WHEN TO EVICT, NOT WHAT TO KEEP: DRAFT-GUIDED EVICTION FOR TRAINING-FREE KV-CACHE COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

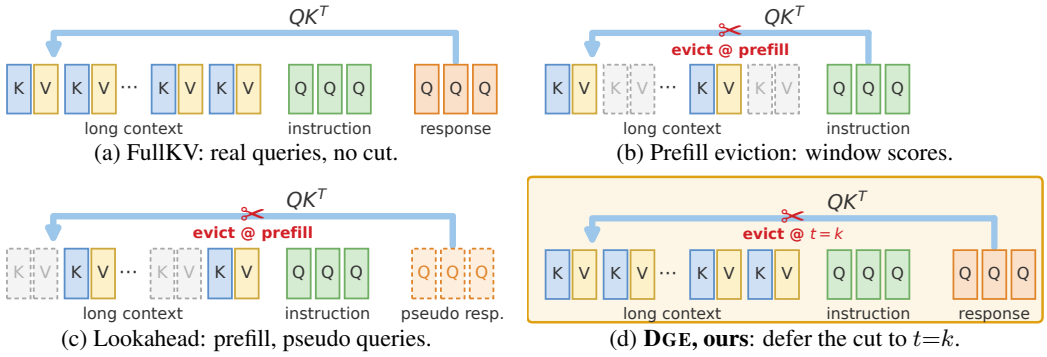
Training-free KV-cache compression methods such as SnapKV, H2O, and PyramidKV evict tokens at the end of prefill, aiming to preserve the attention mass that future queries are expected to use—optimizing *what* to keep. We show that this objective fails in two distinct ways. **(1) Compensation:** restoring the evicted attention mass can recover the attention-level target without recovering task quality. **(2) Selection:** covering more of the true decode-query mass can *hurt* quality when the recovered mass is fragmented rather than concentrated in coherent spans. These failures share a common cause: eviction occurs before the queries that determine the answer trajectory exist. We propose **D**raft-**G**uided **E**viction (DGE), which defers eviction until after drafting the first $k{=}2$ answer tokens using the full cache—just one decode step beyond prefill. Because the draft is generated from the answer's own prefix, no cache entries are discarded before this trajectory signal becomes available. The per-head cache budget remains unchanged, and DGE can be applied directly to SnapKV, PyramidKV, H2O, and StreamingLLM without modifying their eviction scores. Unlike extra-pass methods, DGE changes *when* eviction occurs rather than *what* cache entries are selected. Extensive experiments demonstrate that DGE outperforms prior methods at every evaluated budget on five of six instruct-tuned backbones, achieving 44.2 on LongBench, nearly matching FullKV at 44.3. The timing-only control DGE-W achieves the same score, demonstrating that the gain comes from *when* eviction occurs rather than *what* is selected—an effect we term *trajectory anchoring*.

1 INTRODUCTION

Serving long-context LLMs is a memory problem before it is a compute problem: the KV-Cache grows linearly with context length and batch size—roughly 16 GiB for an 8B model at 128K tokens, on par with the weights themselves—and every decode step reads the cache back. Training-free eviction is an attractive deployment strategy: keep a small per-head budget of B past tokens, discard the rest, and change nothing else about the model. A productive line of work has refined *which* tokens to keep—StreamingLLM keeps attention sinks and recency (Xiao et al., 2024), H2O accumulates attention mass (Zhang et al., 2023), SnapKV max-pools the scores of a late-prompt observation window (Li et al., 2024), and PyramidKV schedules the budget across layers (Cai et al., 2024)—but these methods share one recipe: when prefill ends, score the past tokens, keep the top-B per head, and evict the rest, permanently and *before the first answer token is generated*.

Two choices distinguish these training-free regimes: *which* queries score the cache, and *when* the cut fires. FullKV in Figure 1(a) uses the real decode queries and never evicts; prefill-time eviction in Figure 1(b) cuts before generation using only prefill-time scores; Lookahead Q-Cache (LAQ) in Figure 1(c) also cuts at prefill, but replaces future decode queries with synthesized *pseudo*-queries (Wang et al., 2025). Our **D**raft-**G**uided **E**viction (DGE), in Figure 1(d), instead drafts the first k answer tokens on the full cache and evicts only afterward. It therefore observes the *real* decode queries before making the irreversible cut: DGE changes *when* the cut happens, rather than *what* scores the cache.

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.



(a) FullKV: real queries, no cut. (b) Prefill eviction: window scores.

(c) Lookahead: prefill, pseudo queries. (d) **DGE, ours**: defer the cut to $t{=}k$.

Figure 1: **Four training-free KV-compression regimes:** which queries drive the $QK^\top$ scoring, and when eviction fires, marked by red scissors. Only DGE (d) defers the cut; the window-score regime (b) covers SnapKV, PyramidKV and their descendants.

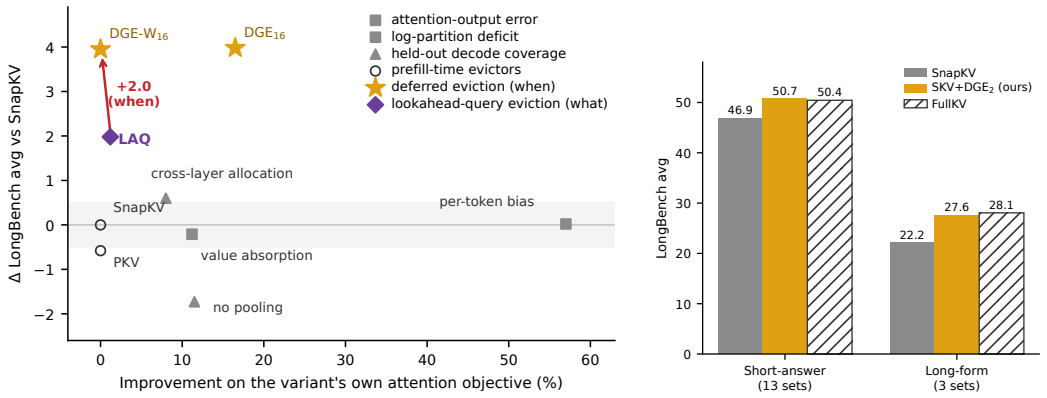
Despite these differences, existing approaches optimize a common implicit objective: preserve the attention mass that future computation would place on the discarded tokens. *Selection* maximizes captured mass, as in SnapKV and H2O (Li et al., 2024; Zhang et al., 2023); *compensation* attempts to restore what selection loses through merging entries or fitting biases and values (Zhang et al., 2024b; Wan et al., 2024b; Zweiger et al., 2026); and *allocation* redistributes the budget across heads and layers (Feng et al., 2024; Cai et al., 2024). This objective is plausible, and its attention-level gains are real. However, whether increasing attention-mass preservation reliably translates into task quality has not been systematically examined under controlled dosage.

Our controlled observations reveal two limitations. **(1) Compensation:** mass is fungible, but content is not. Restoring evicted attention mass through merging, value absorption, or per-token correction does not necessarily recover the information carried by the original tokens. **(2) Selection:** more captured attention mass does not necessarily improve task performance. When retained mass is fragmented rather than concentrated in coherent spans, an extractive answer can lose the information needed for reconstruction. We establish both effects under controlled dosage in §4, with the corresponding sweeps in Figure 2(a). Both limitations arise because eviction occurs before the queries that actually consume the cache exist: prefill-time signals can only estimate future decode queries, however they are constructed.

We propose DGE, which changes *when* eviction occurs rather than *what* is kept. Prior methods improve *what* a prefill-time cut keeps, while leaving the cut at the end of prefill (Kim et al., 2025; Zweiger et al., 2026; Wang et al., 2025). DGE instead drafts the first k answer tokens on the full cache and evicts only afterward, allowing the decision to use real decode queries rather than estimates. The draft is the answer's own prefix, not an auxiliary pass, so the per-head budget is unchanged and eviction leaves the prefill critical path. A single decode step is often sufficient, and §6 shows that the deferral itself, rather than a sharper selection signal, drives the gain. We attribute this to what we call *trajectory anchoring*: the initial answer tokens, produced on the full cache, fix the model's decoding trajectory before compression, and the compressed cache then carries the rest, as Figure 2(b) shows.

Contributions.

- **Draft-guided eviction.** We introduce DGE, a training-free KV-cache compression framework that opens a new *when*-to-evict axis: the model first drafts k answer tokens with the full cache and evicts only afterward, keeping the per-head budget and leaving the scoring rule a free slot—the base evictor's own (the timing-only control DGE-W) or the draft's real queries (the default).
- **Controlled analysis of attention-based eviction.** We identify two failure modes of optimizing attention mass—*compensation* and *selection*—and show that attention-level preservation can decouple from downstream task quality under controlled dosage.
- **Empirical validation.** Across six instruct-tuned backbones from 3B to 14B parameters, DGE outperforms prior methods at every evaluated budget on five backbones and achieves an average LongBench score of 44.2, compared with 40.1 for LAQ and 35.3 for SnapKV, within 0.2 of uncompressed FullKV. The same deferral



(a) Moving *when* beats improving *what*.

(b) One decode step restores FullKV.

Figure 2: **When beats what**, on LongBench with Llama-3.1-8B at B=128; subscripts are the draft length k. **(a)** Each *what*-axis mechanism's gain on its own attention objective against its LongBench Δ vs SnapKV; gray band: ± 0.5; hollow circles: SnapKV and PKV; LAQ (Wang et al., 2025) at m=16. Both deferred markers defer the cut by 16 tokens, DGE-W$_{16}$ scoring with SnapKV's window and DGE$_{16}$ with the 16 draft queries; the red arrow is deferral's gain over LAQ. **(b)** Short-answer (13 sets) vs long-form (the three summarization sets) averages.

lifts PyramidKV, H2O, and StreamingLLM, while DGE-W also reaches 44.2, isolating eviction timing as the source of the gain.

2 Related Work

Selection and allocation: which tokens to keep. The dominant training-free family scores past tokens at the end of prefill under a per-head budget. SnapKV (Li et al., 2024) max-pools observation-window scores; H2O (Zhang et al., 2023) and ScissorHands (Liu et al., 2023) accumulate attention; FastGen (Ge et al., 2024) adapts budgets per head; PyramidKV (Cai et al., 2024) schedules them across layers; AdaKV (Feng et al., 2024) and HeadKV (Fu et al., 2024) reallocate them across heads; and StreamingLLM (Xiao et al., 2024), ThinK (Xu et al., 2024), L_2 selection (Devoto et al., 2024), and Quest (Tang et al., 2024) use sinks, recency, or key statistics. These methods improve *which* tokens survive, but commit before the first answer token exists. DGE leaves the per-head budget unchanged and can leave the scoring rule unchanged too, deferring the same selection decision until after the initial answer tokens. On SnapKV's own scores, this timing change alone already outperforms the selection rules we evaluate.

Compensation: restoring what selection discarded. A second line of work attempts to repair the cut rather than improve selection. CaM (Zhang et al., 2024b), LOOK-M (Wan et al., 2024b), KVMerger (Wang et al., 2024), and D2O (Wan et al., 2024a) merge evicted entries into retained ones, with the vision analogue ToMe (Bolya et al., 2023); FAST (Zweiger et al., 2026) instead fits per-token biases and refits values. These methods can restore the attention mass targeted by the repair, but they cannot recover the original token-level content once it has been compressed away. Our controlled counterparts in §4 expose this gap: attention-level compensation can improve without a corresponding gain in task quality. DGE avoids the repair problem altogether by generating the steering tokens while the cache is still intact, so the information needed to anchor the answer trajectory is never discarded.

Extra passes: sharpening the prefill-time signal. A third line spends additional computation to improve the eviction signal. KVzip (Kim et al., 2025) repeats the prefill pass; FAST, above, calibrates its biases and values on self-generated Q&A; LAQ (Wang et al., 2025) synthesizes *pseudo* future queries; and LookaheadKV (Ahn et al., 2026) trains adapters to predict such scores. These methods improve *what* to keep, but retain the same schedule: the cut still fires at prefill end, so the signal remains a proxy for the queries that actually consume the cache. Our experiments show that a matched

LAQ run saturates with dose and, even at its best, trails a pure timing change. Other KV-efficiency methods change the representation, memory management, or decoding objective, including learned compression (Zhang et al., 2024a; Nawrot et al., 2024; Mu et al., 2023), quantization (Liu et al., 2024b; Hooper et al., 2024), PagedAttention (Kwon et al., 2023), InfiniGen (Lee et al., 2024), and speculative decoding (Leviathan et al., 2023); concurrent analyses also question the attention-mass objective itself (Feng et al., 2025; Yang et al., 2026; Zhang et al., 2026). In contrast, DGE changes only the *timing of eviction*, using the answer's own prefix to expose real decode queries before the cut; App. F contrasts it with the draft-like designs.

3 PRELIMINARIES

Notation. A prompt of length L is prefilled; eviction keeps a per-head budget of B tokens ($B{-}w$ selected past tokens plus the last w window tokens, $w{=}8$ throughout) per layer. For a future query q, the full attention output decomposes exactly as

$$O(q) \;=\; (1-\beta(q))\, O_{\text{keep}}(q) \;+\; \beta(q)\, V_{\text{evict}}(q), \qquad \beta(q) = \textstyle\sum_{j\in E} a_j(q), \tag{1}$$

where E is the evicted set: β is the *missed mass* and V_{evict} the attention-weighted centroid of evicted values. Every compensation method estimates terms of Eq. 1; every selection method minimizes β.

Held-out instrumentation. Our diagnostics run the model once with the full cache, capture post-RoPE Q, K, V via forward pre-hooks, and replay *real decode queries* against candidate caches offline. Metrics, per App. B and H: missed decode mass β_{dec} is the β of Eq. 1 on a held-out half of the decode steps, and a rule's *held-out coverage* Δ is its relative reduction in β_{dec} vs SnapKV. Also relative attention-output error, retained-mask run length, and a train-free attention-matching loss $|\log Z - \log \hat{Z}|$, the log-partition deficit between the softmax normalizer Z over all keys and $\hat{Z}$ over the retained ones.

Evaluation protocol. LongBench (Bai et al., 2024) (16 English datasets) on Llama-3.1-8B-Instruct (Dubey et al., 2024), budgets $B \in \{32, 64, 128\}$ (larger only in the offline diagnostics), greedy decoding, the stop-token-corrected Llama-3 chat protocol of App. A; matched sample sets across methods. Backbone checks span Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), Qwen2.5-7B/14B-Instruct (Qwen Team, 2024), Llama-3.2-3B-Instruct, and Llama-3-8B-Instruct, reported in Table 2 and App. D. Reruns are bit-identical, so the only uncertainty is sampling over the 3,750 examples; comparisons use a paired bootstrap, whose intervals on differences are $\sim \pm 0.3$ against ± 1.1 marginally, as App. I details.

Headline diagnostic. At a per-head budget of $B{=}128$ tokens, SnapKV misses $\beta_{\text{dec}} = 16.7\%$ of decode attention mass on average, with a median of 9.4% and a 90th percentile of 43% over examples, layers and heads, a heavy tail; the window-query β correlates 0.72 with β_{dec} (0.68–0.76 over $B{=}64$–1024)—the missed mass is apparent headroom that §4's analysis shows is *not* convertible into task quality by content-side means.

4 ATTENTION-MASS ANALYSIS

We test the attention-mass objective under increasing dosage on each axis (Llama-3.1-8B, $B{=}128$ unless a budget is named). Figure 2(a) plots each mechanism's attention-level metric against its LongBench Δ vs SnapKV: the four *what*-axis mechanisms gain 8–57% on their objective yet at most $+0.6$ on LongBench, LAQ's 16 pseudo queries buy $+2.0$, and deferring the cut by the same 16 tokens adds a further $+2.0$—both DGE points near $2\times$ the best prior. Every family meets its objective and none of it transfers (Table 1(a) *fit*, (b) *select*).

Compensation restores mass, not content. Table 1(a) is ours throughout: compensation token and multi-centroid following LOOK-M/KVMerger (Wan et al., 2024b; Wang et al., 2024), value absorption following CaM (Zhang et al., 2024b), per-token bias following FAST (Zweiger et al., 2026). Mechanistically (App. C), renormalized mass flows *multiplicatively* (inflow corr. 0.97 with a token's own attention, so merging by key similarity cannot route it), post-RoPE key averaging loses 30% of key norm to phase interference, and a per-token least-squares fit overfits the eight window queries a prefill affords.

Table 1: **The attention-mass objective is met and the score does not follow** (Llama-3.1-8B, B=128; LongBench (LB) Δ vs SnapKV; ↑: higher is better; full grids in App. C). Attention-level: each mechanism's gain on its own objective (mass recovered, error or deficit reduced); Cov. Δ: relative gain in held-out coverage over SnapKV; Run len.: mean contiguous run of retained tokens. All rows are ours except SnapKV (Li et al., 2024).

(a) Compensation (*fit*)

Mechanism	Attention-level ↑	LB Δ ↑
Compensation token	+8.4% mass	−0.03
Multi-centroid	+35.7% mass	−0.48
Value absorption	+11.2% error	−0.21
Per-token bias	+57.0% deficit	+0.02

(b) Selection (*select*)

Rule	Cov. Δ ↑	Run len.	LB Δ ↑
SnapKV	–	10.4	–
No pooling	+11.5%	1.8	−1.73
Decode queries (DGE_{16})	+16.5%	1.6	+3.98
Token oracle	+31.8%	1.6	(bound)

Selection trades coverage for contiguity. SnapKV's max-pooling *loses* 9–22% held-out coverage across B=64–512 yet *wins* scores by buying span contiguity, and raw "no pooling" inverts both at B=128 in Table 1(b). Oracle bounds cap every practical rule (token oracle +31.8%; block and segment oracles in App. C). The only signal materially above the window is *real decode queries*: +16.5% in Table 1(b) (DGE_{16}'s k=16 draft queries, matching LAQ's m=16), +30.4% at B=512, against +1.2% for lookahead's pseudo queries in App. L. It is also the only entry in Table 1 whose gain reaches the score: with runs as short as no pooling's (1.6 vs 1.8) it gains +3.98 where no pooling loses 1.73—because it is a cut deferred until the real queries exist (§5), not a better prefill-time rule: the timing-only control, keeping SnapKV's long runs, gains the same (+3.95, DGE-W_{16}). Figure 2(b) cashes it out: SnapKV collapses on long-form generation, and SKV+DGE_2 recovers FullKV in both regimes.

