REVERSING THE MATTHEW SPLIT IN KV RETENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

KV pairs are evicted once the cache exceeds its budget—a common practice in long-context serving. Existing compression methods default to greedy eviction on a single metric (e.g., accumulated attention). We observe that this greedy rule produces a bipolar "Matthew split" quality distribution: some retain or even surpass the uncompressed baseline, while others degrade significantly, so some answers markedly worsen. Motivated by this observation, we develop a training-free re-ranking post-process, named "RankKV", which reallocates the host keep set under a controllable NDCG floor (a minimum-utility guarantee), using multiple criteria. RankKV balances the two poles of this Matthew split, which we attribute to the local optimum induced by single-surrogate greedy eviction. Our experimental results show that behind mainstream compressors (e.g., H_2O, SnapKV), the post-process yields a positive gain across a wide range of settings. RankKV maintains the host's already-good prompts while substantially lifting the badly damaged ones. Moreover, it hot-plugs behind existing compressors at no training cost, and reverses this Matthew split. Anonymous code is available at https://anonymous.4open.science/r/RankKV-D66A.

1 INTRODUCTION

KV pairs are evicted once the cache exceeds its budget—a common practice in long-context serving. Existing compression methods default to greedy eviction on a single metric (e.g., accumulated attention) (Zhang et al., 2023; Li et al., 2024; Cai et al., 2024; Xiao et al., 2024; Zhong et al., 2024; Ni et al., 2026). The premise is sound: positions that receive a lot of attention are what the next-token predictor has relied on, and dropping attention sinks or heavy hitters produces immediate perplexity spikes (Xiao et al., 2024; Zhang et al., 2023).

We observe that this greedy rule produces a bipolar "Matthew split" quality distribution: some retain or even surpass the uncompressed baseline, while others degrade significantly, so some answers markedly worsen (Figure 1; wrap only). With H_2O at a 50% budget, pooling QMSum, GovReport and arXiv *within* each model, 44% of Llama-3.1-8B prompts with a finite Full-Referenced Index (FRI) lose less than 2% of Full-KV FRI while 38% lose 10% or more; the same almost-none damage ($d{<}2\%$) / severe damage ($d{\geq}10\%$) shares are 48%/31% on Qwen2.5-14B and 53%/29% on Qwen3-32B. The severe-damage bin drops both NativeScore recovery r_A and FactRecall recovery r_B; unsupported-rate recovery r_U stays closer to one—fabrication is not the main loss (Figure 1b).

Motivated by this observation, we develop a training-free re-ranking post-process, named RankKV, to find a trade-off among the two poles of this Matthew split, rather than a new compressor from scratch. We attribute the Matthew split to the local optimum induced by single-surrogate greedy eviction. Replacing attention by another single criterion is the other pole, not the remedy: discarding the host compressor's eviction policy altogether is likely to make quality worse. The hard part is that intervention strength must be controllable—too strong a rewrite hurts already-healthy prompts; too weak a rewrite cannot lift the damaged tail.

Thus, RankKV reallocates the host keep set under a controllable NDCG floor (a minimum-utility guarantee), using multiple criteria (Section 4). To ensure controllable intervention, RankKV allows the post-processing strength to be tuned continuously, where setting the intervention strength to zero automatically recovers the original wrap policy. To escape single-score greed, it pairs that floor with a query-independent information-value score v_i: in-context IDF of the surface token, blended at layer 0 with the ℓ_2 norm of the cached value vector, and that value-vector norm alone on deeper

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

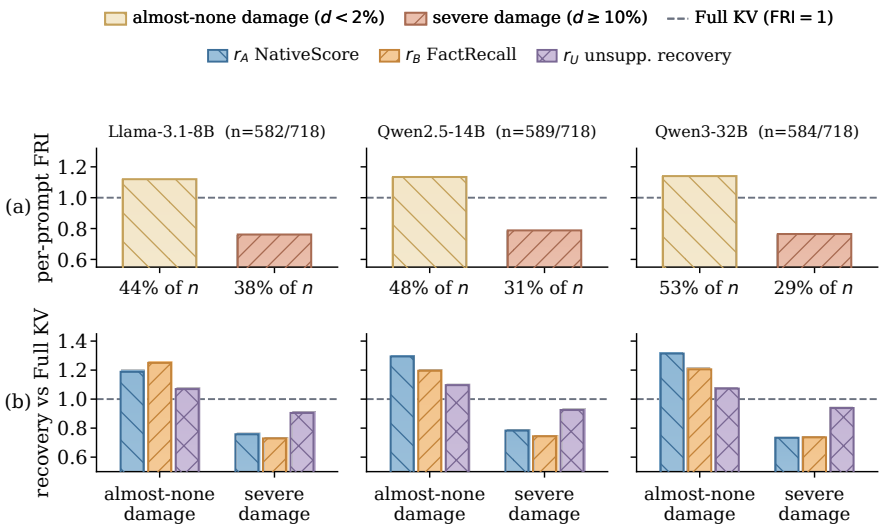

Figure 1: Attention-greedy H_2O (B/T=0.5) splits prompts into almost-none damage (d<2%) and severe damage ($d\geq$10%) versus Full KV—wrap only. (a) Mean per-prompt FRI; x-axis percentages are bin shares of finite prompts; dashed line is Full KV (FRI=1). (b) Same prompts decomposed into NativeScore, FactRecall and unsupported-rate recovery r_U. One column per model—Llama-3.1-8B, Qwen2.5-14B, Qwen3-32B—each pooling QMSum, GovReport and arXiv (finite FRI n=582/589/584 of 718). Per-task shares range 38–58% almost-none damage vs. 22–48% severe damage.

layers (Section 4.3). Our experimental results show that behind mainstream compressors (e.g., H_2O, SnapKV; Figures 2 and 7), the post-process yields a positive gain across a wide range of settings. RankKV maintains the host's already-good prompts while substantially lifting the badly damaged ones, hot-plugs at no training cost, and reverses this Matthew split.

