score which tokens to keep by attention importance; StreamingLLM (Xiao et al., 2024b) keeps attention sinks plus a recency window and shows decoding remains stable under unbounded streams. StreamingLLM in fact operates inside our regime—its surviving window KV was computed while the since-evicted tokens were present—but asks a different question (stability of continued decoding) and never compares against recomputation of the kept text, which is unaffordable in a stream. We quantify the value of exactly that comparison, and our keep-late result (§7.1) supplies a mechanistic justification for the recency-window design itself. Absorption—whether a token's information already survives in downstream kept rows—is an axis none of the importance-based methods measure (§8).

Block retrieval and position conventions. InfLLM (Xiao et al., 2024a) and ReKV (Di et al., 2025) retrieve cached blocks and re-rotate positions to be contiguous. Our results are compatible: the position-compaction convention is harmless (a faithful ReKV reimplementation ties plain reuse; Appendix B), and their retrieved blocks carry the same memory we characterize.

Multimodal KV management. Multimodal eviction work (e.g. MEDA (Wan et al., 2025)) optimizes cross-modal cache allocation for efficiency. None tests modality-absent recovery—whether a deleted modality's information survives in the other modality's KV (§4). To our knowledge (June 2026), no prior work occupies the combination presented here: the conditional quality map, the causal mechanism, the cross-modal recovery demonstration, and mechanism-derived training-free policies.

10 Conclusion

A KV cache sliced from a full-context prefill is not a degraded approximation of recomputation; it is a different object that remembers part of what was deleted. The quality consequences are governed by coverage, explained by a causally established downstream-attention trace, bounded by salience amplification, and exploitable by training-free policies—keep later context, gate coverage on degeneration, and, in rolling agent contexts, slice rather than re-prefill. For practitioners the summary is one sentence: when the keep-set changes, slicing the existing cache is both cheaper and better than re-prefilling—except below the coverage cliff, which is detectable at runtime.

Reproducibility. Models: Qwen3-30B-A3B-Instruct-2507 (text), Mistral-Small-24B-Instruct-2501 (cross-family replication; family-adapted prompt conventions, same instrument and seeds), Qwen3-Omni-30B-A3B Thinker (video, M-RoPE)—stock checkpoints, no fine-tuning. Datasets: LongBench 2WikiMQA/HotpotQA/MuSiQue (Bai et al., 2024; Trivedi et al., 2022; Yang et al., 2018; Ho et al., 2020); original MuSiQue ans-v1.0 dev, HotpotQA dev-distractor, 2WikiMQA dev; EgoSchema-Subset (Mangalam et al., 2023); Video-MME (Fu et al., 2025). Gold position is manipulated over {first, random-interior, last} with downstream counts recorded; the counterfactual probe uses eligibility-filtered year-answer items ($n = 461/226$) scored by word-bounded year-token match for both the fabricated and the true year; NLL sweeps use $n = 231$–1500; every run carries the cov100

identity gate; bf16 with an fp32 control. Per-claim evidence tables and the experiment scripts accompany this paper in the project repository.

A Experimental protocol

The identity gate. Every run carries a built-in falsification check: at 100% coverage nothing is dropped, the two arms are the same computation, and the measured gap must vanish. It does—exactly, to the last item, in every accuracy experiment (e.g. .560/.560/.560 on MuSiQue), and within noise on the NLL curves ($+0.02 \pm 0.01$). This is what separates a mechanism from an instrumentation artifact: every difference reported in the main text appears only when something is deleted, and vanishes exactly when nothing is.

Pairing and significance. All accuracy comparisons are paired (same items, greedy decoding, alias-match scoring, persisted generations); significance is the exact McNemar test (two-sided binomial on the discordant pairs), reported as flip counts a:b (items reuse fixes vs. items reuse breaks). Text cells use $n = 796$–800 ($n = 2400$ for the MuSiQue gold-position cells); video uses the paired Wilcoxon signed-rank test, $n = 236$–597.

Verbosity control. The accuracy gaps are not a length artifact of alias-match scoring: reuse generations are the same length or slightly shorter than fresh in every dataset×coverage cell (-3 to $+1$ words on average), and scoring is paired item-by-item.

B What we ruled out

Negative results that constrain the interpretation (Table 4).

Table 4: Competing explanations and their status.

Competing story	Verdict
Position-preserving reuse beats compaction	Null. Re-rotating kept rows costs +0.006 NLL, n.s., even under maximal M-RoPE shear. The win is the memory, not the position convention.
This beats ReKV/InfLLM-style systems	No. A faithful ReKV reimplementation ties plain reuse. We claim the mechanism and the policies, not a systems win.
Cache $\geq$ fresh is universal	No. The text low-coverage cliff is real, convention-independent, and is starvation.
The benefit/penalty tracks hop count	No. Sign tracks (need for dropped evidence) × (distractor adjacency); HotpotQA flips to +4.1 pp when the trace is maximized.
The cross-modal trace lives in deep layers	No. Early–mid readout: the first 8 layers carry 68% of the gap.
Gold-first wins by primacy / attention sink / lost-in-the-middle	No. Random interior slots 1–2 recover as much as slot 0 (+4.9 pp, both families); the predictor is downstream mass, concentrated where the supporting paragraph is downstream.
The trace just primes parametric retrieval	No. Counterfactual probe: fabricated-year reproduction 6:0 (p=.031) while the true-year channel is null.
It is a Qwen quirk	No. Curve and causal A/B replicate on Mistral-Small-24B with larger effects.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL), 2024.

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.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2023.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoran Li, Tao Zhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video KV-cache retrieval. In International Conference on Learning Representations (ICLR), 2025.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2025.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In International Conference on Learning Representations (ICLR), 2024.

Yingsheng Geng, Yuchong Gao, Weihong Wu, Guyue Liu, and Jiang Liu. RelayCaching: Accelerating LLM collaboration via decoding KV cache reuse. arXiv preprint arXiv:2603.13289, 2026.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Proceedings of the 28th International Conference on Computational Linguistics (COLING), 2020.

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 29th Symposium on Operating Systems Principles (SOSP), 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 (NeurIPS), 2024.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics (TACL), 12:157–173, 2024a.

Yuhan Liu, Hanchen Li, Yihua Cheng, Siddhant Ray, Yuyang Huang, Qizheng Zhang, Kuntai Du, Jiayi Yao, Shan Lu, Ganesh Ananthanarayanan, et al. CacheGen: KV cache compression and streaming for fast large language model serving. In Proceedings of the ACM SIGCOMM 2024 Conference, 2024b.

Songshuo Lu, Hua Wang, Yutian Rong, Zhi Chen, and Yaohua Tang. TurboRAG: Accelerating retrieval-augmented generation with precomputed KV caches for chunked text. arXiv preprint arXiv:2410.07590, 2024.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In Advances in Neural Information Processing Systems (NeurIPS), 2023.

Jesse Mu, Xiang Lisa Li, and Noah D. Goodman. Learning to compress prompts with gist tokens. In Advances in Neural Information Processing Systems (NeurIPS), 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 (TACL), 10:539–554, 2022.

Zhongwei Wan, Hui Shen, Xin Wang, Che Liu, Zheda Mai, and Mi Zhang. MEDA: Dynamic KV cache allocation for efficient multimodal long-context inference. In Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), 2025.

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

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

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2018.

Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. CacheBlend: Fast large language model serving for RAG with cached knowledge fusion. In Proceedings of the Twentieth European Conference on Computer Systems (EuroSys), 2025. arXiv:2405.16444.

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 (NeurIPS), 2023.