5 Draft-Guided Eviction

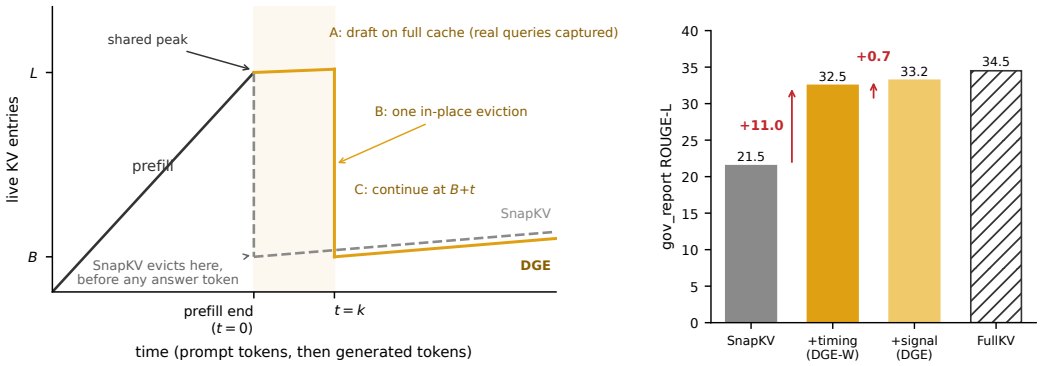
The two problems of §4 share a cause: the cut fires before the queries that matter exist. DGE removes the cause rather than either symptom by making the real decode queries exist *before* the cut, and both problems then dissolve: nothing is left to compensate, and the coverage–contiguity trade-off is an artifact of pre-generation scoring—once the queries reading the cache are the real ones, the spans an extractive answer copies have already been read.

Policy. DGE is a scheduling change, not a new scorer: it defers eviction past the opening of the generation, as App. E sets out:

1. **Draft (phase A).** Prefill normally; generate the first k answer tokens with the *full* cache, capturing the k real decode queries per layer through forward pre-hooks—no extra pass, the draft being the beginning of the final answer. Prefill's own forward emits token one, so k=1 is the base evictor and k=2 the minimal dose, used throughout.
2. **Evict (phase B).** Score every past token, keep the per-head budget, and edit the cache in place—once, irrevocably. The scorer is a free slot. By default DGE uses the mean attention the k draft queries pay to each token, which the draft provides at no cost; the timing-only DGE-W keeps the base evictor's own scorer—SnapKV's window, PyramidKV's layer budgets, H2O's accumulator or StreamingLLM's sinks—unchanged.
3. **Continue.** Resume on the compressed cache, keeping the draft tokens as ordinary output.

§6 finds the two scorers equivalent on average: the contribution is *when* the cut fires. The change is correspondingly small to deploy: two stock `generate()` calls around one in-place cache edit, with no training, no extra pass, and no hyper-parameter beyond k (App. E).

Cost. DGE leaves the asymptotic peak unchanged, as Figure 3(a) sketches, and the dose itself is free: total latency is flat in k. The cost is one cache surgery, 1.45× on short answers and 1.02× on long in Table 3(b), against a TTFT of 0.93× SnapKV's that no score-based prior evictor there beats. If the answer terminates within the draft, no eviction occurs and the output is bit-identical to FullKV's—the rational policy, since the cache is freed at answer end.



(a) Same steady state, later eviction.

(b) Timing carries the recovery.

Figure 3: **DGE overview and gain attribution. (a)** Live KV entries over time: one shared prefill ramp to L, the same $B{+}t$ steady state, and the single eviction firing at prefill end for SnapKV but after a k-token draft for DGE, which holds the full cache for those k steps. **(b)** gov_report ROUGE-L, Llama-3.1-8B, $B{=}128$, matched 50-sample subset; FullKV reaches 34.5.

A hypothesis for why DGE works: trajectory anchoring. The first k tokens are the generation's high-information steering segment: for extractive QA they contain the copied span; for summarization they fix topic and structure. The draft segment is also precisely where prefill-time eviction bites. Produced from the whole cache, the draft anchors a trajectory a compressed cache then carries—consistent with the timing-only DGE-W matching the 44.2 average of DGE across Table 2 and carrying nearly all of the recovery in Figure 3(b), as §6 attributes.

6 EXPERIMENTS

§4 closed the *what* axis; this section measures what §5's deferral buys on the *when* axis: a lead over every prior method in every backbone × budget cell, FullKV parity within noise, and, through a timing-only control, the deferral itself as the cause.

6.1 EXPERIMENTAL SETUP

Backbone LLMs. Llama-3.1-8B-Instruct (Dubey et al., 2024) is the primary backbone, with replications on Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), Qwen2.5-7B/14B-Instruct (Qwen Team, 2024) and Llama-3.2-3B in Table 2, and on Llama-3-8B in App. D; decoding and prompt protocol follow §3's preliminaries.

Datasets. We use LongBench (Bai et al., 2024), evaluating all 16 English datasets: single-document QA (NarrativeQA, Qasper, MultiFieldQA; Kočiský et al., 2018; Dasigi et al., 2021), multi-document QA (HotpotQA, 2WikiMultiHopQA, MuSiQue; Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022), summarization (GovReport, QMSum, MultiNews; Huang et al., 2021; Zhong et al., 2021; Fabbri et al., 2019), few-shot learning (TREC, TriviaQA, SAMSum; Li and Roth, 2002; Joshi et al., 2017; Gliwa et al., 2019), synthetic retrieval and counting, and code completion (LCC, RepoBench-P; Guo et al., 2023; Liu et al., 2024a). Inputs average 1,235–18,409 tokens; longer prompts are middle-truncated per the official protocol, and each set is scored by its official metric (F1, ROUGE-L, accuracy, edit similarity).

Baselines. All baselines keep the same per-layer, per-head KV budget B and differ only in their retention strategy; our reproductions at $B{=}128$ rank as published (SnapKV 42.2, PyramidKV 41.6, StreamingLLM 36.5):

- **FullKV (FKV)** caches every token and is the uncompressed reference.
- **StreamingLLM (SLM)** (Xiao et al., 2024) keeps attention sinks and a recency window only.

Table 2: **One decode step of deferral leads every prior method at every budget** (LongBench average over 16 sets; five backbones ordered by size; KV budgets $\{32, 64, 128\}$; FullKV is budget-independent; Avg. is the mean over all fifteen backbone $\times$ budget cells; **bold** is the best compression method per column, FullKV excluded as the uncompressed reference). Both deferred rows are ours: the shaded **bold** row is the default DGE, DGE-W_2 the timing-only control (same deferral, SnapKV's scores). LAQ (Wang et al., 2025) is at its best dose, m=16.

	Llama-3.2-3B			Mistral-7B-v0.2			Qwen2.5-7B			Llama-3.1-8B			Qwen2.5-14B			Avg.
Method	32	64	128	32	64	128	32	64	128	32	64	128	32	64	128	
FKV	42.45			41.35			45.54			46.24			46.05			44.33
SLM	28.14	29.30	30.86	24.58	25.68	27.10	27.11	28.90	30.19	33.44	35.14	36.51	28.63	30.40	31.82	29.85
PKV	33.07	33.50	37.68	27.87	31.83	34.91	32.69	30.74	38.38	36.54	39.58	41.64	35.07	34.64	40.41	35.24
SKV	31.57	35.64	38.07	26.37	31.35	34.53	31.56	36.77	39.98	35.85	39.38	42.22	29.94	36.42	40.51	35.34
SKV+LAQ	35.79	39.13	40.90	33.94	37.21	39.86	37.81	41.59	43.47	40.65	43.17	44.20	37.10	42.63	44.46	40.13
SKV+DGE-W_2	42.10	42.30	42.32	40.58	**41.10**	40.87	**45.17**	45.20	**45.36**	**46.40**	46.20	46.33	**46.18**	**46.31**	46.18	**44.17**
SKV+DGE$_2$ (Ours)	**42.32**	**42.36**	**42.56**	**40.72**	40.90	**41.00**	45.05	**45.32**	45.32	46.39	**46.36**	**46.37**	45.74	45.94	**46.27**	**44.17**

- **SnapKV (SKV)** (Li et al., 2024) scores past tokens with an observation window and max-pools before per-head top-B; it is the base of all our variants.
- **PyramidKV (PKV)** (Cai et al., 2024) adds a layer-decaying budget schedule.
- **Lookahead Q-Cache (LAQ)** (Wang et al., 2025), the closest prior extra-pass baseline, at its best dose (m=16) in Table 2, across doses in Table 3(a) and over the full grid in App. K.

Our variants. Each variant is prefixed with the base evictor it modifies and keeps its budget; the suffixes isolate the two axes of §5 (pseudocode in App. E):

- **SKV+DGE$_k$** (*eviction timing* + *signal*) is the default of §5: the cut moves to decode step k and is scored by the k real draft queries rather than the observation window. Budget and allocation stay SnapKV's, so any difference from the SKV row comes from those two changes alone.
- **SKV+DGE-W$_k$** (*timing only*) defers eviction identically but scores with SnapKV's unmodified window recipe, so DGE-W differs from the SKV row solely in the step at which eviction happens.

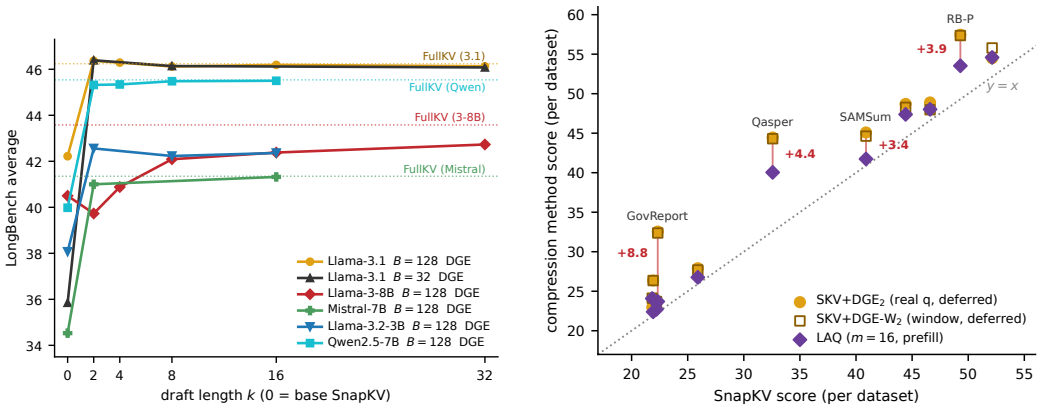
The deferral composes with the other retention rules the same way (PKV+/H2O+/SLM+DGE rows, App. K).

Unless noted otherwise: budget B=128 for single-budget results ($B \in \{32, 64, 128\}$ in Table 2), observation window w=8, pooling kernel 7, draft length k=2 (k up to 32 in the dose grid), matched sample sets across methods, one run per configuration (greedy; CIs in App. I).

6.2 Main Results

DGE leads every prior method at every budget, near FullKV. Averaged over all fifteen backbone $\times$ budget cells, as in the Avg. column of Table 2, SKV+DGE$_2$ scores 44.17 against 40.13 for the strongest prior method, SKV+LAQ at its best dose, and 35.34 for SnapKV; both DGE variants beat every prior method in all fifteen cells, by at least 1.01, and stay within 0.77 of FullKV—matching or exceeding it in five cells—where the strongest prior method trails FullKV by 1.5–9.0 (per-dataset scores: App. K, except Mistral-7B and Llama-3.2-3B at B=32). The deferral is essentially *budget-invariant*: averaged over backbones, as B shrinks from 128 to 32, SKV+DGE$_2$ moves only from 44.30 to 44.04, while SnapKV falls from 39.06 to 31.06, so DGE's lead over the best prior method grows from 1.1–2.2 at B=128 to 5.7–8.6 at B=32. On the sixth backbone, Llama-3-8B, a k=32 draft closes 72% of the SnapKV gap to FullKV at B=128, as App. D shows. The deferral also composes with other base evictors: PKV/H2O/SLM+DGE$_2$ land within 0.9 of FullKV in the three settings App. L reports, SLM gaining +12.3 on average (PKV's non-monotone Qwen columns: a discrete-schedule artifact, explained there too).

Where the gain lives. Draft termination (answers ending inside the draft) is rare at the headline dose—11.2% of samples at k=2, 0% on the long-generation categories, per App. K—so the gain must come from generation on the compressed cache. On gov_report, where ∼99% of tokens decode after eviction, SKV+DGE$_2$ lifts ROUGE-L from 22.3 to 32.6 against FullKV's 33.8 (Figure 4(b)); on



(a) Minimal dose is backbone-dependent.

(b) Timing rescues hard sets; LAQ does not.

Figure 4: **When, not what: dose and per-dataset view** (Llama-3.1-8B unless noted; LongBench). **(a)** Four of the five swept backbones saturate at k=2 (k=0: base SnapKV; dotted: FullKV; Llama-3.1-8B also at B=32); the older Llama-3-8B climbs through k=32, as App. D analyzes. **(b)** Per-dataset scores at B=128 ($y>x$ beats SnapKV), zoomed to the sets where methods diverge; red connectors mark the lead over Lookahead Q-Cache (Wang et al., 2025), +3.4 to +8.8 on the labelled sets.

qmsum both deferred variants beat FullKV (App. K). The opening tokens, not the retained content, carry the generation. Off LongBench, Needle-in-a-Haystack (Qwen2.5-7B, B=32)—where a query-aware selector should simply keep the needle—makes the same point: SnapKV recovers 74.7 and H2O 42.1, while DGE$_2$ returns 100.0 against FullKV's 98.9 (App. J).

6.3 Attribution Ablation

The attribution rests on the timing-only control DGE-W of §6.1, which splits DGE's recovery into a pure timing term, SnapKV to DGE-W, and a pure signal term, DGE-W to DGE. On the matched 50-sample gov_report split of Figure 3(b): Δ(timing) = DGE-W − $SnapKV$ = +11.0 and Δ(signal) = DGE − DGE-W = +0.7—timing accounts for 94% of the recovery. The split holds across Table 2: averaged over all fifteen cells, deferring eviction alone lifts SnapKV from 35.34 to DGE-W's 44.17, and swapping in the real decode queries adds nothing further, DGE also averaging 44.17. On Llama-3.1-8B at B=128, App. I's paired bootstrap separates the timing gain of +4.11 from zero but not the 0.04 signal term. The eviction signal—the axis every selection method optimizes—is thus worth at most 0.44 LongBench points in any cell of Table 2 once the cut is deferred: the deferral is the effect. Hence §5's free scorer slot: the draft queries cost nothing, and the base evictor's own scores do as well on average—in the few cells where App. I's bootstrap separates the two, DGE-W is ahead (Qwen2.5-14B at $B\leq64$).

6.4 Dose-Response in k

One decode step is the whole dose, as Figure 4(a) shows on every backbone but Llama-3-8B (App. D; why the dose is backbone-dependent: App. L). At B=128 on Llama-3.1-8B, the one backbone the full grid of Table 13 (App. K) covers, k=2/4/8/16/32 score 46.37/46.29/46.12/46.19/46.13. The minimal dose k=2 (§5) already jumps +4.2 from SnapKV and stays flat within noise thereafter, with or without the draft-query scores (DGE-W$_{16}$: 46.17). The pattern is budget-invariant: at B=32 the dose is flat from k=2 (Table 13) despite SnapKV collapsing to 35.85. This flatness, and DGE-W tracking DGE although its scorer never sees the draft, are consistent with App. E.1's eviction-mass argument.

6.5 Lookahead-Query Eviction

LAQ (Wang et al., 2025) scores with m *pseudo* future queries but still evicts at prefill end—the *what* axis at its limit. Run in its own code base under our matched protocol, its dose saturates ∼2 points

Table 3: **One deferral step clears the *what*-axis ceiling at no prefill cost** (Llama-3.1-8B; matched protocol). DGE rows are ours; App. K and G give the full grids. Latency is wall clock vs SnapKV at a pinned 32 new tokens for short, 512 for long; TTFT uses one fixed 7.8k-token prompt.

(a) Pushing *what* to its limit (LongBench avg.)

Method	Evicts / signal	B=64	B=128
FKV	never	46.24	46.24
SKV	prefill / window	39.38	42.22
LAQ$_{m=4}$	prefill / pseudo	42.75	43.81
LAQ$_{m=8}$	prefill / pseudo	43.11	44.13
LAQ$_{m=16}$	prefill / pseudo	43.17	44.20
SKV+DGE-W$_2$	step 2 / window	46.20	46.33
SKV+DGE$_2$	step 2 / draft	**46.36**	**46.37**

(b) What deferral costs

Method	Extra scoring	TTFT	Latency short/long
FKV	–	0.91×	0.89/0.83
SKV	$O(Lwd)$	1.00×	1.00/1.00
LAQ$_{m=16}$	$O(mLd)$	1.42×	0.95/0.83
H2O	$O(L^2d)$	3.45×	2.15/1.06
SKV+DGE-W$_k$	$O(Lwd)$	0.92×	1.45/1.02
SKV+DGE$_k$	$O(kLd)$	0.93×	1.45/1.02

below FullKV. DGE-W$_2$ defers one decode step and clears LAQ's best dose by +2.1 at B=128 in Table 3(a), and by +5.8 at B=32 (App. L): tighter budgets widen the gap, and Figure 4(b) places the margin on the hard summarization and code sets. Across all fifteen backbone × budget cells, the Avg. column of Table 2 puts the margin at +4.0.