Contributions.

1. **Diagnosis.** The Matthew split is a structural failure of single-surrogate KV eviction: damage versus Full KV is bipolar, and the severe-damage bin drops both NativeScore and FactRecall (Figure 1 and Section 3).
2. **Method.** Unlike prior work, RankKV does not retune the attention surrogate or bucket size, but is a training-free post-process behind the wrap that rewrites the host keep set S^* under an NDCG floor, with intervention strength that can adaptively recover the original wrap policy (Section 4).
3. **Evidence.** Behind mainstream compressors, across a wide range of settings, RankKV yields a positive gain while maintaining already-good prompts and lifting the damaged tail (Section 6).

2 Related Work

Attention-greedy eviction and budget allocation. Most KV cache compressors assume tokens with high attention mass are the most critical for generation. Policies like H_2O (Zhang et al., 2023), Scissorhands (Liu et al., 2023), SnapKV (Li et al., 2024), and StreamingLLM (Xiao et al., 2024) retain heavy hitters and local windows. Subsequent refinements modify the eviction statistic (Oren et al., 2024; Adnan et al., 2024; Ge et al., 2024; Chen et al., 2024; Cai et al., 2025; Tang et al., 2024; Zhang et al., 2024; Wan et al., 2025) or reallocate the cache budget $\{B_\ell\}$ across layers and heads (Cai et al., 2024; Qin et al., 2025; Feng et al., 2024; Xiao et al., 2025; Zhong et al., 2024; Ni et al., 2026). Yet, regardless of budget shape, these methods remain strictly attention-greedy within each bucket, leaving them vulnerable to structural blind spots when attention monopolies form.

Heterogeneous damage and long-context factuality. The failure mode of attention-greedy eviction is highly heterogeneous: damage is task- and prompt-dependent (Chen et al., 2026; Liu et al., 2025a;b). While some prompts survive compression flawlessly, others suffer severe degradation in factual grounding (Liu et al., 2024a; Wu et al., 2024). Prior work traces this to rich-get-richer

cumulative attention dynamics (Gould & van den Hengel, 2026) and structural-role bias (Mandal, 2026). We build on these observations by measuring factuality with a Full-KV-referenced index (FRI) and explicitly binning this heterogeneous damage per prompt.

Hot-plug post-processes and our positioning. To correct eviction errors dynamically, recent approaches post-process a host policy's decisions. Role-aware filters (Mandal, 2026) and K-Norm heuristics attempt to re-weight token importance, while CAOTE (Goel et al., 2025) hot-plugs by re-scoring a candidate list using attention-output error. While we share their goal of correcting the host, RankKV fills a distinct gap: it is a training-free post-process that reallocates the host keep set S^* subject to a controllable set-coverage NDCG floor. This minimum-utility guarantee ensures that setting the intervention strength to zero exactly recovers the underlying wrap, allowing RankKV to safely lift the severely damaged tail without destabilizing already-good prompts.

Orthogonal compression. RankKV operates purely on the logical retention policy, making it orthogonal to and composable with KV quantization (Hooper et al., 2024; Liu et al., 2024b), learned memory (Nawrot et al., 2024), prompt compression (Jiang et al., 2023), and paged memory management (Kwon et al., 2023).

3 Problem: Attention Monopoly

This section introduces the notation used throughout the paper, formally defines the Matthew split of Figure 1 in the Full-referenced recoveries r_A, r_B, r_U and their geometric mean FRI, and formulates the post-process problem as a bounded rewrite of the attention-greedy keep set.

Setup. Consider one transformer layer at decoding step t with T cached positions and a budget $B < T$. Following StreamingLLM and H_2O, the first P sink tokens and the most recent W tokens are never evicted; retention decisions concern the *middle region* of $T-P-W$ positions and its budget $B_{\text{mid}} = B-P-W$ (we write B for B_{mid} when no confusion arises). Each middle-region position i carries two scores: the **attention score** $a_i \geq 0$ is the attention mass key i has accumulated from all queries processed so far (prefill and decode), averaged over heads and min–max normalized to $[0, 1]$ at every eviction event—the same statistic H_2O ranks by. The **information-value score** $v_i \in [0, 1]$ is a query-independent, training-free estimate of the unique content a position carries (Section 4.3).

Fidelity metrics. We score every method M against Full KV on the same prompts. **NativeScore** (NS) is the task's LongBench ROUGE-L scorer with a repetition penalty. **FactRecall** (FR) is the fraction of the reference's factual units—named entities, numbers, years, proper-noun phrases, technical terms—that appear in the output. **UnsupportedRate** (U) is the fraction of the output's factual units that are absent from the source (lower is better). **Rep4** is the 4-gram repetition rate of the output. Let $\bar{s}_A^{(M)}$, $\bar{s}_B^{(M)}$, $\bar{s}_U^{(M)}$ be column means of NS, FR and U. Full-referenced recovery ratios of M versus Full KV are

$$r_A = \frac{\bar{s}_A^{(M)}}{\bar{s}_A^{(\text{Full})}}, \quad r_B = \frac{\bar{s}_B^{(M)}}{\bar{s}_B^{(\text{Full})}}, \quad r_U = \frac{1 - \bar{s}_U^{(M)}}{1 - \bar{s}_U^{(\text{Full})}}. \tag{1}$$

The **Full-Referenced Index** is the geometric mean $\text{FRI}(M) = (r_A\, r_B\, r_U)^{1/3}$. FRI=1 is Full-KV fidelity: fabricating units or losing fluency is not rewarded. ΔFRI is FRI of (wrap+RankKV) minus FRI of the wrap on the same subset. Llama-3.1-8B, Qwen2.5-14B and Qwen3-32B are reported as separate series; a task mean is that model's tasks. Per-prompt FRI uses the same three ratios on one prompt and is used only to bin damage. Let $d = \max(0, 1 - \text{FRI}_{\text{prompt}}(H_2O))$ be greedy damage versus Full KV; Figure 1 bins prompts at $d < 2\%$ (almost-none damage) and $d \geq 10\%$ (severe damage).

The post-process problem. Let $S^* = \arg\max_{|S|=B} \sum_{i \in S} a_i$ be the attention-greedy keep set proposed by the wrap. A post-process may rewrite S^* into another B-set S, but must not spend the wrap's attention coverage without a bound. Informally: inject v by swapping evicted tokens into S^* subject to keeping at least a fraction δ of $\sum_{S^*} a_i$. Section 4.1 turns this into set-NDCG-constrained reranking; δ=1 is a copy of the wrap. The bound may be tight: not every prompt needs a rewrite of S^*.

Definition 3.1 (Attention monopoly index)**.** The monopoly index of a layer is the Gini coefficient of attention inside the greedy keep set,

$$\mathcal{M}(S^*) = \mathrm{Gini}\big(a[S^*]\big) = \frac{\sum_{j,k\in S^*} |a_j - a_k|}{2|S^*| \sum_{j\in S^*} a_j} \in [0, 1). \tag{2}$$

A large $\mathcal{M}$ means retained attention mass sits on a small fraction of kept positions: the budget is nominally full, but most slots are occupied by tokens kept only because everything else scored even lower. That is the regime in which a content swap can change the cache, and it is the *dose* signal: how hard to correct when the keep set is monopolized (Section 4.1).

4 METHOD: RANKKV

RankKV is a within-layer post-processor (Figure 2): the wrap proposes a keep set, RankKV scores and re-ranks the middle region, a bisection finds the largest NDCG-feasible μ^*, and a bounded swap writes the new B-set. An overflow buffer amortizes reranking; production doses the floor by keep-Gini.

4.1 NDCG-BOUNDED RERANKING

Let S^* be the wrap's attention-greedy keep set from Section 3. Cached positions are accessed in parallel, so there is no order inside S and the position discount is flat:

$$\mathrm{NDCG}(S, S^*) = \frac{\sum_{i\in S} a_i}{\sum_{i\in S^*} a_i} \in [0, 1], \tag{3}$$

the fraction of greedy attention mass that S retains—a *constraint*, not a new training objective. We combine the two scores through a single trade-off $\mu \geq 0$,

$$r_i(\mu) = a_i + \mu\, v_i, \qquad S(\mu) = \text{top-}B \text{ by } r_i(\mu), \tag{4}$$

and ask for the most content-aware set that still honours a floor $\delta \in (0, 1]$:

$$\mu^* = \max\big\{\mu \geq 0 :\ \mathrm{NDCG}(S(\mu), S^*) \geq \delta\big\}, \qquad S \leftarrow S(\mu^*). \tag{5}$$

Every token that overtakes into $S(\mu)$ displaces a greedy token with higher attention and lower value. The floor stops the race: Equation (5) spends at most a $1-\delta$ share of greedy attention mass on value. This is a *minimum-utility / floor guarantee* on the wrap's attention coverage.

A constraint, not a new score—and a dose for monopoly. Any fixed re-weighting $a_i + \lambda v_i$ is another single surrogate whose behaviour varies uncontrollably across layers: the same λ that barely touches one cache can rewrite most of another. Equation (5) fixes the *outcome*—the retained attention share—and lets μ^* adapt to each layer's score geometry. Where the keep set is nearly egalitarian we should barely correct; where it is monopolized we can spend more of the floor. After the wrap proposes S^*, we map $\mathcal{M}(S^*)$ through three frozen cut-points $(0.24, 0.28, 0.32)$ to a discrete floor $\delta \in \{1.00, 0.95, 0.90, 0.85\}$: $\mathcal{M} \leq 0.24$ leaves the wrap unchanged (δ=1), and more concentrated keep sets receive more correction. The polarity—higher monopoly $\rightarrow$ lower floor—is load-bearing (Section 6.4). The cut-points were fixed once on Llama development prompts and reused for both Qwen models and all four wraps. Which tokens enter versus leave is the bounded swap of Figure 2(a); a same-layer wrap vs RankKV keep-set is Figure 7.