6.6 Complexity and overheads

Table 3(b) prices the deferral (d: head dimension): TTFT improves to 0.93× SnapKV's, since eviction leaves the prefill path, and the one-off cache surgery costs 1.45× on 32-token answers (H2O's accumulator: 2.15×), amortising to 1.02× by 512.

7 Discussion

Sharper benchmarking. The standard protocol—evict at end-of-prefill, then generate—folds two decisions into one number; one deferred-eviction row separates them, and averaged over backbones and budgets a timing-only deferral closes almost all of the SnapKV-to-FullKV gap of Table 2: the reported penalty is timing, not selection quality.

What the objective measures. Attention-output fidelity is a geometric quantity §4's mechanisms already deliver near their ceilings; task quality asks instead for retrievable content and a stable trajectory.

Limitations and research directions. Trajectory anchoring is inferred from ablations, not isolated, and decoding is greedy throughout, so how far it survives sampling is open. LongBench's short answers cap the doses it can probe—at k=16 a third of Llama-3.1-8B's answers terminate inside the draft—pointing at long-form, agentic and multilingual workloads. The draft holds the full cache, so the prototype's working set is 2× FullKV's (1.13× in the native GQA layout, App. G); batching it in a paged-attention server would turn the TTFT gain into throughput.

8 Conclusion

Training-free KV-cache compression evicts at the end of prefill to preserve the attention mass future queries will use—optimizing *what* to keep—and that objective fails in two ways: restored mass does not restore task quality, and extra coverage can hurt when it arrives fragmented. Both occur because eviction fires before the queries that will read the cache exist. DGE removes that cause by deferring eviction until after drafting the first k=2 answer tokens on the full cache; it drops into SnapKV, PyramidKV, H2O and StreamingLLM alike, changing *when* eviction occurs, not *what* is selected. It leads every prior method at every budget on five of six backbones (44.2 on LongBench vs FullKV's 44.3), and the timing-only control DGE-W reaches the same score: the gain is *trajectory anchoring*, from the deferral itself. Before optimizing *what* to keep, fix *when* to evict.

AI USE STATEMENT

In this work, we used generative AI tools to draft the structure of the paper, to revise its sentences, and to correct typographical errors. We have reviewed all AI-assisted text and take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work evaluates an inference-time efficiency method on public benchmarks, LongBench and Needle-in-a-Haystack, with openly released models—Llama-3.1-8B, Llama-3.2-3B, Llama-3-8B, Mistral-7B and Qwen2.5-7B and 14B—each used under its license; it involves no human subjects and collects no personal data. KV-Cache compression changes what a model can attend to, so a compressed deployment can lose content a full-cache run would have used. We therefore report quality against FullKV at every budget rather than cost alone, and state where the deferral does not close the gap, on Llama-3-8B as App. D analyzes. Every baseline follows its published rule or runs in its own released code, and each table marks which rows are ours, so the comparisons can be checked rather than taken on trust.

REPRODUCIBILITY STATEMENT

§6.1 gives the backbones, the sixteen LongBench datasets and their metrics, the KV budgets, and the fixed settings—observation window w=8, pooling kernel 7, draft length k=2 and greedy decoding—with prompt, truncation and context details in App. A. The method is specified in §5 and as pseudocode in App. E; the diagnostic harness behind every coverage and attention-mass figure is described in App. B, and the compensation and selection mechanisms are derived in App. C. Per-dataset scores are in App. K, paired bootstrap intervals in App. I, and the latency and memory microbenchmark in App. G. The implementation builds on the open-source KVCache-Factory codebase, and we will publicly release our core source code on GitHub: the draft-guided eviction policy and the forward pre-hooks that capture the draft queries.

REFERENCES

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. *arXiv preprint arXiv:2603.10899*, 2026.

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 (Volume 1: Long Papers)*, pages 3192–3213, 2024.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.

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

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of NAACL-HLT*, 2021.

Alessio Devoto, Yu Zhao, Simone Scardapane, and Pasquale Minervini. A simple and effective L_2 norm-based strategy for KV cache compression. In *Proceedings of EMNLP*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The LLaMA 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. Multi-News: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of ACL*, 2019.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S. Kevin Zhou. Ada-KV: Optimizing KV cache eviction by adaptive budget allocation for efficient LLM inference. *arXiv preprint arXiv:2407.11550*, 2024.

Yuan Feng, Junlin Lv, Haoyu Guo, Yukun Cao, S. Kevin Zhou, and Xike Xie. Criticalkv: Optimizing KV cache eviction from an output perturbation perspective. *arXiv preprint arXiv:2502.03805*, 2025.

Yu Fu, Zefan Cai, Abedelkadir Asi, Wayne Xiong, Yue Dong, and Wen Xiao. Not all heads matter: A head-level KV cache compression method with integrated retrieval and reasoning. *arXiv preprint arXiv:2410.19258*, 2024.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive KV cache compression for LLMs. In *International Conference on Learning Representations*, 2024.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, 2019.

Daya Guo, Canwen Xu, Nan Duan, Jian Yin, and Julian McAuley. LongCoder: A long-range pre-trained language model for code completion. In *International Conference on Machine Learning*, 2023.

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 COLING*, 2020.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. Efficient attentions for long document summarization. In *Proceedings of NAACL-HLT*, 2021.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of ACL*, 2017.

Jang-Hyun Kim, Jinuk Kim, Sangwoo Kwon, Jae W. Lee, Sangdoo Yun, and Hyun Oh Song. KVzip: Query-agnostic KV cache compression with context reconstruction. In *Advances in Neural Information Processing Systems*, volume 38, 2025. arXiv:2505.23416.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of SOSP*, 2023.

Wonbeom Lee, Jungi Lee, Junghwan Seo, and Jaewoong Sim. InfiniGen: Efficient generative inference of large language models with dynamic KV cache management. In *Proceedings of OSDI*, 2024.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, 2023.

Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of COLING*, 2002.

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, 2024.

Tianyang Liu, Canwen Xu, and Julian McAuley. RepoBench: Benchmarking repository-level code auto-completion systems. In *International Conference on Learning Representations*, 2024a.

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, 2023.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2-bit quantization for KV cache. In *International Conference on Machine Learning*, 2024b.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Piotr Nawrot, Adrian Łańcucki, Marcin Chochowski, David Tarjan, and Edoardo M. Ponti. Dynamic memory compression: Retrofitting LLMs for accelerated inference. In *International Conference on Machine Learning*, 2024.

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

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

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, et al. D2O: Dynamic discriminative operations for efficient generative inference of large language models. *arXiv preprint arXiv:2406.13035*, 2024a.

Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. LOOK-M: Look-once optimization in KV cache for efficient multimodal long-context inference. In *Findings of EMNLP*, 2024b. arXiv:2406.18139.

Yixuan Wang, Shiyu Ji, Yijun Liu, Yuzhuang Xu, Yang Xu, Qingfu Zhu, and Wanxiang Che. Lookahead Q-Cache: Achieving more consistent KV cache eviction via pseudo query. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025. arXiv:2505.20334.

Zheng Wang, Boxiao Jin, Zhongzhi Yu, and Minjia Zhang. Model tells you where to merge: Adaptive KV cache merging for LLMs on long-context tasks. *arXiv preprint arXiv:2407.08454*, 2024.

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

Yuhui Xu, Zhanming Jie, Hanze Dong, Lei Wang, Xudong Lu, Aojun Zhou, Amrita Saha, Caiming Xiong, and Doyen Sahoo. ThinK: Thinner key cache by query-driven pruning. *arXiv preprint arXiv:2407.21018*, 2024.

Jiaming Yang, Chenwei Tang, Liangli Zhen, and Jiancheng Lv. Rethinking KV cache eviction via a unified information-theoretic objective. *arXiv preprint arXiv:2604.25975*, 2026.

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 EMNLP*, 2018.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with activation beacon. *arXiv preprint arXiv:2401.03462*, 2024a.

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

Yuxin Zhang, Yuxuan Du, Gen Luo, Yunshan Zhong, Zhenyu Zhang, Shiwei Liu, and Rongrong Ji. CaM: Cache merging for memory-efficient LLMs inference. In *International Conference on Machine Learning*, 2024b.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, et al. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, 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 NAACL-HLT*, 2021.

Adam Zweiger, Xinghong Fu, Han Guo, and Yoon Kim. Fast KV compaction via attention matching. *arXiv preprint arXiv:2602.16284*, 2026.

Appendix map. Sections A and B fix the protocol behind every number: the LongBench run itself, then the offline diagnostic harness. Section C derives the compensation and selection mechanisms §4 analyzes and holds their two full grids, and Section H the attention-matching loss they optimize. Section E specifies DGE as pseudocode and derives a per-step eviction-mass bound that motivates why deferral can recover FullKV; Section F contrasts it with the other passes that also run before the cut, and Section G prices it in latency, TTFT and memory. Sections D and J are the two settings the body reports only in summary—the sixth backbone and retrieval—and Section I the paired bootstrap behind every interval. Section K holds the per-dataset score tables, Section L the remaining ablations, and Section M the diagnostics that ruled a hypothesis out.

A Experimental details

Backbones and context. We evaluate six instruct backbones. Llama-3.1-8B-Instruct (Dubey et al., 2024) is the primary model (128K native context via RoPE scaling); backbone checks use Mistral-7B-Instruct-v0.2 (Jiang et al., 2023) (32K context, a different architecture family), Qwen2.5-7B-Instruct (Qwen Team, 2024) (a third architecture family with 28 layers and 4 KV heads, ChatML template), Qwen2.5-14B-Instruct (the largest backbone, 48 layers and 8 KV heads, same ChatML protocol), Llama-3.2-3B-Instruct (a smaller model of the same instruct lineage, for a scale check), and Llama-3-8B-Instruct (8K native context, no RoPE scaling; the shorter-context predecessor of the primary backbone). All share the per-head KV-Cache geometry DGE operates on and differ only in long-context adaptation and instruction-tuning generation, contrasted in Table 6.

Datasets and metrics. LongBench (Bai et al., 2024), all 16 English datasets: single-document QA (NarrativeQA, Qasper, MultiFieldQA), multi-document QA (HotpotQA, 2WikiMQA, MuSiQue), summarization (GovReport, QMSum, MultiNews), few-shot learning (TREC, TriviaQA, SAMSum), synthetic retrieval/counting (PassageCount, PassageRetrieval), and code completion (Lcc, RepoBench-P). We report the official per-dataset metric—F1 for QA, ROUGE-L for summarization, classification accuracy for TREC and the synthetic tasks, and edit similarity for code—and the official per-dataset max-new-token budget. Prompts exceeding the truncation limit are middle-truncated (first half + last half) following the official protocol; the limit is 7,500 tokens for the Llama and Qwen backbones (matched across methods) and 31,500 for Mistral.

Decoding and prompt protocol. Greedy decoding throughout, so results are deterministic and there is no seed variance to average over. We use the repository's corrected Llama-3 chat template and stop-token set (adding `<|eot_id|>` and a per-dataset newline stop for the few-shot sets, which otherwise over-generate); Mistral uses its `[INST]` wrapping. Every method sees the identical decoding pipeline and matched sample sets.

Compression settings. Per-head budget $B \in \{32, 64, 128, 512\}$ (B=128 unless noted), observation window w=8, max-pool kernel 7 (the SnapKV recipe). The compressed cache is stored per query head (GQA layout). DGE draft length $k \in \{2, \ldots, 32\}$ (k=2 for both DGE and the timing-only DGE-W in the main tables, k=16 for the two deferred markers of Figure 2(a)); at k=1 the policy coincides with the base evictor by construction (the first token comes from the prefill forward), so k=2—one full-cache decode step—is the smallest dose the method admits; whether it is also *sufficient* is backbone-dependent, as §D shows.

Software and hardware. PyTorch with `transformers` 4.44.2; flash-attention 2.6.3 for the window/positional scorers and DGE, and the SDPA backend for the compensation variants (β-bias and value absorption apply a decode-time attention-mask bias). Each run uses a single 80 GB GPU; all reported cells are our own runs (H2O is omitted on the Mistral-7B backbone).

Lookahead Q-Cache comparison. We run LAQ (Wang et al., 2025) in its own released code base on the same backbone, datasets, truncation, and greedy decoding, porting in only our Llama-3 chat template and stop tokens. Without this alignment LAQ's verbose, unterminated generations score $\sim$5 points lower (e.g. Qasper $10.96 \rightarrow 39.37$), so the alignment is required for the matched comparison of Table 14 and §6.5.

B Diagnostic details

All diagnostics share one harness (released with the code). The model runs as plain FullKV; forward *pre*-hooks on every attention layer recompute post-RoPE Q, K from the layer input, so the probe observes exactly what the eviction code would see without altering the forward pass. Selection replicates SnapKV's scoring path operation for operation—float32 softmax over the last w queries, score sum over past keys, maxpool (kernel 7), per-head top-B—so a rule's diagnostic score and its deployed behavior cannot diverge through dtype or pooling drift.

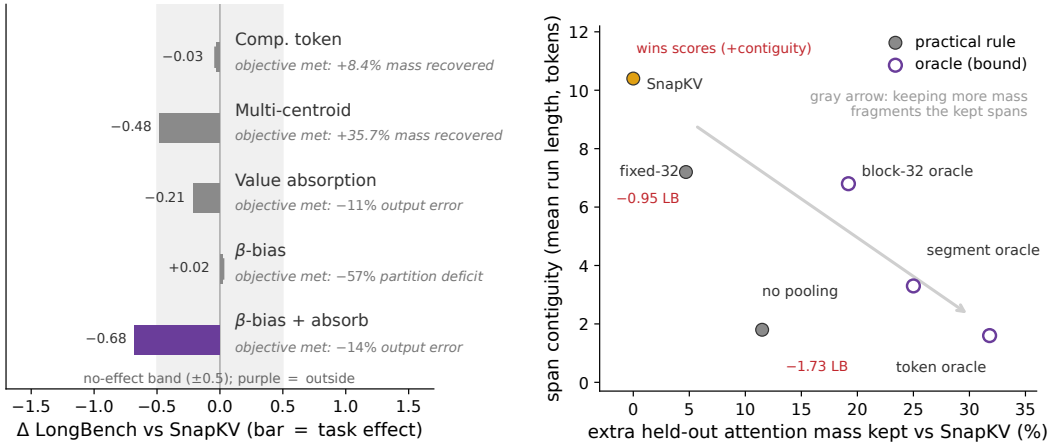
The key design choice is the *held-out split*: every decode-step query is replayed against the full uncompressed K, giving the attention mass a FullKV run would place on the tokens a rule evicts (β_{dec}, §4). In-window quantities (β_{win}) are what the rule can see; decode-side quantities are what it is graded on. Every selection variant in the ladder of §4 (pooling ablations, window sizes, layer-decay, key-norm re-weighting, block granularities $8/16/32$, cross-layer sharing, repeat-prefill) is scored by this held-out decode coverage under identical budgets; Table 5 reports the resulting granularity ladder and Section M logs the rules the diagnostic ruled out.

C Compensation and selection details

Figure 5 summarizes the two key points of this section: compensation mechanisms meet their attention-level objective yet move LongBench by $\sim$0 in Table 4, and selection rules trade held-out coverage against span contiguity—SnapKV keeps the least raw mass but the longest contiguous runs and wins scores, while raw-mass rules and oracles buy coverage by fragmenting spans in Table 5.

Mechanisms (all train-free). (i) a single attention-matched compensation token per head (value centroid, phase-aligned key centroid, window-estimated mass target matched by key-scale search; FA2-compatible); (ii) multi-centroid variant (c spherical-$k$means cluster tokens); (iii) value absorption along the empirical window co-attention transport; (iv) per-token bias in the style of FAST (Zweiger et al., 2026), $\beta_i = \log(1 + m_i/s_i)$ applied through the attention mask.

Baseline. Every LongBench Δ tabulated in Tables 4 and 5, and each of the four *what*-axis points of Figure 2(a), save the no-pooling entry, is measured against one SnapKV run on all sixteen sets, whose average is 42.15. That is 0.06 below the 42.22 Table 3(a) and Table 2 print (42.1535 against 42.2161), and the gap is one of sample size rather than of runs: this run stops at 50 of the 200 gov_report and qmsum prompts, its predictions there are a prefix of the master sweep's, and the two agree exactly on the other fourteen sets. Each Δ here is that variant's own sixteen-set average minus this 42.1535—the no-pooling entry, in Table 5 and in Figure 2(a) alike, excepted, its retained -1.73 coming from a 3-dataset $\times$ 50 judgment run rather than a sixteen-set average—so the variant's own number is



(a) Compensation: objective met, score flat.

(b) Selection: coverage vs contiguity.

Figure 5: **The attention-mass analysis, visualized** (data from Tables 4 and 5): **(a)** one row per compensation mechanism, **(b)** one point per selection rule.

Table 4: **Compensation dose-response.** Weak positives appear only on diffuse tasks (narrativeqa $+1.1$, gov_report $+0.7$, qmsum $+1.3$). All five rows are our own train-free implementations, not runs of the published methods: the compensation-token and multi-centroid rows fold evicted entries into retained ones in the style of Wan et al. (2024b) and Wang et al. (2024), and value absorption in the style of Zhang et al. (2024b); the per-token bias is FAST-style (Zweiger et al., 2026) but a closed form $\beta_i = \log(1+m_i/s_i)$ that replaces FAST's NNLS fit; the last row combines that bias with absorption. All five are derived below. Effects here are signed measurements, so a reduction reads negative; Table 1(a) reports the same quantities as improvements, where higher is better throughout.