4.2 SOLVER: EXPLORE UP, REFINE DOWN

$\mathrm{NDCG}(\mu) \coloneqq \mathrm{NDCG}(S(\mu), S^*)$ is non-increasing in μ with $\mathrm{NDCG}(0) = 1$ (Lemma 5.1). The feasible set of Equation (5) is an interval $[0, \mu^*]$. Algorithm 1 uses one shared probe budget: uplink computation runs first—it probes μ_0 and doubles while the floor holds, taking the *right* (larger) feasible endpoint—and leftover uplink capacity is given to downlink bisection, which returns the *left* endpoint. If uplink never leaves the feasible set, residual is unused and the call returns without downlink; if the leftover downlink budget is 0, downlink is skipped as well. Each FEASIBLE probe is one top-B selection; S is updated on feasible probes and returned (a fixed probe budget, not that many quicksorts).

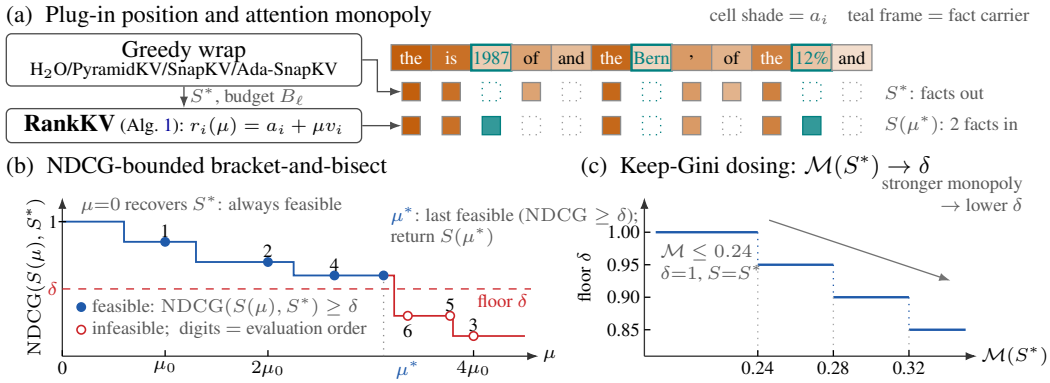

Figure 2: RankKV as a within-layer post-process (schematic). (a) A greedy wrap proposes S^* under the layer budget B_ℓ; RankKV only rewrites which tokens fill it: heavy hitters crowd fact carriers out of S^*, and $S(\mu^*)$ swaps a bounded number back (same-layer wrap vs RankKV keep-set: Figure 7). (b) Set-NDCG is non-increasing from 1 at μ=0; μ^* is the last feasible point (NDCG $\geq \delta$), on or above the floor—empty markers after it are infeasible. Algorithm 1 uses one shared probe budget (uplink leftover → downlink): explore μ upward, spend leftover capacity on downlink, and materialize S once. RankKV is not cheaper than the wrap. (c) Keep-Gini $\mathcal{M}(S^*)$ selects the floor δ: stronger monopoly, lower δ, more correction.

Algorithm 1 RankKV: NDCG-bounded reranking for one layer and one eviction event. One shared probe budget (uplink leftover → downlink).

Input: scores $\{a_i\}, \{v_i\}$ on the middle region, budget B, floor δ, probe μ_0, doubling cap E, halving budget H
$S^* \leftarrow$ top-B by a_i; $\quad A^* \leftarrow \sum_{i \in S^*} a_i$
$\mu_{\text{lo}} \leftarrow 0$; $\quad \mu_{\text{hi}} \leftarrow \mu_0$; $\quad e \leftarrow 0$
define FEASIBLE(μ): $\sum a_i$ on top-B by $a_i + \mu v_i$ is at least δA^*
Upward exploration (take the right feasible endpoint):
while FEASIBLE(μ_{hi}) **and** $e < E$ **do**
  $\mu_{\text{lo}} \leftarrow \mu_{\text{hi}}$; $\quad \mu_{\text{hi}} \leftarrow 2\mu_{\text{hi}}$; $\quad e \leftarrow e + 1$
end while
if FEASIBLE(μ_{hi}) **then return** top-B by $a_i + \mu_{\text{hi}} v_i$
Downward refinement (left endpoint; leftover uplink → downlink; materialize S once):
for $H + (E - e)$ steps **do**
  $\mu \leftarrow (\mu_{\text{lo}} + \mu_{\text{hi}})/2$
  if FEASIBLE(μ) **then** $\mu_{\text{lo}} \leftarrow \mu$ **else** $\mu_{\text{hi}} \leftarrow \mu$
end for
return top-B by $a_i + \mu_{\text{lo}} v_i$

Cost. Each FEASIBLE probe is one top-B selection on the m middle-region keys, implemented by introselect (`numpy.argpartition`): expected $\Theta(m)$, worst-case $\mathcal{O}(m \log m)$—not linear time in the worst case. A full sort is $\mathcal{O}(m \log m)$ on every call; RankKV uses selection, not a full sort. The probe count is one shared budget, not two independent E/H caps: the initial probe of μ_0, an uplink allotment of E doublings, and a base downlink of H bisections, with unused uplink given to downlink. Production uses E=3 and H=6, so at most K=1+E+H=10 probes per eviction event—and fewer if uplink never brackets or residual downlink is unused. The wrap already forms S^* with one top-B (or equivalent) on the same middle region. RankKV adds at most those K selections on $m \leq (1 + \rho)B$ (the overflow buffer), not a scan of the full context T. The extra work is a small constant-factor overhead in the same selection class as the wrap's keep-set construction; RankKV is not cheaper than the wrap.

4.3 INFORMATION-VALUE SCORE

v_i must be cheap, query-independent and available at every layer. We combine two training-free signals. *In-context rarity.* $\text{IDF}(w_i) = \log(N/n_{w_i})$, where N is the prompt length and n_{w_i} the count of token w_i in the prompt; it is computed once at prefill and is high for entities, numbers and technical terms. *Value-vector norm.* $\|V_i^{(\ell)}\|_2$, the ℓ_2 norm of the cached value vector of position i at layer ℓ, averaged over KV heads; it is free because the vector is already in the cache. Both are min–max

normalized over the middle region. Layer 0 blends the surface prior with the norm; every deeper layer uses the value-vector norm alone (lower-case v_i is RankKV's scalar signal, upper-case $V_i^{(\ell)}$ the transformer's value vector).

4.4 Buffer-Overflow Eviction and Two-Level Allocation

Rather than running Algorithm 1 at every decoding step, RankKV lets the cache grow to $B_{\max} = \lfloor (1+\rho)B \rfloor$ and then reranks once, returning to B; prefill is left to the wrap, which compresses once from T to B. The overflow ratio ρ is a scheduler parameter; experiments use ρ=0.1. The budget B_ℓ itself comes from outside: a macro scheduler such as PyramidKV (Cai et al., 2024) or Ada-KV (Feng et al., 2024) decides how much memory each layer or head receives; RankKV decides which tokens fill it.

5 Theoretical Analysis

Monotonicity of NDCG in the trade-off, and therefore exact feasibility of bisection, follows from the swap comparison of Lemma 5.1. RankKV uses set-level coverage (3); any token that enters has weakly higher v and weakly lower a than any token that leaves.

Lemma 5.1 (NDCG monotonicity)**.** *For any non-negative a, v and budget B, with ties in $r_i(\mu)$ broken by index, $\mathrm{NDCG}(\mu)$ is non-increasing in $\mu \geq 0$ and $\mathrm{NDCG}(0) = 1$. Whenever $S(\mu_2) \neq S(\mu_1)$ for $\mu_2 > \mu_1$, every token that enters has value at least as large, and attention at most as large, as every token that leaves.*

Theorem 5.2 (Feasibility and minimum utility)**.** *For every $\delta \leq 1$, Algorithm 1 returns a set S with $|S| = B$ and $\mathrm{NDCG}(S, S^*) \geq \delta$. The floor is exact, not approximate: because NDCG is monotone, every feasible probe lies to the left of μ^*, and the algorithm returns a feasible endpoint. It uses one greedy top-B to form S^* plus at most $1{+}E{+}H$ feasibility probes under the shared budget (uplink first; leftover capacity flows to downlink). δ=1 returns S^* (up to ties in a).*

A short proof of Lemma 5.1 is in Appendix B; Theorem 5.2 is then the standard bisection invariant for a monotone floor.

6 Experiments

This section systematically presents our experimental evaluation. We begin by detailing the experimental design and setup, outlining the evaluation protocol and metrics. Subsequent subsections address each research question (RQ1–RQ4), providing both per-model analyses and aggregate results. For clarity, comprehensive wrap-level summary tables are provided in the appendix, while the main text focuses on the key findings and overall trends.

1. **RQ1.** On the $3{\times}3{\times}4$ cube, does wrap+RankKV boost composite FRI (Figure 4)? How do NativeScore, FactRecall, unsupported rate, Rep4, and length change? (See Tables 3–6.)
2. **RQ2.** Is that mean a redistributive trade: severe-damage prompts recover, while almost-none-damage prompts can sit slightly below the wrap (Figures 5 and 8)?
3. **RQ3.** Are RankKV's mechanisms each effective—the token-aligned score v, the NDCG-bounded dose μ, the decode-Rep4 restart, and the Gini$\to\delta$ polarity—or does the lift collapse when any one of them is removed or reversed (Section 6.4)?
4. **RQ4.** Does RankKV deliver an FRI lift under the sealed keep ratios $\{0.35, 0.50, 0.65\}$ (Figure 6)?

6.1 Setup

Design, in brief.

- **True-dynamic.** Physical KV cropping; streaming per-layer attention hooks; chunked prefill (1024). Not offline simulation.
- **Scenes.** Llama-3.1-8B-Instruct, Qwen2.5-14B-Instruct, Qwen3-32B; QMSum (n=124, 256 new tokens), GovReport (n=194, 1024), arXiv summarization (Cohan et al., 2018) (n=400, 512).