Variant	Attention-level effect	LongBench Δ
Comp. token (v1)	recovers 8.4% of lost mass	−0.03 (16 ds)
Multi-centroid c=8	recovers 35.7% of lost mass	−0.48 (16 ds)
Value absorption	output error $-11 \ldots -14\%$	−0.21 (16 ds)
Per-token bias	partition deficit -57%	+0.02 (16 ds)
Bias + absorption	output error $-14 \ldots -17\%$	−0.68 (16 ds)

untouched and every Δ is exactly 0.0626 more generous than the same variant scored against the 42.2161 run, whatever that variant's own sample counts are (value absorption and the per-token bias are short on sets other than gov_report and qmsum). The other five points of Figure 2(a) are measured against that 42.22, but only SnapKV and the two DGE markers come from the master sweep: PKV is its own 200-prompt run and LAQ the separate code-base run Table 14 reports.

Mechanistic findings. (a) Renormalized mass flows *multiplicatively*: inflow at a retained token correlates 0.97 with its own attention, not key similarity—so similarity-merging (LOOK-M/KVMerger; Wan et al., 2024b; Wang et al., 2024) cannot route mass correctly (kernel-transport flow corr. ≈ 0). (b) Post-RoPE arithmetic key averaging loses 30% of key norm to phase interference (mean per-pair resultant $R \approx 0.33$, the whole-key norm ratio being 0.70); a circular-mean construction preserves it. (c) At $B=128$, an unstructured least-squares value correction—the counterpart of FAST's ordinary-least-squares C_v fit, not of its NNLS mass fit—ridge-fit on half of the real decode queries, does not transfer to the held-out half: its residual is $1.08\times$ the uncorrected error, so an oracle fit with $B \times d$ free parameters ends up worse than no correction at all. Structured closed forms have no fit to overfit. (d) Repeat-prefill queries follow induction/copy patterns (run length 1.2) and score selection *worse* than the window (-13% held-out coverage).

Compensation token (mass matching). For evicted set E with window-score weights w_j ($\sum_{j \in E} w_j = 1$), the token's value is the centroid $v^* = \sum_j w_j v_j$. Its key direction is the *phase-*

Table 5: **Selection decomposition.** The SnapKV row is the published method (Li et al., 2024); every rule below it is ours—two variants of its selection (block granularity, pooling removed) and three oracle bounds—so no external method is run here. The real-decode-query row of Table 1(b) is absent because it is not a prefill-time rule: it needs queries that do not exist until decoding, which is what §5 defers the cut to obtain. Its LongBench entry, $+3.98$ against the 42.22 sweep run, is therefore the SKV+DGE$_{16}$ row of the dose grid of Table 13 (46.19 printed; the difference is taken on unrounded averages)—the same k=16 DGE marker as Figure 2(a)—not a rule scored on this ladder; the coverage and run length beside it are measured offline like every other row here.

Rule (B=128)	held-out coverage $\Delta\uparrow$	run len.	LongBench $\Delta\uparrow$
SnapKV (window+maxpool)	–	10.4	–
fixed 32-blocks	$+4.7\%$	7.2	-0.95
no pooling	$+11.5\%$	1.8	-1.73
token oracle	$+31.8\%$	1.6	(bound)
block-32 oracle	$+19.2\%$	6.8	(bound)
segment oracle	$+25.0\%$	3.3	(bound)

aligned centroid: each RoPE rotate-half pair $(d, d{+}64)$ is treated as a complex number z_j, and the weighted complex sum is renormalized to the weighted magnitude, $\tilde{z} = \frac{\sum_j w_j z_j}{|\sum_j w_j z_j|} \sum_j w_j |z_j|$, so the circular-mean phase is kept and no norm is lost to phase interference. (A naive arithmetic mean shrinks by the resultant length $R = |\sum_j w_j z_j| / \sum_j w_j |z_j| \in [0,1]$; R is also our phase-dispersion measure.) The token's target mass is $\beta^* = \beta_{\text{win}} \cdot R$ (dispersed phases shrink the target), clamped to $[10^{-4}, 0.5]$. With retained log-partition $\log \hat{Z}_i$ per window query i and $c_i = q_i^\top u/\sqrt{d}$ for unit key direction u, the mass the token receives at key scale s is $\sigma(s\,c_i - \log \hat{Z}_i)$; s is chosen on a 129-point grid over $[0,\ 25/\max_i |c_i|]$ (exp-safe) to match β^*. Pure key *scaling*—no logit bias—so the token runs unchanged under FlashAttention-2. The multi-centroid variant first splits E by weighted spherical k-means (deterministic position-stratified init) and builds one token per cluster.

Value absorption and the closed-form bias. Both use the same empirical transport: with window-attention matrix A, co-attention $C_{ij} = \sum_w A_{wi} A_{wj}$ (retained i, evicted j), column-normalized to $T_{j\to i}$. Absorption folds evicted values into retained ones, $c_i = v_i + \big(\sum_j \mu_j T_{j\to i} v_j - m_i v_i\big)/(s_i + m_i)$ with transported mass $m_i = \sum_j \mu_j T_{j\to i}$—closed form, keys and cache shape untouched, decode unchanged. The per-token bias is the closed form $\beta_i = \log(1 + m_i/s_i)$, clamped at 10, added to the decode attention mask, which grows token i's softmax mass from s_i toward $s_i{+}m_i$. This replaces Zweiger et al. (2026)'s NNLS fit, which FAST solves on thousands of reference queries per KV-head per context—a reservoir cap of $50{,}000$, averaging $\sim$16,000 per head on QuALITY. Refit train-free on the w=8 observation queries a single prefill affords, that regression can drive *in-window* error to zero by memorizing them, precisely the failure mode the held-out decode replay is built to catch (cf. repeat-prefill in Section M); the closed form has no fit to overfit, and offline replay measured 11–14% attention-output error reduction for absorption at B=128–512 against -1.5% for the bias alone, as Section M logs.

D THE SIXTH BACKBONE: LLAMA-3-8B AND THE MINIMAL STEERING DOSE

We repeat the main experiment on Llama-3-8B-Instruct, the predecessor of our primary backbone. The two models share their entire architecture and KV geometry—so DGE operates on an identical per-head cache—and differ only in long-context adaptation: Llama-3.1 adds a $16\times$ longer context via RoPE scaling and a stronger, chattier instruction-tuning round, itemized in Table 6. This isolates the effect of the backbone's generation behavior on the deferral.

The minimal steering dose is backbone-dependent. On Llama-3 the *single*-step deferral (k=2) that erases the gap on Llama-3.1 needs a longer draft: SKV+DGE$_2$ recovers $+2.1$ of the 5.8-point gap at B=64 (already the best compression method in the block) and trails SnapKV by 0.77 at B=128. Crucially this is not degeneration—the anchored openings are fluent and often *more* complete

Table 6: **Two backbones, identical KV geometry.** The architecture and long-context rows are the published model configurations; only the *Empirical* block is ours, LongBench averaged over the 16 English sets, with *gap closed* the share of the SnapKV-to-FullKV gap the named draft length recovers ("full" = within noise of FullKV). Abbreviations, used throughout the appendix tables: FKV = FullKV, SKV = SnapKV (Li et al., 2024), PKV = PyramidKV (Cai et al., 2024), SLM = StreamingLLM (Xiao et al., 2024), H2O (Zhang et al., 2023), LAQ = Lookahead Q-Cache (Wang et al., 2025); every DGE and DGE-W row is ours.

	Llama-3-8B-Instruct	**Llama-3.1-8B-Instruct**
Architecture (shared ⇒ identical KV geometry)		
Layers / hidden	32 / 4096	
Query / KV heads (GQA)	32 / 8	
Head dim	128	
Vocabulary	128,256	
RoPE θ	500,000	
Long-context adaptation (differ)		
Native context	8,192	131,072
RoPE scaling	none	llama3, factor 8
Release	2024-04	2024-07
Instruction tuning	initial	improved (longer, chattier)
Empirical (our matched LongBench runs)		
FullKV avg	43.58	46.24
SnapKV avg (B=128)	40.50	42.22
gap to FullKV	3.08	4.02
SKV+DGE minimal dose	$k\approx 8$	k=2
gap closed (k=32 / best)	72%	full

answers—but a *longer draft recovers the gap*: the dose climbs monotonically $39.73 \to 40.88 \to 42.09 \to 42.38 \to 42.73$ at k=2/4/8/16/32. There is no knee—each extra anchor token buys a little more, with 79% of the total climb done by k=8 and the remaining 0.64 spread over k=8 → 32—and the ramp only overtakes SnapKV itself between k=2 (-0.77 against the 40.50 baseline) and k=4 ($+0.38$). SKV+DGE$_{32}$ (42.73) is the best compression method in the block, above SnapKV 40.50 and recovering 72% of the 3.08-point gap to FullKV 43.58 (large per-dataset jumps: Qasper $35.1 \to 42.5$, SAMSum $39.0 \to 42.7$). Two consequences: the headline k=2 result is a property of decisive-opening backbones (the modern long-context generation), and the §4 analysis is unaffected, having been measured on Llama-3.1. An answer-length-controlled rescore separates the verbosity artifact from the real deficit on the worst case, TriviaQA: SKV+DGE$_2$'s first-line score is 75.8 against 89.8 (SnapKV) and 90.6 (DGE$_{16}$)—the k=2 draft leaves Llama-3 mid-preamble, and its chattier full-sentence openings ("United States of America." where the reference is "United States") are scored down by extractive F1. Truncating each first line to the longest reference length (+2 words) recovers +5.9 for DGE$_2$ ($75.8 \to 81.7$) while moving every other method ≤ 1.0 and Llama-3.1's DGE$_2$ by 0.1 ($92.5 \to 92.6$): roughly 40% of the deficit is phrasing, and the residual ∼9 points is what Table 7's longer drafts close (90.3 at k=8).

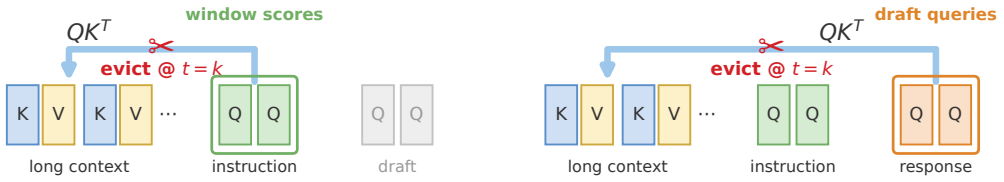
What the dose actually measures. A larger k changes two things at once: the scorer sees more real decode queries, and more tokens are generated before the cut. Three probes separate them, and only the second of the two survives.

(i) It is not a slower start. The natural reading of "longer steering segment" is that Llama-3 takes more tokens to reach its answer. It does not. Scoring the FullKV generations of both backbones for the token position at which a gold answer string first appears—six extractive LongBench sets, 505 and 550 predictions in which a gold string appears at all—gives a median onset of 0 tokens on *both*, with 92% (Llama-3) and 98% (Llama-3.1) of answers begun within two tokens. The chattier phrasing documented above is real, but it does not delay the answer.

(ii) It is not slower selection convergence. If the dose bought scoring information, Llama-3's chosen tokens should keep moving for longer. We replayed DGE's own scoring path—mean softmax attention of the first k draft queries over past keys, group-meaned across GQA heads, per-head top-B—and

measured the overlap between the set kept at k and the set kept at k=32, over six datasets (63 and 64 prompts). The two backbones are indistinguishable: 0.605 (Llama-3) vs 0.611 (Llama-3.1) at k=2, and 0.858 vs 0.838 at k=16. Selection converges at the same rate on both.

(iii) It is consistent with anchoring. The discriminating run is the timing-only control at the same doses. DGE-W$_2$, DGE-W$_{16}$ and DGE-W$_{32}$ keep SnapKV's window scores—the draft contributes *no* scoring information whatsoever—yet they climb $39.81 \rightarrow 42.19 \rightarrow 42.61$ at k=2/16/32, landing within 0.08, 0.19 and 0.12 of full DGE at each dose in Table 7. The dose does not buy a better eviction signal; it buys tokens generated on the full cache before the cut. Llama-3's trajectory simply takes longer to become robust to eviction, which is a statement about *when* the cut can safely happen, not about *what* it should keep. Even the one parameter that looked like a scoring choice turns out to be timing.



(a) SnapKV+DGE-W$_k$: window scores, deferred cut. (b) SnapKV+DGE$_k$: real draft queries, deferred cut.

Figure 6: **DGE-W vs DGE: same *when*, different *what***, drawn in Figure 1's visual language. The two variants differ *only* in the scoring source; the draft is generated on the full cache either way, though faded in (a).

Table 7: **Backbone ablation on Llama-3-8B-Instruct** (8K context, no RoPE scaling; cf. Llama-3.1 in Table 2). Our matched runs; **bold** is the best compression method per column per budget block. Rows: FKV = FullKV; SKV = SnapKV (Li et al., 2024); PKV = PyramidKV (Cai et al., 2024); H2O (Zhang et al., 2023); SLM = StreamingLLM (Xiao et al., 2024); the SKV+DGE and SKV+DGE-W rows are ours.

	Single-Document QA			Multi-Document QA			Summarization			Few-shot Learning			Synthetic		Code		
Method	NrtvQA	Qasper	MF-en	HotpotQA	2WikiMQA	Musique	GovReport	QMSum	MultiNews	TREC	TriviaQA	SAMSum	PCount	PRe	Lcc	RB-P	Avg.
Llama-3-8B-Instruct, KV Size = Full																	
FKV	21.26	42.60	47.78	47.32	39.10	22.78	30.44	22.70	27.54	74.00	90.56	42.67	8.50	67.50	59.38	53.19	43.58
Llama-3-8B-Instruct, KV Size = 64																	
SKV	18.51	28.46	41.86	44.32	35.34	**22.00**	19.35	**19.64**	19.46	50.50	**89.22**	36.25	5.75	66.00	55.29	**52.20**	37.76
PKV	16.99	30.86	**43.15**	43.96	34.83	21.63	19.54	19.38	20.03	56.50	86.76	36.62	5.75	66.00	53.75	49.13	37.80
H2O	17.68	25.46	32.28	42.31	28.82	20.67	**21.44**	17.26	**23.66**	51.50	83.75	22.42	6.25	63.50	48.80	41.91	34.23
SLM	16.76	26.72	30.11	41.78	32.92	17.02	15.73	19.17	15.58	39.50	71.63	32.77	5.50	66.50	53.25	52.08	33.56
SKV+DGE$_2$	20.26	**35.56**	42.20	**45.15**	**38.70**	20.17	20.30	18.01	22.42	**73.00**	77.82	40.92	**7.00**	66.50	58.64	51.51	**39.88**
SKV+DGE-W$_2$	**20.73**	34.22	42.46	43.98	36.92	20.52	20.30	18.01	22.03	71.50	68.96	**41.46**	3.50	**67.50**	**58.66**	51.55	38.89
Llama-3-8B-Instruct, KV Size = 128																	
SKV	18.70	35.14	45.08	45.94	36.39	22.13	21.01	20.35	21.73	65.50	89.78	38.99	5.50	**68.00**	58.71	**54.98**	40.50
PKV	19.17	35.49	44.57	46.03	36.73	22.92	21.02	20.50	21.97	67.50	89.23	38.95	6.00	67.50	57.71	51.43	40.42
H2O	17.70	31.10	35.84	43.42	33.38	20.91	**23.02**	17.79	**24.82**	61.00	86.94	25.97	6.12	60.71	52.23	45.08	36.63
SLM	16.43	25.81	31.69	42.56	33.34	17.43	17.04	19.45	18.04	45.50	74.24	36.45	7.00	65.50	56.24	53.22	35.00
SKV+DGE$_2$	20.33	36.09	43.21	45.39	38.89	20.25	20.33	18.71	22.86	**74.00**	75.81	41.62	5.50	65.50	57.76	49.35	39.73
SKV+DGE-W$_2$	19.43	35.53	44.50	45.33	38.16	20.44	19.90	19.77	22.66	73.00	77.04	41.69	6.50	67.00	57.43	48.62	39.81
SKV+DGE$_4$	20.72	37.62	44.62	46.87	38.73	22.25	20.53	18.62	23.09	**74.00**	85.05	41.91	6.00	66.00	57.30	50.79	40.88
SKV+DGE$_8$	**20.94**	39.47	46.56	47.08	39.10	22.74	20.32	19.45	23.32	**74.00**	90.34	42.16	**8.50**	67.50	58.97	52.91	42.09
SKV+DGE$_{16}$	**20.94**	41.24	46.67	47.32	**39.12**	22.78	20.33	20.18	23.62	**74.00**	**90.56**	42.78	**8.50**	67.50	59.35	53.11	42.38
SKV+DGE-W$_{16}$	20.92	40.95	46.09	**47.35**	**39.12**	**22.93**	20.10	19.76	22.69	**74.00**	**90.56**	42.24	8.25	67.50	59.21	53.33	42.19
SKV+DGE$_{32}$	20.92	**42.45**	**47.34**	47.32	39.10	22.78	21.97	**21.87**	24.06	**74.00**	**90.56**	42.71	**8.50**	67.50	59.39	53.25	**42.73**
SKV+DGE-W$_{32}$	20.92	42.18	47.33	47.32	39.10	22.78	21.46	21.37	23.06	**74.00**	**90.56**	**42.85**	**8.50**	67.50	**59.45**	53.31	42.61

E DRAFT-GUIDED EVICTION IMPLEMENTATION

Implementation notes. Algorithm 1's prototype is two stock `generate()` calls around one cache edit. *Phase A* lifts every layer's `max_capacity_prompt` to 2^{30} (compression disabled) and registers forward pre-hooks that recompute post-RoPE queries from each layer's input: per-step decode queries (the eviction evidence), the last-w prompt queries (for the DGE-W control), and, for the H2O base only, all prompt queries (its accumulator; freed per layer as consumed, ∼60 MB/layer). *Phase B* edits the cache in place, per layer: score, top-B, *sort indices to preserve temporal order*, then gather [retained past] $\|$ [window + draft tail]; the continuation call resumes from the edited

Algorithm 1 Draft-Guided Eviction with its two scorers: DGE$_k$ (draft queries, the default) and the timing-only DGE-W. Both wrap any prefill-time evictor E (retention rule R_E, observation window w), draft k tokens on the full cache (**A**), and evict at $t{=}k$ (**B**–**C**), differing only at line **B** (cf. Figure 6). The baseline E instead evicts at $t{=}0$.

Require: prompt $x_{1:L}$, per-head budget B, draft length k, base evictor E, generation budget T
Ensure: answer y

$\mathcal{C},\ y_1,\ \{q_1^{(\ell,h)}\} \leftarrow \text{prefill}(x_{1:L})$ ▷ full cache — the peak E also holds; prefill emits y_1
for $t = 2$ **to** $\min(k, T)$ **do**
  $y_t,\ \{q_t^{(\ell,h)}\} \leftarrow \text{decode}(\mathcal{C})$; append K, V of y_t ▷ **A: draft** + capture real queries
  if $y_t = \text{EOS}$ **then**
    return $y_{1:t}$ ▷ answer ended; cache is freed — never evict
  end if
end for
for layer $\ell = 1$ **to** N, head $h = 1$ **to** H **do**
  $s \leftarrow \frac{1}{k}\sum_{t \le k} \text{softmax}\big(q_t^{(\ell,h)} K^{(\ell,h)\top}/\sqrt{d}\big)$ ▷ **B: score** — DGE: the k real draft queries
  or $s \leftarrow s_E^{(\ell,h)}$ ▷ DGE-W: E's window scores — *only this line differs*
  $I \leftarrow R_E\big(s,\ B - w\big)$ ▷ top-B (SKV), layer budgets (PKV), sinks (SLM)
  $\mathcal{C}^{(\ell,h)} \leftarrow \mathcal{C}^{(\ell,h)}\big[\, I \cup \text{window} \cup \text{draft} \,\big]$ ▷ one in-place edit, temporal order kept
end for
for $t = k{+}1$ **to** T **or** EOS **do**
  $y_t \leftarrow \text{decode}(\mathcal{C})$; append K, V ▷ **C: continue** at E's decode memory
end for
return y

cache (retained tokens' RoPE positions are already baked into their cached keys, so no re-indexing is needed). *Early termination:* if the draft ends (EOS, or fewer than k tokens produced), phase A's output is returned as-is and no eviction ever fires; this is the FullKV-identical path whose per-dataset frequency Table 16 accounts. Latency and memory of the two-phase prototype are measured in App. G: $+9.9\%$ over SnapKV end-to-end, flat in k, with TTFT $0.93\times$.

E.1 Why deferral can recover FullKV: an eviction-mass argument

Write the per-head decode output at step t as $o_t = \sum_j a_{t,j}\, v_j$ with weights $a_{t,j} = \text{softmax}_j(q_t^\top k_j/\sqrt{d})$. Split the keys into the retained set R and the evicted set E, and let $m_t(E) = \sum_{j\in E} a_{t,j}$ be the *evicted mass*—the attention the step-t query places on discarded tokens. Renormalizing over R gives the compressed output $\tilde{o}_t = \sum_{j\in R} \frac{a_{t,j}}{1-m_t} v_j = \mu_R^t$; writing μ_E^t, μ_R^t for the attention-weighted value means over E, R we have $o_t = (1-m_t)\mu_R^t + m_t\mu_E^t$, hence

$$\|o_t - \tilde{o}_t\| \;=\; m_t(E)\,\big\|\mu_E^t - \mu_R^t\big\| \;\le\; 2\, m_t(E)\,\rho_v, \qquad \rho_v = \max_j \|v_j - \bar{v}\|. \tag{2}$$

The per-step error is *linear in the evicted mass* $m_t(E)$ and vanishes when $m_t(E){=}0$. Summing over decoding (errors accumulate to first order as each output feeds the next step), the end-to-end deviation is bounded by $\sum_t m_t(E)\,\|\Delta\mu_t\|$, and the eviction *time* sets the summation range:

$$\mathcal{L}_{SnapKV} \approx \sum_{t=1}^{T} m_t(E_w)\|\Delta\mu_t\| \;\;\text{(cut at } t{=}0), \qquad \mathcal{L}_{\text{DGE}} \approx \sum_{t=k+1}^{T} m_t(E)\|\Delta\mu_t\| \;\;\text{(cut at } t{=}k). \tag{3}$$

Only the per-step bound (2) is exact; three observations then suggest an ordering. **(i) Termination.** A dataset-dependent fraction ρ of answers finishes within k tokens ($T \le k$)—11.2% overall at $k{=}2$, up to 92.5% on passage_count in Table 16; for those the sum in $\mathcal{L}_{\text{DGE}}$ is empty, so DGE and DGE-W equal FullKV *exactly*. **(ii) Anchoring.** For $T{>}k$, the k tokens $y_{1:k}$ are produced on the full cache and their KV is *kept* (recent), so later queries commit to the context those tokens select and $m_t(E)$ decays for $t{>}k$; the deferred terms in (3) are small even for the window-chosen set E_w (this is DGE-W). **(iii) Signal.** DGE further picks E to minimize $\sum_{t\le k} m_t(E)$ from the *real* draft queries, which at $k{=}16$ (DGE$_{16}$) cover $+16.5\%$ more held-out decode mass than the window estimate (§4); this shrinks $m_t(E)$ for $t{>}k$ a little more, but by (i)–(ii) the residual is already small, so the gain over DGE-W is second order. Together they suggest $\mathcal{L}_{SnapKV} \ge \mathcal{L}_{\text{DGE-W}} \approx \mathcal{L}_{\text{DGE}} \ge 0 = \mathcal{L}_{\text{FKV}}$, i.e. SKV

$<$ DGE-W $\approx$ DGE $\approx$ FullKV, consistent with the measured $42.2 < 46.17 \approx 46.19 < 46.24$ at k=16 (DGE-W$_{16}$ and DGE$_{16}$); at the headline k=2 both deferred rows edge past FullKV, by 0.09 and 0.12 in Table 10's paired bootstrap, a margin it does not resolve from zero.

Why DGE and DGE-W barely differ. The two share every step and differ only in the evicted set, so their loss gap is $\mathcal{L}_{\text{DGE-W}} - \mathcal{L}_{\text{DGE}} = \sum_{t>k} \big[m_t(E_w) - m_t(E)\big] \|\Delta\mu_t\|$. Two effects make this doubly small. First, by anchoring (ii) both $m_t(E_w)$ and $m_t(E)$ are already small for $t>k$, so each summand is a difference of two small numbers. Second, the window set E_w and the draft-query set E *largely overlap*: both keep the salient tokens and disagree only on the residual $+16.5\%$ of decode mass (k=16), which anchoring has already made negligible after step k. The gap is thus second order in the deferred mass, and DGE-W tracks DGE to within measurement noise (46.17 vs 46.19 for DGE-W$_{16}$ and DGE$_{16}$; DGE-W even edges DGE at B=32 and k=2, 46.40 vs 46.39 for DGE-W$_2$ and DGE$_2$). In one line: once the cut is deferred, *which* tokens are dropped barely matters—the timing has already removed the error that selection was fighting over.

F DRAFT-LIKE PASSES COMPARED

Several training-free compression methods insert an extra forward pass between prefill and generation because the observation window is a weak estimate of the queries the answer will actually ask. Table 8 compares them along three design choices: what the pass generates, whether its compute survives into the output, and which axis its product feeds. Repeat-prefill (Kim et al., 2025) replays the prompt and scores keys by reconstruction; self-study (Zweiger et al., 2026) generates synthetic Q&A about the context to calibrate biases and values, at an up-front per-context cost; Lookahead Q-Cache (Wang et al., 2025) synthesizes m pseudo future queries and re-scores retention with them. All three discard the pass afterwards, and all three still cut at prefill end: the pass only sharpens *what* is kept. Speculative decoding (Leviathan et al., 2023) appears because DGE is mechanically closer to it than to the eviction scorers—a draft generated by the model itself, kept when it is right—but it targets latency, not memory. LookaheadKV (Ahn et al., 2026) is the trained sibling of the same *what*-axis move, predicting lookahead scores draft-free; we compare training-free passes only.

DGE breaks both invariants at once. Its pass is the answer's own first k tokens, so the compute is not an overhead to amortize but output the user receives either way; and its product is not a better score but a later cut. The ablations say the second difference is the one that pays: with SnapKV's unmodified window scores, the timing-only DGE-W stays within 0.44 of full DGE in every cell of Table 2, while LAQ—a strictly richer scoring signal evicting at the old time—saturates $\geq$2 points below the pure timing change at its best dose in Table 14. Sharpening *what* without moving *when* runs into a ceiling that one deferred step clears.

Table 8: **Draft-like passes in training-free KV compression:** repeat-prefill (KVzip; Kim et al., 2025), self-study (FAST; Zweiger et al., 2026), Lookahead Q-Cache (Wang et al., 2025), speculative decoding (Leviathan et al., 2023).

Extra pass	Queries	Kept?	Evicts	Axis moved
Repeat-prefill	reconstruction	no	prefill end	what (scores)
Self-study	generated Q&A	no[a]	prefill end	what (β, values)
Lookahead Q-Cache	pseudo future	no	prefill end	what (scores)
Speculative decoding	draft tokens	if verified	—	latency
DGE (ours)	first k answer tokens	yes (answer prefix)	step k	**when** (scorer: free slot)

[a] Amortizable only under prefix-cache reuse.

G COMPLEXITY AND OVERHEADS

Table 9 reports, per method, when it evicts, the extra scoring compute, peak/decode KV memory, and measured per-sample latency relative to SnapKV. All evictors share $O(L)$ *asymptotic* peak memory (the end-of-prefill cache) and $O(B{+}t)$ decode memory; DGE changes only the eviction time, holding the full cache for k extra decode steps before dropping to the same steady state.

Where the overhead actually goes. To separate the method's intrinsic cost from our prototype's, we ran a controlled microbenchmark (Llama-3.1-8B, B=128, a fixed 7,762-token prompt, 256 new tokens, 1 warm-up + 3 timed repeats on one A100-class GPU; Table 9's Latency column comes from the separate LongBench probe described there and is not directly comparable). Three things follow.

(i) The dose is free. Total latency is flat in k: 10.88s, 10.49s and 10.49s at k=2, 4, 8—a 3.7% spread with no monotone trend, against a run-to-run σ of up to 0.23s. Quadrupling the draft costs nothing measurable, so the k full-cache decode steps are *not* the cost driver. The overhead over SnapKV (+0.96s, +9.9% here) is therefore the *one-off* per-layer cache surgery.

(ii) TTFT improves. Because the draft runs before any eviction, the first token is produced on the uncompressed cache and the compression never sits on the prefill critical path: TTFT is 0.93× SnapKV's, essentially FullKV's (0.91×). H2O is the opposite case—its $O(L^2)$ accumulator runs during prefill, giving a 3.45× TTFT that its total latency hides whenever the generation is long (2.15× on short answers, 1.06× on long-form). LAQ is the same case for the same reason: its m pseudo queries come from an extra forward pass that is discarded, so the whole pass lands before the first token. Measured under this protocol in LAQ's own code base (m=16, 1 warm-up + 5 repeats), it is 1.42×. That code base's SnapKV is not ours (1.068 vs 1.170 s), so we anchor on FullKV, unpatched in both and agreeing to 0.2% (1.0593 vs 1.0616 s): LAQ is 1.561× FullKV there and FullKV is 0.907× SnapKV here. Dividing LAQ straight by our SnapKV gives 1.41×, so the anchoring changes little. DGE's draft is the answer's opening, and is not discarded.

(iii) The real price is memory, and it is layout, not deferral. Separating the 14.96 GiB of weights from the working set (KV-Cache + activations), the peak is 5.69 GiB for SnapKV, 6.58 GiB for FullKV and 13.13 GiB for DGE$_2$—2.0× FullKV's working set, and dose-independent (13.126 vs 13.132 GiB at k=2 and k=8), confirming (i): the cost is the one-off draft machinery, not holding the cache longer. The dominant term is the *cache layout*: to score eviction per head, this codebase repeats KV to all 32 query heads *before* caching, so the uncompressed prefill cache is 3.79 GiB where the native 8-head GQA layout stores 0.95 GiB. Every evictor pays this constant, but only DGE pays it at full prompt length, because only DGE holds the cache uncompressed through the draft.

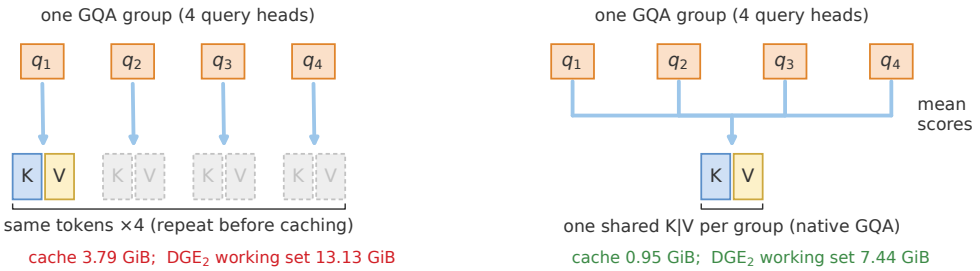
LAQ shows the same cost from the same cause. Holding the generated count fixed at 256 on this prompt, so every method does equal work, its working set is FullKV's, not SnapKV's: 7.60 against FullKV's 7.62 and SnapKV's 5.68 GiB. The lookahead pass runs before any eviction, so the peak is taken on the uncompressed cache—the same structural reason its TTFT is 1.42×. (These three are measured with the count pinned, which FullKV's uncapped cache grows with, so its 7.62 is not the 6.58 above; SnapKV's 5.68 and DGE's 13.11 reproduce the 5.69 and 13.13 either way.)

Removing it, measured. We implemented the fix sketched in Figure 7(b): cache in the native GQA layout, score eviction with the captured query-head queries, and mean the scores over each group of 4 query heads (the codebase's `gqa_score_agg` convention). DGE$_2$'s working set drops 13.13 → 7.44 GiB—from 2.0× to 1.13× FullKV—exceeding the +2.84 GiB layout term because FlashAttention-2 consumes GQA natively, so the 32-head expansion disappears from the *compute* path as well as the cache. The result is base-independent (PKV+DGE$_2$: identical 13.126/7.439 GiB in the two layouts) and score-neutral on a spot check (Llama-3.1, B=128: Qasper −0.06, TREC ±0.00, SAMSum −0.34—all inside the ±0.5 no-effect band). The full-suite numbers elsewhere in this paper remain in the query-head layout, the one every baseline also uses, so all comparisons stay matched; the asymptotics ($O(L)$ peak) were never at issue, and the constant is now demonstrated—not merely argued—to be a layout choice rather than a property of deferral.

H ATTENTION-MATCHING LOSS

Definition. Fix a layer and head. A compression keeps a retained set R of past keys and drops E; let q range over the *observation* queries—in SnapKV the w window queries at the end of the prompt, used as an estimate of the not-yet-seen decode queries. Write the full-attention log-partition and its retained restriction as

$$\log Z(q) = \log \sum_{j \in R \cup E} e^{q^\top k_j/\sqrt{d}}, \qquad \log \hat{Z}(q) = \log \sum_{j \in R} e^{q^\top k_j/\sqrt{d}}. \tag{4}$$



(a) Per-head: KV repeated ×4 before caching. (b) `kv_head`: one shared K|V, mean scores.

Figure 7: **The memory constant is a layout choice** (microbenchmark numbers).

Table 9: **Complexity and measured overhead.** "Extra scoring" is cost on top of the forward pass. *Latency*: per-sample wall clock vs SnapKV, reported short-answer / long-form (hotpotqa+triviaqa, 32 new tokens / gov_report+multi_news, 512), 40 samples each at a pinned generation length, B=128 prototype; the overhead is a prefill-side constant, so it dominates the short regime and amortises in the long one. *TTFT*: vs SnapKV on the 7,762-token microbenchmark—different protocols, so the two columns do not divide into each other. Peak memory is $O(L)$ for all; the prototype's working-set constant and its removal via the native GQA layout (13.1 → 7.4 GiB): App. G. FKV = FullKV (never evicts); the two SKV+DGE rows are ours.