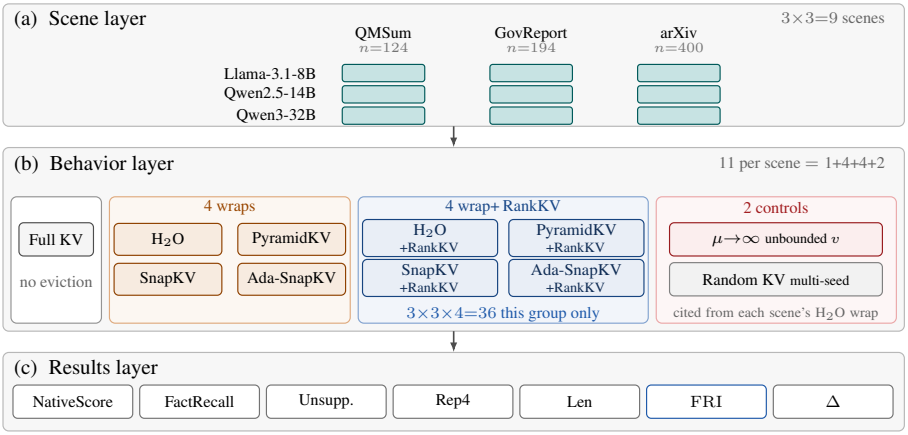

Figure 3: Four research questions on a $3\times3\times4$ wrap cube, with Random and $\mu\to\infty$ as identification controls—not a second cube. (a) Nine scenes: three models $\times$ three long-form tasks. (b) Eleven behaviors per scene: Full KV; four greedy wraps; four wrap+RankKV post-processors ($3\times3\times4=36$ cells); Random KV and $\mu\to\infty$ as shared wrap-agnostic controls (Tables 3 to 6). (c) One row of equal chips: NativeScore, FactRecall, unsupported rate, Rep4, length, FRI and Δ versus the wrap.

Long-form tasks with measurable greedy damage versus Full KV. Contexts truncated to 16k; prompts identical across arms.

- **Wraps.** Official H_2O (Zhang et al., 2023), PyramidKV (Cai et al., 2024) (β=20), SnapKV (Li et al., 2024), and Ada-SnapKV (Feng et al., 2024) (head-adaptive scores collapsed to one keep set per layer). Same sink, window and total budget on every wrap.
- **Controls.** Full KV; Random middle-region eviction (10 seeds; 8 on arXiv); RankKV $\mu\to\infty$ (top-B by v alone). Both Random and $\mu\to\infty$ are wrap-agnostic (same B, sink, window and overflow; $\mu\to\infty$ ignores host attention ranking), so Tables 3 to 6 print the same trailing control rows.
- **Shared knobs.** P=4, W=512, ρ=0.1 on RankKV arms. Main cube at B/T=0.5. Figure 6 plots every sealed E5 Llama B/T on disk ($\{0.35, 0.50, 0.65\}$); probes outside that expansion protocol are omitted.
- **Metrics.** NS, FR, U, Rep4, length, FRI, ΔFRI (Section 3).

6.2 RQ1: Hot-plug FRI and the other metrics

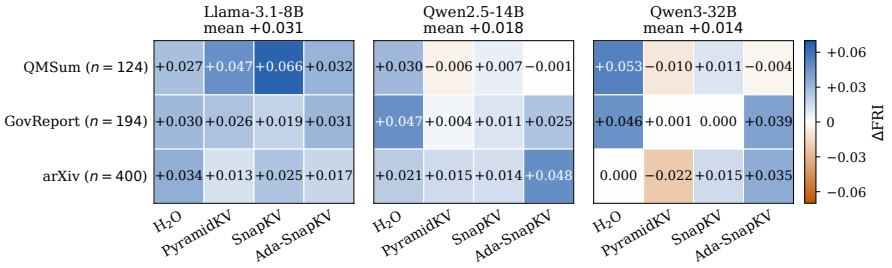

Figure 4: RankKV is net-positive per model on the $3\times3\times4$ cube; five of 24 Qwen cells are negative, all behind PyramidKV or Ada-SnapKV. Cell colour is lexical ΔFRI (wrap+RankKV $-$ wrap): blue $=$ RankKV better, hatched orange $=$ wrap better. Uniform H_2O is the most consistent column; PyramidKV leaves the least headroom on the two Qwen models.

Figure 4 is the 36-cell lexical ΔFRI map. Because of that volume, Tables 3 to 6 report each wrap with metrics as columns and models as row-groups. Each sub-row is the unweighted mean over that model's tasks.