Method	Evicts	Extra scoring	Peak KV	Decode KV	Latency short/long	TTFT
FKV	never	–	$O(L{+}T)$	$O(L{+}t)$	0.89/0.83	0.91×
SLM (Xiao et al., 2024)	prefill end	$O(1)$ (positional)	$O(L)$	$O(B{+}t)$	0.98/1.00	0.91×
H2O (Zhang et al., 2023)	prefill end	$O(L^2 d)^a$	$O(L)$	$O(B{+}t)$	2.15/1.06	3.45×
SKV (Li et al., 2024)	prefill end	$O(Lwd)$	$O(L)$	$O(B{+}t)$	1.00/1.00	1.00×
PKV (Cai et al., 2024)	prefill end	$O(Lwd)$	$O(L)$	$O(B{+}t)$	1.01/1.04	0.93×
LAQ (Wang et al., 2025)	prefill end	$O(mLd)^b$	$O(L)$	$O(B{+}t)$	0.95/0.83	1.42×
AdaKV (Feng et al., 2024)	prefill end	$O(Lwd + L \log L)$	$O(L)$	$O(B{+}t)$	1.30/1.47	—
SKV+DGE$_k$	after k steps	$O(kLd)$	$O(L{+}k)$	$O(L{+}t)_{t\le k}$, then $O(B{+}t)$	1.45/1.02	0.93×
SKV+DGE-W$_k$	after k steps	$O(Lwd)$	$O(L{+}k)$	$O(L{+}t)_{t\le k}$, then $O(B{+}t)$	1.45/1.02	0.92×

[a] H2O accumulates softmax attention over all L prefill queries, unlike the window-based scorers; its deferred variant in Table 12 simply keeps that accumulator running through the k draft rows before evicting.
[b] LAQ's m pseudo queries are produced by an extra forward pass that is discarded, so the scoring *and* the pass sit before the first token; DGE's k draft queries are the answer's own opening. LAQ runs in its own code base, whose SnapKV differs from ours, so its TTFT is anchored on FullKV—unpatched in both, and agreeing to 0.2% across them—rather than divided by the wrong baseline; App. G gives the protocol.

The training-free attention-matching loss is the mean log-partition deficit, which we measure directly at compression time,

$$\mathcal{L}_{\text{match}} \;=\; \mathbb{E}_q\left|\log Z(q) - \log \hat{Z}(q)\right| \;=\; \mathbb{E}_q\left[-\log\bigl(1-\rho_q\bigr)\right], \quad \rho_q = \frac{\sum_{j\in E} e^{q^\top k_j/\sqrt{d}}}{\sum_{j\in R\cup E} e^{q^\top k_j/\sqrt{d}}} \in [0,1), \tag{5}$$

where ρ_q is exactly the softmax mass the observation query places on the evicted keys—the observation-query analogue of the evicted mass $m_t(E)$ of Equation 2. So the loss is a monotone $-\log(1-\cdot)$ transform of the dropped attention mass: it is zero iff the retained set carries all of q's attention and diverges as the evicted mass approaches one. Minimizing it over R under a size budget is the window-query counterpart of FAST's mass-matching objective (Zweiger et al., 2026), the one its NNLS fit solves; SnapKV's top-B selection on the pooled window-query scores is its greedy solution, and the per-token bias β that FAST fits to reweight the retained logits shifts $\log \hat{Z}$ by exactly the recoverable part of (5) (a run with β closes the deficit to $\mathbb{E}_q|\log Z - \log(\hat{Z}\, e^{\beta})|$).

The loss does not predict the downstream deficit. The crucial point for this paper is that (5) is an *observation-query, prefill-time* quantity: it scores how well R reconstructs attention for the window

queries, on the full cache, before any decoding. Equation 3 shows the error that actually accrues is $\sum_t m_t(E_t)\|\Delta\mu_t\|$ over the *real* decode queries q_t, whose weight is set by *when* E is dropped, not by how well the prefill observations were matched. Figure 8 makes this quantitative at scale on Llama-3.1-8B (B=128). Across the sixteen sets the loss varies over a narrow $2\times$ band (0.14–0.28) yet SnapKV's deficit from FullKV ranges from -1.5 to 13.2 points, and the two are essentially uncorrelated (Pearson $r\approx 0.2$, i.e. the loss explains $\sim 4\%$ of the deficit variance): the sets SnapKV hits hardest—Qasper (13.2) and GovReport (11.5)—sit at *low-to-mid* loss, while the highest-loss set (TREC, 0.28) loses only 7. Keeping the identical selection and evicting one decode step later (DGE-W_2) drives the deficit to -0.09 on average (worst case 1.5) at *every* loss. The quantity the *what*-axis optimizes is orthogonal to the error the *when*-axis removes.

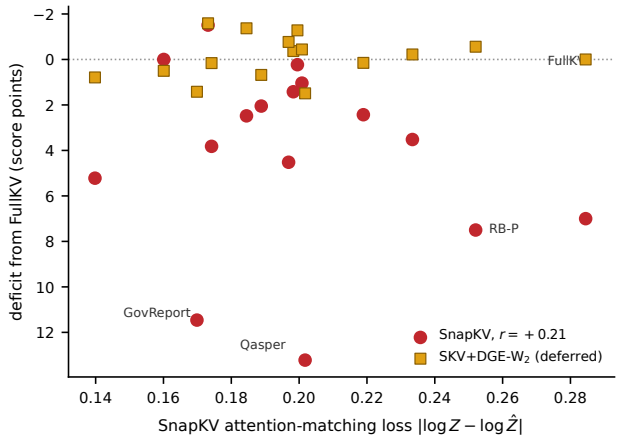


Figure 8: **The attention-matching loss does not predict the downstream deficit.** One point per LongBench set (B=128, Llama-3.1-8B), loss as in (5); DGE-W_2 defers SnapKV's identical selection by one step.

I Confidence intervals and significance

All decoding is greedy, so re-running a configuration reproduces it bit for bit and there is no seed variance to report. The uncertainty that does exist is sampling over the 3,750 LongBench examples. We bootstrap it (2000 resamples): within each dataset we resample examples with replacement, preserving dataset sizes, recompute each dataset score and macro-average over the 16 sets—exactly how the headline number is formed.

Comparisons use a *paired* bootstrap: both methods are scored on the same resample, which is legitimate because every method here is run on identical example sets, and is far sharper than comparing two marginal intervals. The distinction matters: the marginal intervals are ± 1.1 (dominated by which datasets are drawn), so read naively they would suggest *nothing* in Table 2 is resolvable, whereas the paired intervals on differences are $\sim \pm 0.3$.

This sharpens the paper's central claim. On Llama-3.1 the paired test says DGE$_2$ and DGE-W_2 are *not distinguishable at all* on 3,750 examples, while both separate from SnapKV by $\sim$4.1 with intervals nowhere near zero. Qwen2.5-14B is stricter still: at $B \leq 64$ the difference between the two deferred variants *does* resolve, and it favors the timing-only control by 0.4—swapping SnapKV's window scores for real draft queries adds no signal and, on this backbone, costs 0.4—while both variants stay indistinguishable from FullKV at every budget. The *when* axis carries a gain that survives resampling; the *what* axis never produces one, on either backbone.

J Needle-in-a-Haystack retrieval

The LongBench averages aggregate generation quality; Figure 9 isolates *retrieval* under the same tight budget. We run the standard Needle-in-a-Haystack probe on Qwen2.5-7B-Instruct at B=32: the

Table 10: **Paired bootstrap** (2000 resamples of the 3,750 examples): Llama-3.1-8B at B=128 and Qwen2.5-14B at all three budgets (the 14B SKV separations, $+15.8/+9.5/+5.8$, all resolve and are omitted for space). SKV = SnapKV (Li et al., 2024), FKV = FullKV; DGE_2 and DGE-W_2 are the two variants §5 defines. *Resolvable* means the 95% interval excludes zero. Shading here marks the signal-only contrast this paper's attribution turns on, not the "ours" shading of the result tables.

Comparison	Δ	95% CI	Resolvable?
Llama-3.1-8B, B=128			
SKV+DGE_2 − SKV	+4.15	[+3.60, +4.70]	yes
SKV+DGE-W_2 − SKV	+4.11	[+3.61, +4.67]	yes
SKV+DGE_2 − SKV+DGE-W_2	+0.04	[−0.19, +0.26]	**no**
SKV+DGE_2 − FKV	+0.12	[−0.16, +0.44]	no
SKV+DGE-W_2 − FKV	+0.09	[−0.19, +0.38]	no
Qwen2.5-14B, B=32			
SKV+DGE_2 − SKV+DGE-W_2	−0.44	[−0.76, −0.13]	yes (control ahead)
SKV+DGE_2 − FKV	−0.30	[−0.66, +0.06]	no
SKV+DGE-W_2 − FKV	+0.13	[−0.21, +0.47]	no
Qwen2.5-14B, B=64			
SKV+DGE_2 − SKV+DGE-W_2	−0.37	[−0.64, −0.11]	yes (control ahead)
SKV+DGE_2 − FKV	−0.11	[−0.45, +0.21]	no
SKV+DGE-W_2 − FKV	+0.26	[−0.06, +0.58]	no
Qwen2.5-14B, B=128			
SKV+DGE_2 − SKV+DGE-W_2	+0.10	[−0.18, +0.37]	**no**
SKV+DGE_2 − FKV	+0.23	[−0.10, +0.57]	no
SKV+DGE-W_2 − FKV	+0.13	[−0.20, +0.44]	no

needle sentence is planted at 10 depths across 15 context lengths (1k–8k), for 150 probes per method, and a cell is scored by the fraction of the expected answer's words the model recovers.

This is the regime most favorable to prefill-time eviction: the retrieval question is in the prompt, so SnapKV's window queries *can* see which tokens matter, and a query-aware selector should simply keep the needle. It does not. At B=32 SnapKV recovers only 74.7 against FullKV's 98.9—it drops needles across the whole depth×length grid—and H2O, whose accumulated-attention rule is not query-aware, falls to 42.1. Deferring the same SnapKV selection by a single decode step (DGE_2) restores 100.0: every needle, at every depth and length. The 25.3-point recovery is bought without changing which tokens are scored or how, only *when* they are dropped—the Table 2 story reproduced on a task where the *what*-axis was supposed to be sufficient.

K Full result tables

The full 16-dataset breakdown of the main results (summarized as averages in Table 2) is split for readability into the prefill-time baselines of Table 11 and the deferred-eviction rows of Table 12; the complete SnapKV-variant dose grid and the lookahead-query comparison follow. Table 15 aggregates the 16 sets into the six standard LongBench categories, and Table 16 reports, per dataset, how often the answer finishes inside the k-token draft—together with the sample-count check. Every configuration in these tables is run at LongBench's full official sizes (200 per set, 150 for multifieldqa_en, 500 for lcc and repobench-p; 3,750 samples per configuration) with no subsampling; the diagnostic runs behind the grids of App. C are the exception that section records. The only truncation is the standard per-backbone context limit of App. A.

L Additional analysis

Draft-termination fractions and the two regimes. When an answer finishes inside the k-token draft the cache is freed at answer end, no eviction ever fires, and DGE is bit-identical to FullKV—so it is worth asking how much of DGE's gain this path accounts for. Measuring it directly in Table 16, re-tokenizing every saved answer, shows it is a small share: at the headline dose k=2 only 11.2% of

Table 11: **Full per-dataset LongBench results: prefill-time baselines** on Llama-3.1-8B-Instruct ($B \in \{32, 64, 128\}$), Mistral-7B-v0.2, Qwen2.5-7B/14B and Llama-3.2-3B (averages: Table 2; deferred rows: Table 12; Llama-3-8B: Table 7). FullKV is budget-independent; **bold**: best baseline per column per budget block; H2O omitted on Mistral-7B (App. A). *Avg. input*: mean input length in tokens. Scores are each set's official LongBench metric (higher is better; scales differ across columns). FKV = FullKV, SLM = StreamingLLM, SKV = SnapKV, PKV = PyramidKV, H2O—all run by us under App. A's protocol.

	Single-Document QA			Multi-Document QA			Summarization			Few-shot Learning			Synthetic		Code		
Method	NrtvQA	Qasper	MF-en	HotpotQA	2WikiMQA	Musique	GovReport	QMSum	MultiNews	TREC	TriviaQA	SAMSum	PCount	PRe	Lcc	RB-P	Avg.
Avg. input	18409	3619	4559	9151	4887	11214	8734	10614	2113	5177	8209	6258	11141	9289	1235	4206	
Llama-3.1-8B-Instruct, KV Size = Full																	
FKV	24.27	45.79	55.95	48.65	46.90	27.32	33.79	23.26	27.14	69.50	91.49	44.41	10.91	68.50	65.19	56.80	46.24
Llama-3.1-8B-Instruct, KV Size = 32																	
SKV	**20.53**	**25.53**	41.75	43.76	34.70	24.31	17.72	19.06	16.81	41.50	80.78	34.27	**12.50**	65.00	51.63	43.76	35.85
PKV	18.70	25.33	**43.71**	**44.35**	32.71	**25.36**	18.62	**20.73**	17.73	43.50	81.11	**36.46**	**12.50**	67.00	**52.94**	**43.83**	**36.54**
H2O	19.44	25.03	33.12	40.50	29.22	19.71	**23.49**	18.22	**22.39**	**50.50**	**81.49**	21.43	8.92	56.45	44.04	36.64	33.16
SLM	17.81	20.73	29.68	41.93	**37.98**	20.27	15.92	20.03	14.61	36.50	77.05	30.36	**12.50**	**68.50**	48.49	42.62	33.44
Llama-3.1-8B-Instruct, KV Size = 64																	
SKV	20.96	27.38	47.88	44.42	40.24	23.90	20.71	**21.35**	19.68	53.00	**86.83**	**39.74**	12.50	68.00	**56.94**	46.58	39.38
PKV	21.01	27.72	**49.57**	**45.44**	**41.70**	**24.44**	20.94	**21.35**	20.40	55.00	83.39	39.16	**12.75**	68.00	55.59	46.82	**39.58**
H2O	**22.44**	**28.67**	34.96	44.44	33.76	21.19	**25.32**	19.58	**23.35**	**57.50**	85.57	25.62	8.53	64.45	47.11	38.86	36.33
SLM	18.35	21.89	30.78	41.94	40.15	20.25	17.22	20.40	16.16	37.00	79.75	35.23	12.50	**69.00**	54.18	**47.45**	35.14
Llama-3.1-8B-Instruct, KV Size = 128																	
SKV	21.84	32.57	**52.13**	46.60	**44.42**	**25.90**	22.33	22.22	21.92	62.50	**91.26**	**40.89**	12.42	**68.50**	**60.67**	**49.30**	**42.22**
PKV	**22.25**	**33.47**	51.59	**47.34**	44.40	24.66	22.84	**22.32**	21.89	**63.50**	86.52	40.59	12.25	68.00	57.08	47.60	41.64
H2O	22.20	32.67	40.42	44.72	35.95	20.47	**27.08**	20.00	**24.40**	61.50	88.22	28.62	8.60	62.20	50.96	41.63	38.10
SLM	19.03	21.63	31.88	42.17	40.92	20.64	18.82	20.60	18.19	41.00	82.89	38.14	**12.50**	68.00	59.02	48.69	36.51
Mistral-7B-Instruct-v0.2, KV Size = Full																	
FKV	22.06	29.15	47.59	37.57	21.81	18.36	31.42	23.95	26.75	71.00	86.23	42.90	3.73	87.40	57.12	54.51	41.35
Mistral-7B-Instruct-v0.2, KV Size = 128																	
SKV	16.66	18.90	40.12	26.59	15.95	12.80	**20.59**	**21.89**	**21.68**	**67.00**	**85.06**	**40.35**	2.43	62.97	**52.20**	**47.31**	34.53
PKV	**17.21**	**21.14**	**40.80**	**27.94**	**16.14**	**13.39**	20.44	21.81	21.47	66.00	83.51	39.76	**3.48**	**67.44**	51.61	46.42	**34.91**
SLM	13.67	10.92	24.57	20.17	14.73	10.03	15.00	19.27	16.79	44.00	79.92	37.46	2.62	27.05	51.44	45.95	27.10
Mistral-7B-Instruct-v0.2, KV Size = 64																	
SKV	14.63	16.08	34.28	22.15	11.78	11.53	18.41	**21.31**	19.28	53.00	**82.32**	**38.10**	1.95	63.65	48.56	**44.57**	31.35
PKV	**15.40**	**17.00**	**35.95**	**25.28**	**14.26**	**12.19**	**18.45**	20.88	**19.50**	**54.00**	81.97	37.13	1.76	**63.79**	**48.58**	43.17	**31.83**
SLM	12.05	10.21	23.52	19.11	13.90	9.36	13.43	19.48	14.65	39.50	78.84	35.52	**2.25**	29.33	47.01	42.73	25.68
Qwen2.5-7B-Instruct, KV Size = Full																	
FKV	23.16	44.54	53.01	49.48	47.39	25.20	32.34	21.90	24.15	69.50	91.34	45.25	7.00	66.50	61.82	66.11	45.54
Qwen2.5-7B-Instruct, KV Size = 32																	
SKV	**16.63**	26.15	34.51	37.83	28.80	21.27	16.36	**17.34**	12.69	38.00	82.71	35.14	**7.00**	43.00	43.32	44.19	31.56
PKV	15.68	**27.01**	**35.28**	**40.11**	31.88	**23.87**	17.05	17.29	13.18	41.50	**82.77**	**36.00**	**7.00**	**44.00**	**45.01**	**45.46**	**32.69**
H2O	16.07	21.43	24.86	37.09	23.74	19.43	**23.10**	16.69	**20.21**	**49.50**	76.91	21.37	**7.00**	31.75	39.16	36.79	29.07
SLM	11.19	19.74	24.26	35.11	**36.50**	15.88	12.87	16.97	10.72	35.00	75.06	30.36	**7.00**	26.00	38.18	38.87	27.11
Qwen2.5-7B-Instruct, KV Size = 64																	
SKV	**17.79**	**30.47**	**40.74**	**45.69**	**38.42**	**23.24**	18.97	**18.70**	15.99	45.00	**85.27**	**39.28**	**7.00**	**61.50**	**48.55**	**51.73**	**36.77**
PKV	15.47	26.41	32.00	36.64	27.01	19.08	17.26	17.14	13.54	42.50	79.33	33.45	**7.00**	37.00	44.42	43.59	30.74
H2O	15.99	28.45	29.08	40.72	30.02	22.44	**25.21**	17.86	**21.34**	**56.50**	82.07	25.40	**7.00**	30.75	43.10	38.51	32.15
SLM	12.11	25.24	25.17	36.49	38.16	16.12	14.30	17.35	12.12	37.50	79.90	36.69	**7.00**	16.00	44.09	44.22	28.90
Qwen2.5-7B-Instruct, KV Size = 128																	
SKV	**21.02**	**35.77**	**45.71**	44.71	42.94	**23.49**	21.65	**19.71**	18.49	57.50	86.67	**41.90**	**7.00**	**64.00**	**53.91**	**55.24**	**39.98**
PKV	20.44	32.72	43.56	**46.37**	**43.21**	22.34	20.21	18.85	16.53	53.00	**86.73**	41.14	**7.00**	60.00	50.98	51.01	38.38
H2O	19.15	34.21	32.25	40.76	36.40	22.52	**27.45**	18.95	**22.07**	**63.50**	86.68	28.84	**7.00**	40.56	46.55	43.60	35.66
SLM	13.13	24.40	24.76	37.04	37.79	14.75	16.29	17.35	14.38	41.00	82.64	40.62	**7.00**	14.00	48.39	49.52	30.19
Qwen2.5-14B-Instruct, KV Size = Full																	
FKV	27.03	45.10	51.82	52.24	57.59	30.10	29.07	21.92	22.78	75.00	89.75	46.99	7.04	67.00	62.90	50.40	46.05
Qwen2.5-14B-Instruct, KV Size = 32																	
SKV	17.48	18.25	27.87	36.98	28.21	21.59	14.15	16.49	12.48	36.50	79.13	35.07	10.16	40.38	48.29	36.03	29.94
PKV	**19.98**	**24.49**	**36.23**	**43.99**	38.96	**25.01**	17.04	**18.10**	15.25	45.50	**84.76**	**39.61**	**10.37**	48.83	**52.04**	**41.03**	**35.07**
H2O	17.45	13.25	24.25	40.93	29.11	22.47	**20.73**	15.92	**19.42**	**56.50**	79.38	24.34	6.16	23.92	45.09	32.85	29.49
SLM	12.37	10.36	23.08	31.98	**43.75**	16.74	12.97	17.04	11.47	35.50	73.44	30.92	9.42	**50.33**	43.10	35.56	28.63
Qwen2.5-14B-Instruct, KV Size = 64																	
SKV	**21.59**	**27.85**	**38.97**	44.92	40.97	**26.15**	17.48	**18.55**	15.88	49.00	**84.48**	**41.57**	**9.68**	**50.08**	**54.32**	**41.21**	**36.42**
PKV	20.01	26.28	35.28	**45.28**	40.23	23.82	16.83	18.06	15.35	48.00	79.87	39.57	8.85	46.83	51.83	38.07	34.64
H2O	20.68	20.17	26.51	42.61	33.36	23.89	**22.77**	17.32	**20.82**	**64.00**	80.67	26.49	6.36	20.42	47.93	33.94	31.75
SLM	14.72	15.40	22.20	33.13	**44.69**	17.26	14.21	17.34	12.86	38.00	77.50	38.06	5.80	46.75	49.35	39.09	30.40
Qwen2.5-14B-Instruct, KV Size = 128																	
SKV	20.98	30.87	41.50	**48.77**	51.74	28.03	19.87	**19.88**	17.86	64.50	**86.75**	**43.18**	**9.00**	**63.75**	**56.52**	**44.95**	**40.51**
PKV	**22.05**	**31.52**	**42.34**	47.97	**52.30**	**28.77**	19.44	19.86	17.76	67.00	84.79	42.98	8.52	63.00	55.19	43.06	40.41
H2O	21.57	27.46	31.04	44.81	40.74	24.20	**24.57**	18.09	**21.00**	**68.00**	82.84	29.55	6.71	20.08	52.01	36.81	34.34
SLM	13.94	18.73	25.14	33.67	43.91	18.04	15.70	18.03	14.63	40.50	80.31	40.29	5.37	47.75	52.08	41.08	31.82
Llama-3.2-3B-Instruct, KV Size = Full																	
FKV	21.50	40.41	50.26	44.97	39.07	18.09	32.32	23.52	25.83	69.50	88.22	42.81	4.50	67.50	54.36	56.41	42.45
Llama-3.2-3B-Instruct, KV Size = 128																	
SKV	16.65	24.91	46.11	43.99	36.51	**15.77**	22.03	20.52	20.02	**65.00**	**88.01**	**38.22**	4.50	**66.50**	50.63	**49.77**	**38.07**
PKV	**19.36**	**25.19**	**47.55**	**44.98**	**37.33**	14.13	21.67	**20.94**	19.91	63.50	85.01	37.77	4.50	66.00	47.93	47.15	37.68
H2O	17.65	24.05	33.31	41.84	31.50	13.20	**25.21**	18.43	**23.47**	63.00	84.61	25.38	**4.84**	54.00	45.38	37.78	33.98
SLM	16.11	19.93	26.15	37.43	32.33	12.58	17.70	19.41	17.28	42.50	78.65	36.91	4.50	34.00	**50.86**	47.44	30.86
Llama-3.2-3B-Instruct, KV Size = 64																	
SKV	**18.08**	**24.16**	41.94	**43.44**	31.79	**15.06**	20.21	**20.27**	18.12	56.50	**85.13**	**36.35**	4.50	61.00	**48.26**	**45.45**	**35.64**
PKV	17.55	22.18	**42.05**	40.77	27.63	14.07	19.39	19.44	17.43	49.50	81.20	34.41	4.50	**65.00**	41.92	38.89	33.50
H2O	16.94	20.43	27.94	38.05	26.30	13.26	**23.20**	17.84	**22.10**	**61.00**	80.80	21.99	**4.63**	60.00	41.25	34.63	31.90
SLM	15.27	20.62	25.50	36.52	**32.40**	13.56	16.52	19.56	14.68	37.50	76.14	33.36	4.50	33.00	46.02	43.63	29.30