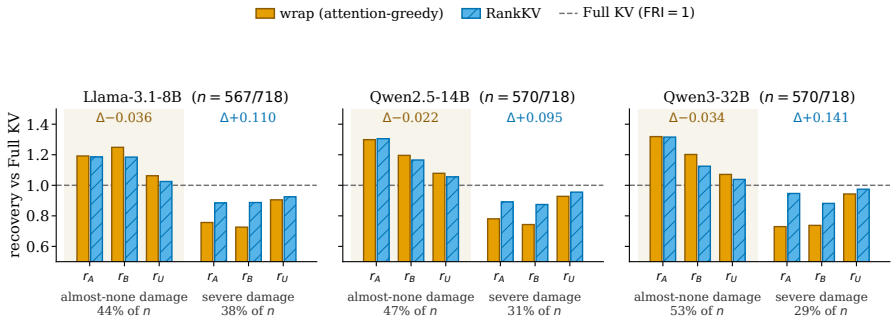

Figure 5: The almost-none-damage dip is the redistributive trade: the host's good regime is largely kept (near-host), and RankKV recovers NativeScore and FactRecall on the severe-damage bin. One panel per model—Llama-3.1-8B, Qwen2.5-14B, Qwen3-32B—of H_2O vs RankKV on the bins of Figure 1 ($d = \max(0, 1 - \mathrm{FRI}_{\mathrm{prompt}}(H_2O))$ vs Full KV). Each panel pools QMSum, GovReport and arXiv for that model (finite pairs n=567/570/570 of 718). Solid orange = wrap (attention-greedy); hatched blue = RankKV; cream band = almost-none damage (d<2%). Metrics: NativeScore r_A, FactRecall r_B, unsupported-rate recovery r_U. SnapKV, PyramidKV and Ada-SnapKV: the same per-model bins (Figure 8).

6.3 RQ2: Redistributive trade

Mean ΔFRI is RankKV minus wrap—a trade, not a uniform lift. In the almost-none-damage bin ($d <$ 2%) per-prompt FRI of RankKV sits slightly *below* the wrap: Llama -0.036 (n=247/567), Qwen2.5-14B -0.022 (269/570), Qwen3-32B -0.034 (302/570). That dip is the trade: the host's already-good regime is largely kept (near-host), and the small negative ΔFRI is the price of recovering the severe-damage tail. In the severe-damage bin ($d \geq 10\%$) the same difference is $+0.110$ (n=217), $+0.095$ (178), $+0.141$ (163). Figure 5 puts that tail recovery on NativeScore and FactRecall; unsupported-rate recovery stays nearer one (bins are descriptive); SnapKV, PyramidKV and Ada-SnapKV are Figure 8.

6.4 RQ3: Are RankKV's mechanisms each effective?

Four sealed ablations on the H_2O scenes. Production uses a length-scaled Rep4 restart (Equation (6)) that regenerates under the bare wrap, not an NDCG fallback; that can only match the wrap, so it does not inflate FRI.

Shuffled v. This ablation is sealed on the H_2O host only—three models × three tasks (9 cells)—as a mechanism check that the production v signal matters under fixed a, G4 polarity and Algorithm 1. Permuting production v on those nine H_2O cells is not a better signal than production v (Table 7). Llama×QMSum shuffled ΔFRI $+0.026$ ties production $+0.027$; Qwen2.5-14B×QMSum shuffled -0.008 while production stays $+0.030$. Six cells sit between wrap and production. The only cell where shuffle exceeds production is Qwen3-32B×arXiv ($+0.003$ vs 0.000), both near zero.

Random and unbounded v. Random and $\mu\to\infty$ are shared wrap-agnostic trailing rows on every wrap table (Tables 3 to 6). Qwen2.5-14B×arXiv Random is the 8-seed mean (FRI 0.941); Qwen3-32B Random pools to FRI 0.992 (Table 2). $\mu\to\infty$ FRI is $3/3$ on every model. On the Llama H_2O slice, Random is 0.03–0.06 FRI below H_2O; $\mu\to\infty$ is below H_2O on QMSum (FRI 0.932 vs 0.947) and GovReport (0.908 vs 0.936). Several Qwen μ cells are competitive; we do not treat $\mu\to\infty$ as uniformly worse than the wrap. Table 2 reports per-task Random FRI seed ranges: Llama×QMSum moves by 0.050, other Llama scenes 0.020–0.022, Qwen2.5-14B 0.022–0.033, Qwen3-32B 0.022–0.062—on the order of wrap+RankKV ΔFRI—so Random is unstable at the scene level; pooling shrinks seed std to $0.005/0.003$ by cancellation.

Decode guard on/off. Table 8 reports wrap / RankKV without the guard / production RankKV on the nine H_2O scenes. Guard-off ΔFRI stays point-positive on all nine cells: Llama $+0.028/+0.012/+0.034$; Qwen2.5 $+0.025/+0.009/+0.021$; Qwen3 $+0.023/+0.026/+0.007$. On Qwen2.5-14B×arXiv, guard-off ties production ($+0.021$; fallback $1/400$). The guard still carries a

large share of the Qwen2.5 Gov lift (production $+0.047$ vs guard-off $+0.009$): a stabilizer, not the sole source.

Polarity. Holding bins, tiers and the decode guard fixed and reversing the Gini$\rightarrow\delta$ map lowers ΔFRI by 0.030 on QMSum (n=119, H_2O FRI 0.954) and 0.022 on GovReport (n=194, H_2O FRI 0.909), and turns the QMSum lift negative (G4 $+0.025 \rightarrow -0.005$; Gov $+0.056 \rightarrow +0.034$; Table 9). Correction must be dosed where the keep set is most concentrated; we freeze the canonical direction.