Table 12: **Full per-dataset LongBench results: deferred eviction** (baselines in Table 11). Each +DGE$_2$ row defers its base evictor's eviction one full-cache decode step (budget in the row label); each SnapKV anchor precedes its deferred counterpart. Shaded rows: ours; **bold**: best deferred row per column per backbone (FKV and anchors excluded). Rows: FKV = FullKV; SKV = SnapKV (Li et al., 2024); PKV = PyramidKV (Cai et al., 2024); H2O (Zhang et al., 2023); SLM = StreamingLLM (Xiao et al., 2024). Every +DGE$_2$ row is ours.

	Single-Document QA			Multi-Document QA			Summarization			Few-shot Learning			Synthetic		Code		
Method	NrtvQA	Qasper	MF-en	HotpotQA	2WikiMQA	Musique	GovReport	QMSum	MultiNews	TREC	TriviaQA	SAMSum	PCount	PRe	Lcc	RB-P	Avg.
Avg. input	18409	3619	4559	9151	4887	11214	8734	10614	2113	5177	8209	6258	11141	9289	1235	4206	
Llama-3.1-8B-Instruct																	
FKV	24.27	45.79	55.95	48.65	46.90	27.32	33.79	23.26	27.14	69.50	91.49	44.41	10.91	68.50	65.19	56.80	46.24
SKV (B=32)	20.53	25.53	41.75	43.76	34.70	24.31	17.72	19.06	16.81	41.50	80.78	34.27	12.50	65.00	51.63	43.76	35.85
SKV+DGE$_2$ (B=32)	24.13	45.03	54.96	48.58	48.08	27.50	32.53	23.32	26.67	**69.50**	92.65	44.74	**12.50**	68.17	65.61	58.34	46.39
SKV (B=64)	20.96	27.38	47.88	44.42	40.24	23.90	20.71	21.35	19.68	53.00	86.83	39.74	12.50	68.00	56.94	46.58	39.38
SKV+DGE$_2$ (B=64)	23.85	45.20	54.65	48.24	48.21	27.95	32.52	23.18	26.49	**69.50**	92.48	44.97	**12.50**	**69.00**	65.26	57.74	46.36
SKV (B=128)	21.84	32.57	52.13	46.60	44.42	25.90	22.33	22.22	21.92	62.50	91.26	40.89	12.42	68.50	60.67	49.30	42.22
SKV+DGE$_2$ (B=128)	22.97	44.45	54.49	**48.89**	**48.70**	27.91	**32.55**	**23.73**	26.36	**69.50**	92.48	45.09	**12.50**	**69.00**	**65.78**	57.46	46.37
PKV+DGE$_2$ (B=32)	24.40	**45.70**	54.32	47.98	48.21	27.40	32.47	23.55	26.54	**69.50**	92.21	**45.67**	**12.50**	**69.00**	65.48	57.81	**46.42**
H2O+DGE$_2$ (B=32)	24.56	45.22	54.43	48.17	48.45	27.86	32.06	23.03	26.87	**69.50**	92.71	43.88	**12.50**	68.58	65.26	58.07	45.70
SLM+DGE$_2$ (B=32)	23.67	45.30	**55.14**	47.95	47.11	**27.99**	31.66	22.92	26.14	**69.50**	92.56	44.76	**12.50**	68.50	65.49	**58.48**	46.23
PKV+DGE$_2$ (B=128)	22.62	44.71	54.55	48.58	48.63	27.40	32.13	23.41	26.22	**69.50**	92.35	44.67	**12.50**	68.75	65.64	57.75	46.21
H2O+DGE$_2$ (B=128)	**24.85**	44.70	54.05	48.42	48.26	26.96	32.40	23.31	**27.01**	**69.50**	**92.84**	43.34	**12.50**	60.83	65.44	57.29	45.73
SLM+DGE$_2$ (B=128)	23.87	44.86	53.87	47.54	45.91	26.98	31.38	22.97	25.83	**69.50**	92.56	45.49	12.25	67.00	65.54	57.69	45.83
Mistral-7B-Instruct-v0.2																	
FKV	22.06	29.15	47.59	37.57	21.81	18.36	31.42	23.95	26.75	71.00	86.23	42.90	3.73	87.40	57.12	54.51	41.35
SKV (B=128)	16.66	18.90	40.12	26.59	15.95	12.80	20.59	21.89	21.68	67.00	85.06	40.35	2.43	62.97	52.20	47.31	34.53
SKV+DGE$_2$ (B=128)	**21.60**	**28.68**	44.36	**38.13**	**21.53**	**18.57**	30.27	22.68	**26.46**	**71.00**	85.40	**43.40**	**5.34**	**85.21**	**57.80**	**55.51**	**41.00**
SKV (B=64)	14.63	16.08	34.28	22.15	11.78	11.53	18.41	21.31	19.28	53.00	82.32	38.10	1.95	63.65	48.56	44.57	31.35
SKV+DGE$_2$ (B=64)	21.04	28.52	**45.57**	37.87	21.01	18.28	**30.46**	**22.86**	26.17	**71.00**	**85.65**	43.17	4.68	85.02	57.76	55.39	40.90
Qwen2.5-7B-Instruct																	
FKV	23.16	44.54	53.01	49.48	47.39	25.20	32.34	21.90	24.15	69.50	91.34	45.25	7.00	66.50	61.82	66.11	45.54
SKV (B=32)	16.63	26.15	34.51	37.83	28.80	21.27	16.36	17.34	12.69	38.00	82.71	35.14	7.00	43.00	43.32	44.19	31.56
SKV+DGE$_2$ (B=32)	22.75	43.86	50.34	50.22	47.06	24.70	31.40	21.14	24.08	**69.50**	91.21	44.62	**7.00**	66.50	60.71	**65.69**	45.05
SKV (B=64)	17.79	30.47	40.74	45.69	38.42	23.24	18.97	18.70	15.99	45.00	85.27	39.28	7.00	61.50	48.55	51.73	36.77
SKV+DGE$_2$ (B=64)	22.56	43.93	**51.69**	49.76	**47.95**	25.72	31.60	21.69	23.95	**69.50**	91.71	44.78	**7.00**	**67.00**	60.78	65.52	**45.32**
SKV (B=128)	21.02	35.77	45.71	44.71	42.94	23.49	21.65	19.71	18.49	57.50	86.67	41.90	7.00	64.00	53.91	55.24	39.98
SKV+DGE$_2$ (B=128)	22.61	43.93	51.24	**50.38**	47.90	25.96	**31.86**	21.51	24.15	**69.50**	91.24	**45.18**	**7.00**	66.50	60.65	65.59	**45.32**
PKV+DGE$_2$ (B=128)	22.10	**44.30**	50.17	50.12	47.49	24.91	31.11	**21.91**	23.79	**69.50**	91.19	**45.18**	**7.00**	66.50	59.64	65.64	45.03
H2O+DGE$_2$ (B=128)	21.96	43.05	49.37	48.90	46.83	25.47	30.88	21.85	24.20	**69.50**	**91.92**	43.88	**7.00**	65.50	59.66	64.82	44.67
SLM+DGE$_2$ (B=128)	**24.21**	43.72	48.88	49.45	46.42	**26.05**	30.95	21.29	**24.52**	**69.50**	91.45	43.34	**7.00**	64.00	**61.47**	65.46	44.86
Qwen2.5-14B-Instruct																	
FKV	27.03	45.10	51.82	52.24	57.59	30.10	29.07	21.92	22.78	75.00	89.75	46.99	7.04	67.00	62.90	50.40	46.05
SKV (B=32)	17.48	18.25	27.87	36.98	28.21	21.59	14.15	16.49	12.48	36.50	79.13	35.07	10.16	40.38	48.29	36.03	29.94
SKV+DGE (B=32)	25.55	44.66	51.00	51.18	57.15	30.24	29.08	21.96	22.81	**75.00**	90.73	46.96	8.20	62.00	63.06	52.31	45.74
SKV+DGE-W$_2$ (B=32)	**27.29**	44.90	51.17	51.46	57.70	30.06	29.34	21.90	22.65	**75.00**	**91.28**	45.79	**8.93**	66.00	63.67	51.71	46.18
SKV (B=64)	21.59	27.85	38.97	44.92	40.97	26.15	17.48	18.55	15.88	49.00	84.48	41.57	9.68	50.08	54.32	41.21	36.42
SKV+DGE$_2$ (B=64)	25.57	43.80	50.82	**51.77**	57.07	30.24	**29.48**	22.27	**22.89**	**75.00**	90.43	46.94	8.23	65.25	63.48	51.80	45.94
SKV+DGE-W$_2$ (B=64)	26.38	**45.64**	**51.37**	51.12	**57.81**	30.11	29.36	21.72	22.62	**75.00**	90.82	46.14	8.70	**67.00**	64.54	52.56	**46.31**
SKV (B=128)	20.98	30.87	41.50	48.77	51.74	28.03	19.87	19.88	17.86	64.50	86.75	43.18	9.00	63.75	56.52	44.95	40.51
SKV+DGE$_2$ (B=128)	25.86	44.79	50.77	51.59	57.26	**30.64**	29.43	**22.66**	22.64	**75.00**	90.48	**47.51**	8.71	65.75	**64.76**	52.52	46.27
SKV+DGE-W$_2$ (B=128)	25.91	45.42	50.28	51.45	57.78	30.38	29.36	21.83	22.49	**75.00**	90.17	46.90	8.39	66.50	64.03	**52.94**	46.18
Llama-3.2-3B-Instruct																	
FKV	21.50	40.41	50.26	44.97	39.07	18.09	32.32	23.52	25.83	69.50	88.22	42.81	4.50	67.50	54.36	56.41	42.45
SKV (B=128)	16.65	24.91	46.11	43.99	36.51	15.77	22.03	20.52	20.02	65.00	88.01	38.22	4.50	66.50	50.63	49.77	38.07
SKV+DGE$_2$ (B=128)	**22.19**	**40.19**	**51.60**	**45.46**	**38.87**	18.34	**31.56**	22.48	**25.18**	**69.50**	**88.26**	**43.09**	**4.50**	**66.50**	**55.99**	57.31	**42.56**
SKV (B=64)	18.08	24.16	41.94	43.44	31.79	15.06	20.21	20.27	18.12	56.50	85.13	36.35	4.50	61.00	48.26	45.45	35.64
SKV+DGE$_2$ (B=64)	22.13	39.83	51.11	45.29	38.79	**18.36**	31.34	**22.62**	25.10	**69.50**	87.81	42.79	**4.50**	65.50	55.75	**57.33**	42.36

Table 13: **SnapKV-variant ablation** on Llama-3.1-8B at B=128 (full dose grid) and 32 (sampled). **Bold**: best compression method per column per budget block (FullKV excluded). Axes: draft length k and eviction signal (DGE: the k real decode queries; DGE-W: SnapKV's unmodified window scores, i.e. timing only). On this backbone both signals are flat from k=2: every point is ≥ 3.9 above SKV and within 0.16 of FullKV. FKV = FullKV, SKV = SnapKV (Li et al., 2024); every SKV+DGE and SKV+DGE-W row is ours.

	Single-Document QA			Multi-Document QA			Summarization			Few-shot Learning			Synthetic		Code		
Method	NrtvQA	Qasper	MF-en	HotpotQA	2WikiMQA	Musique	GovReport	QMSum	MultiNews	TREC	TriviaQA	SAMSum	PCount	PRe	Lcc	RB-P	Avg.
KV Size = Full (budget-independent reference)																	
FKV	24.27	45.79	55.95	48.65	46.90	27.32	33.79	23.26	27.14	69.50	91.49	44.41	10.91	68.50	65.19	56.80	46.24
KV Size = 32 (sampled dose grid: k=2, 8, 32)																	
SKV	20.53	25.53	41.75	43.76	34.70	24.31	17.72	19.06	16.81	41.50	80.78	34.27	**12.50**	65.00	51.63	43.76	35.85
SKV+DGE$_2$	24.13	45.03	54.96	48.58	**48.08**	27.50	**32.53**	23.32	**26.67**	**69.50**	**92.65**	44.74	**12.50**	68.17	65.61	**58.34**	46.39
SKV+DGE$_8$	24.28	**45.80**	55.23	**48.68**	47.46	**27.60**	31.96	**23.49**	26.42	**69.50**	91.18	44.86	11.33	**68.50**	65.20	56.67	46.14
SKV+DGE$_{32}$	24.27	45.79	55.89	48.65	46.85	27.32	31.82	23.31	26.55	**69.50**	91.49	44.42	10.91	**68.50**	65.26	56.82	46.08
SKV+DGE-W$_2$	**24.47**	45.32	**56.39**	48.18	47.76	27.15	31.98	23.23	26.49	**69.50**	92.53	**45.11**	**12.50**	68.00	**66.21**	57.55	**46.40**
KV Size = 128 (full dose grid)																	
SKV	21.84	32.57	52.13	46.60	44.42	25.90	22.33	22.22	21.92	62.50	91.26	40.89	12.42	68.50	60.67	49.30	42.22
SKV+DGE$_2$	22.97	44.45	54.49	**48.89**	**48.70**	**27.91**	32.55	23.73	26.36	**69.50**	92.48	45.09	**12.50**	**69.00**	65.78	**57.46**	**46.37**
SKV+DGE$_4$	23.95	44.42	55.86	48.65	48.24	27.44	32.49	23.47	26.75	**69.50**	92.29	44.97	12.17	68.50	65.22	56.77	46.29
SKV+DGE$_8$	24.18	45.32	55.70	48.63	47.23	27.42	32.26	23.35	26.65	**69.50**	91.25	44.68	11.33	68.50	65.19	56.73	46.12
SKV+DGE$_{16}$	**24.31**	45.76	55.63	48.72	46.86	27.34	**32.56**	23.92	26.62	**69.50**	91.56	44.93	10.91	68.50	65.23	56.72	46.19
SKV+DGE$_{32}$	24.27	45.81	55.84	48.65	46.85	27.32	32.36	23.42	**26.93**	**69.50**	91.49	44.28	10.91	68.50	65.13	56.82	46.13
SKV+DGE-W$_2$	24.12	44.30	55.79	47.97	48.27	27.69	32.37	23.70	26.35	**69.50**	**92.77**	44.63	**12.50**	68.00	**65.96**	57.36	46.33
SKV+DGE-W$_4$	23.64	45.30	**56.13**	48.58	47.69	27.49	32.20	**24.04**	26.05	**69.50**	92.20	**45.40**	12.17	68.50	65.21	56.74	46.30
SKV+DGE-W$_8$	24.25	46.01	55.99	48.62	47.23	27.45	31.79	23.82	26.13	**69.50**	91.28	44.95	11.17	68.50	65.14	56.42	46.14
SKV+DGE-W$_{16}$	24.29	**46.20**	55.86	48.72	46.86	27.34	31.55	**24.04**	26.62	**69.50**	91.48	44.80	11.21	68.50	65.04	56.69	46.17

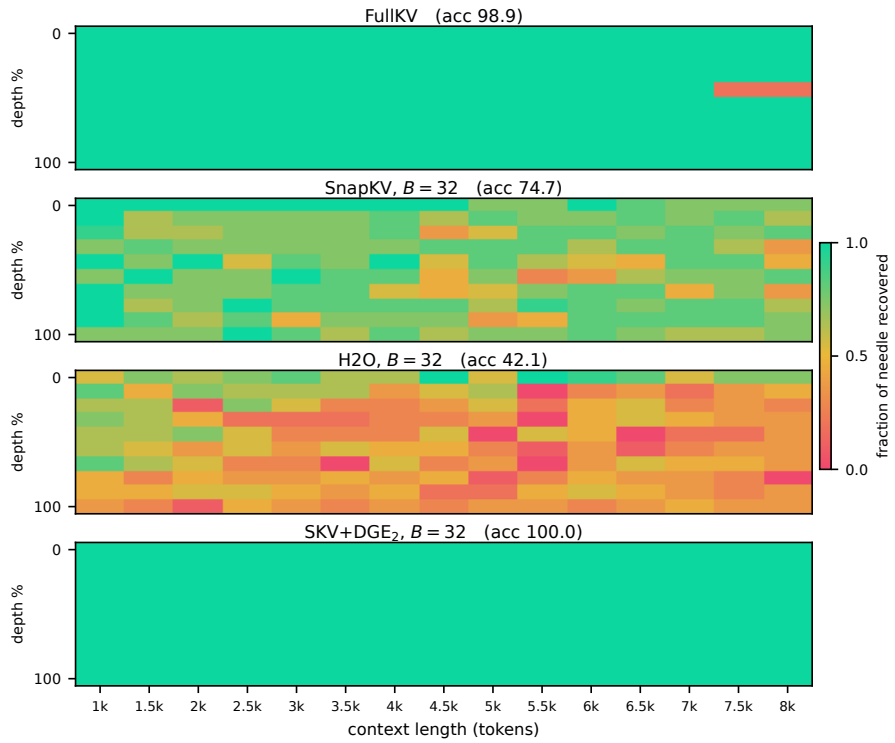


Figure 9: **Needle-in-a-Haystack at B=32 (Qwen2.5-7B).** 150 probes per panel, with the retrieval question visible in the prompt at compression time.

Table 14: **Lookahead-query eviction vs deferred eviction** (Llama-3.1-8B, 16 LongBench sets, matched protocol; LAQ run in its own code). LAQ (Wang et al., 2025) sharpens the *what* axis with m pseudo future queries but still evicts at prefill end. DGE-W keeps SnapKV's window scores and only defers eviction one decode step; DGE also swaps in the real decode queries. **Bold**: best per budget (FullKV excluded).

	Eviction time / signal	B=64	B=128
FullKV	— (budget-independent)	46.24	46.24
SnapKV (Li et al., 2024)	prefill end / window	39.38	42.22
Lookahead Q-Cache (Wang et al., 2025) — prefill end / pseudo-query scores (m lookahead queries)			
LAQ (m=4)	prefill end / pseudo	42.75	43.81
LAQ (m=8)	prefill end / pseudo	43.11	44.13
LAQ (m=16)	prefill end / pseudo	43.17	44.20
LAQ (m=32)	prefill end / pseudo	—	44.10
Deferred eviction (ours) — one full-cache decode step, then evict			
SnapKV+DGE-W$_2$	**decode step 2** / window	46.20	46.33
SnapKV+DGE$_2$	**decode step 2** / draft queries	**46.36**	**46.37**

the 3,750 samples terminate in the draft, and termination is confined to short-answer QA (5–40%) and passage_count (92.5%), with Summarization, Few-shot and Code at 0%. Even at k=16 the overall figure is 34.0%. The gain therefore cannot be a draft-termination artifact: ∼89% of samples at k=2 genuinely decode on the compressed cache, and DGE$_2$ still matches FullKV (46.37 vs 46.24). The cleanest case is long-form generation, where the answer far exceeds k: on gov_report (512-token generations, 0% termination) ∼99% of tokens decode on the compressed cache, so the recovery (22.3→32.6 ROUGE at B=128) is best explained by trajectory anchoring.

Composition across retention rules. The single full-cache decode step transfers to the three other base evictors we test, not just SnapKV: at B=128, PKV+DGE$_2$ reaches 46.21 (from PyramidKV's

Table 15: **LongBench per-category averages** (Llama-3.1-8B; FullKV is budget-independent). SnapKV's damage concentrates in Few-shot Learning, Code, Single-Doc QA and Summarization (10–16 points at B=32), while Synthetic—essentially retrieval, so the window queries see what matters—already matches FullKV. One deferred decode step restores *every* category at both budgets; at B=32 it matches FullKV category-for-category where SnapKV is 10.4 points down overall and as much as 16.3 on Few-shot Learning. **Bold** marks the best compression method per column within each budget block. FKV = FullKV, SKV = SnapKV (Li et al., 2024); the SKV+DGE and SKV+DGE-W rows are ours.

	Method	Single-Doc QA	Multi-Doc QA	Summar-ization	Few-shot Learning	Synthetic	Code	All
	FKV	42.00	40.96	28.06	68.47	39.70	60.99	46.24
B=32	SKV	29.27	34.26	17.86	52.18	38.75	47.70	35.85
	SKV+DGE$_2$	**41.37**	**41.39**	**27.51**	**68.96**	**40.34**	**61.98**	**46.39**
B=128	SKV	35.51	38.97	22.16	64.88	40.46	54.98	42.22
	SKV+DGE-W$_2$	**41.40**	41.31	27.47	68.97	40.25	**61.66**	46.33
	SKV+DGE$_2$	40.64	**41.83**	**27.55**	**69.02**	**40.75**	61.62	**46.37**

Table 16: **Draft-termination fractions and sample accounting** (Llama-3.1-8B, B=128). *term$_k$* = share of answers finishing within the k-token draft, where DGE is bit-identical to FullKV. All 16 sets at full official sizes.

Category	Dataset	n	term$_{k=2}$	term$_{k=16}$	median answer (tokens)
Single-Doc QA	narrativeqa	200	5.0	87.0	7
	qasper	200	12.5	48.0	17
	multifieldqa_en	150	13.3	58.7	11
Multi-Doc QA	hotpotqa	200	39.5	89.5	3
	2wikimqa	200	24.0	96.0	4
	musique	200	26.0	81.5	4
Summarization	gov_report	200	0.0	0.0	415
	qmsum	200	0.0	1.0	87
	multi_news	200	0.0	0.0	471
Few-shot	trec	200	0.0	0.0	64
	triviaqa	200	0.0	0.5	32
	samsum	200	0.0	0.5	128
Synthetic	passage_count	200	92.5	98.0	1
	passage_retrieval_en	200	0.0	91.0	3
Code	lcc	500	0.0	0.0	64
	repobench-p	500	0.0	0.0	64
Overall		**3750**	**11.2**	**34.0**	—

41.64), H2O+DGE$_2$ reaches 45.73 (+7.6), and StreamingLLM+DGE$_2$ reaches 45.83—a +9.3 jump for a rule that keeps four sinks and a recency window *without any attention scores at all*. The timing-only compositions, run in the same sweep but not tabulated, land within 0.35 of their draft-scored counterparts (PKV+DGE-W$_2$ 46.38, H2O+DGE-W$_2$ 45.39), confirming that the recovery is a property of the eviction *event* and not of any particular retention rule. The recovery also holds at the tightest budget: at B=32, PKV/H2O/StreamingLLM+DGE$_2$ reach 46.42/45.70/46.23, a +12.8 jump for StreamingLLM from 33.44. The same composition holds on a second backbone: on Qwen2.5-7B at B=128, PKV+DGE$_2$ reaches 45.03 (from PyramidKV's 38.38, +6.7), H2O+DGE$_2$ reaches 44.67 (from 35.66, +9.0), and StreamingLLM+DGE$_2$ reaches 44.86 (+14.7)—all within 0.9 of Qwen's FullKV 45.54, per Table 12.

Lookahead-query dose saturation. LAQ's pseudo-query dose helps but saturates ∼2 points below a pure timing change. At B=128 the m=4/8/16/32 averages are 43.81/44.13/44.20/44.10—the curve peaks at m=16 and is flat (even slightly down) at m=32, so spending more pseudo queries does not close the gap to FullKV (46.24) or to DGE-W$_2$ (46.33). The same ordering holds at B=64. The deficit against the timing-only control also *widens* as the budget shrinks—1.0–2.1 points at B=128 (Qwen2.5-14B: 1.7) vs 3.0–3.9 at B=64 and 5.8–9.1 at B=32—so tighter budgets make *when* matter more, not less. Sharpening the *what*-signal has a ceiling that a single deferral step clears outright, as Table 14 shows.

What LAQ's pseudo queries are actually worth. The dose curve says the *what*-signal saturates; this probe says why. We replay LAQ's two stages inside the selection harness of Section B—compress the prefill cache to B with SnapKV (window 32), generate m=16 tokens on that compressed cache keeping their post-RoPE queries, then score the full prompt keys with the last $8 + m$ rows, exactly as our matched runs configure LAQ—and score the resulting selection against the same held-out decode attention every other rung of the ladder is measured on. LAQ's pseudo queries buy +1.2% covered mass over SnapKV's window estimate at B=128 (+1.3% at B=64; 12 prompts), and −3.1% when the window rows are removed and the pseudo queries score alone. The real draft queries DGE captures at the same dose (k=16, i.e. DGE$_{16}$) are worth +18.7% on the same 12 prompts, and the oracle +30.5%—the two rungs Table 1(b) reports as +16.5/+31.8% on its own diagnostic split, re-measured here so all three are scored on identical prompts. Generating the lookahead on a cache already compressed to B costs essentially all of the signal the draft was supposed to provide. The pseudo continuation also leaves the real answer early—it agrees with the full-cache answer for 5.1 tokens on average, and on `gov_report`, where the compression gap lives, it diverges after one to eight. This is the measurement behind LAQ's position in Figure 2(a): it improves its own objective by almost nothing, yet still gains +2.0 LongBench—a second, opposite reading of the miscalibration the analysis reports, and the reason the timing-only control beside it, with no signal change at all, gains twice as much.

The minimal steering dose is capability-, not scale-, dependent. The dose k that recovers FullKV is backbone-dependent: Llama-3.1-8B (both B=32 and 128), Mistral-7B, Qwen2.5-7B, and the smaller Llama-3.2-3B all saturate at k=2, whereas the older, shorter-context Llama-3-8B climbs gradually in Figure 4(a) and needs k≈8, analyzed in App. D. That the 3B model still needs only k=2 while an 8B model needs k≈8 shows the dose tracks the decisiveness of a backbone's opening tokens (how quickly its instruction tuning commits to a trajectory), not parameter count. Scaling up does not shrink the problem either: on Qwen2.5-14B—the largest backbone we run—SnapKV at B=32 posts its largest LongBench-average collapse in the paper (29.94 vs FullKV 46.05, −16.1), and the same k=2 deferral recovers 45.74 in Table 12, within 0.31 of FullKV.

PyramidKV's non-monotone Qwen columns are a discrete-schedule artifact. In Table 2 PKV on Qwen2.5-7B scores *lower* at B=64 (30.74) than at B=32 (32.69), and Qwen2.5-14B repeats the inversion (34.64 vs 35.07); the runs reproduce bit-identically under greedy decoding, so it is systematic, and the cause is arithmetic. PKV allocates the per-layer past budget on a linear ramp from $\max = 2c - \lfloor c/20 \rfloor$ down by $\text{steps} = \lfloor (\max - \min)/(L-1) \rfloor$ per layer, where $c = B - w$ and L is the layer count. With Qwen's L=28 and B=64 (c=56) the floor lands on steps=4, so the deepest layers keep only $[14, 10, 6, 2]$ past tokens—the last layer retains 2 tokens beside the 8-token window—whereas at B=32 (steps=1) the same layers keep $[23, 22, 21, 20]$: the *smaller* budget leaves *more* cache in the deep layers, inverting the column. Qwen2.5-14B (L=48) is the extreme case: at B=32 the floor collapses to steps=0, so the ramp degenerates and *every* layer keeps the flat max=47 past tokens—nearly twice the nominal budget—while at B=64 (steps=2) the deepest layer keeps only 16. On L=32 backbones the floor lands on steps=3 at B=64 (last layer 17), so no inversion appears. Deferral is indifferent to this: SKV+DGE$_2$ is budget-invariant on Qwen (7B: 45.05/45.32/45.32; 14B: 45.74/45.94/46.27), so the artifact affects only the prefill-time schedule, not the timing result.

Per-dataset view of timing vs pseudo queries. Figure 4(b) plots, per dataset at B=128, SnapKV's score against each method's; points above y=x beat SnapKV. Both timing methods (SKV+DGE$_2$, SKV+DGE-W$_2$) sit well above the diagonal and gain most on the hard, low-SnapKV summarization and code sets (clearing LAQ by +8.8 on GovReport), which they lift to FullKV level; Looka-

head Q-Cache (m=16, prefill-time) hugs the diagonal on exactly those sets, so its improvement is concentrated where SnapKV was already adequate.

M What the Diagnostics Ruled Out

Each alternative below was settled by the held-out probes of Section B in under a day of compute, before any LongBench run. We log them because the *pattern* is the point: every entry improves an in-window quantity and fails on held-out decode behavior—the same miscalibration §4 measures.

- **Repeat-prefill selection.** Re-forwarding the context to harvest "future-like" queries (the KVzip recipe, as a selection signal): -13% held-out coverage with mean run length 1.2—repetition queries attend to the locally-next token, shredding selection into singletons. Killed by the decode-replay coverage probe.
- **Key-kernel mass-flow transport.** Predicting where evicted attention mass re-flows from a key-similarity kernel (instead of the empirical co-attention T of Section C): correlation with the true decode-time flow ≈ 0. Killed by the offline transport-validation replay of Section C.
- **Layer-only allocation.** Re-allocating budget across layers without per-head granularity collapses the achievable held-out coverage gain from $+8.0\%$ (free per-head) to $+2.7\%$: the variance that matters lives across heads, not layers. The free per-head variant is what Figure 2(a) plots on its *allocate* axis, under the family label *cross-layer allocation*: at its $+8.0\%$ coverage headroom a full sixteen-set run scores $+0.60$ LongBench over SnapKV, still under a point. Killed by the allocation headroom probe, which bounds the axis as a whole: on the same held-out split a better scoring signal is worth 4.4 points of β_{dec} against 0.7 for reallocating the per-head budget.
- **Capped block waterfilling.** Waterfilling the budget over fixed-size blocks with per-block caps: never separated from plain top-B within noise at matched budgets, while adding a tunable. Dropped on the granularity ladder of Table 5.
- **Per-token bias alone.** The closed-form β_i of Section C without value absorption moves attention-output error by $+1.5\%$ at B=128—the wrong way; re-weighting retained logits cannot restore mass whose *value* content is gone. Killed by the same offline replay that validates absorption (11–14%).