6.5 RQ4: Resource constraints and compression thresholds

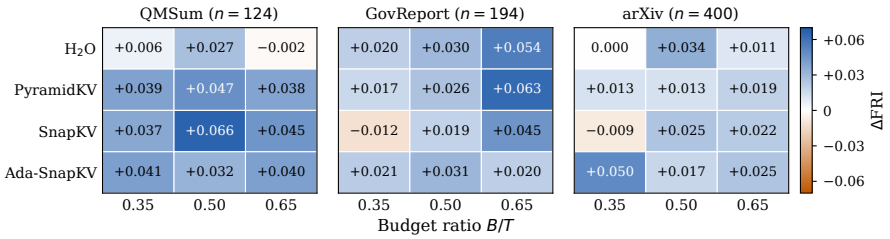

Figure 6: FRI lift under the sealed E5 Llama resource constraints / compression thresholds $\{0.35, 0.50, 0.65\}$ (0.50 cited from the cube). Same sign encoding as Figure 4; no denser B/T grid is on disk.

Figure 6 is every sealed E5 ratio—four wraps $\times$ three thresholds $\times$ three tasks. The lift is not 50%-only: H_2O+RankKV stays point-positive at B/T=0.35 on QMSum ($+0.006$) and GovReport ($+0.020$), and grows further on GovReport at 0.65 ($+0.054$; Pyramid $+0.063$); QMSum Pyramid/Snap/Ada stay in $+0.037$–$+0.045$ at both off-cube ratios.

7 Conclusion

Single-surrogate greedy eviction induces a bipolar "Matthew split" in prompt quality. To address this, we propose RankKV, a training-free, hot-pluggable post-process that reallocates the host keep set. Our experiments demonstrate positive gains across mainstream compressors and robustness to varying resource constraints. Validating its individual mechanisms confirms that RankKV achieves a redistributive trade-off: it maintains the host's already-good prompts while substantially lifting badly damaged ones. Ultimately, RankKV reverses the Matthew split caused by attention-greedy compression at no training cost.

Limitations. There is no direct monopoly identifier: keep-Gini is an auxiliary signal that the polarity ablation supports (Table 9 shows the Gini$\rightarrow\delta$ direction is load-bearing: QMSum $+0.025 \rightarrow -0.005$, Gov $+0.056 \rightarrow +0.034$). Sealed limits also stand: cost ($K \leq 1+E+H$ introselect probes); $d<2\%$ FRI dip as redistributive trade.

Reproducibility Statement

All reported numbers use true-dynamic eviction with streaming attention hooks (prefill chunk 1024) and the protocol of Section 6.1. FRI, wrap settings, Gini cut-points, decode-guard, wrap tables and ablations are in Appendices C to G; code/subsets at `https://anonymous.4open.science/r/RankKV-D66A`.

AI Use Statement

Large language models (LLMs) were only used in this work to assist with writing, including language polishing and grammar checking. They did not invent experiments, numbers, or analyses. Authors verified every reported number against sealed CSVs/JSON and take responsibility for the final content.

REFERENCES

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

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

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

Alex Chen, Renato Geh, Aditya Grover, Guy Van Den Broeck, and Daniel Mingyi Israel. The pitfalls of KV cache compression. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 41530–41553, 2026. doi: 10.18653/v1/2026.acl-long.1926. URL https://aclanthology.org/2026.acl-long.1926/.

Yilong Chen, Guoxia Wang, Junyuan Shang, Shiyao Cui, Zhenyu Zhang, Tingwen Liu, Shuohuan Wang, Yu Sun, Dianhai Yu, and Hua Wu. NACL: A general and effective KV cache eviction framework for LLM at inference time. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7913–7926, 2024.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2097. URL https://aclanthology.org/N18-2097/.

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, 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.

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

Stephen Gould and Anton van den Hengel. Back from the future: Key-value cache management by counter-causal surprise, 2026. URL https://arxiv.org/abs/2607.27600.

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. *arXiv preprint arXiv:2401.18079*, 2024.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*, 2023.

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

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.

Minghui Liu, Aadi Palnitkar, Tahseen Rabbani, Hyunwoo Jae, Kyle Rui Sang, Dixi Yao, Shayan Shabihi, Fuheng Zhao, Tian Li, Ce Zhang, Furong Huang, and Kunpeng Zhang. Hold onto that thought: Assessing KV cache compression on reasoning, 2025a. URL https://arxiv.org/abs/2512.12008.

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

Xiang Liu, Zhenheng Tang, Hong Chen, Peijie Dong, Zeyu Li, Xiuze Zhou, Bo Li, Xuming Hu, and Xiaowen Chu. Semantic integrity matters: Benchmarking and preserving high-density reasoning in KV cache compression, 2025b. URL https://arxiv.org/abs/2502.01941.

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 2bit quantization for KV cache. *arXiv preprint arXiv:2402.02750*, 2024b.

Soumil Mandal. Adaptive filtering of the KV cache: Diagnosing and correcting structural-role bias in LLM inference, 2026. URL https://arxiv.org/abs/2607.13205. Accepted at IEEE MLSP 2026.

Piotr Nawrot, Adrian Łancucki, Marcin Chochowski, David Tarjan, and Edoardo Maria Ponti. Dynamic memory compression: Retrofitting LLMs for accelerated inference. *arXiv preprint arXiv:2403.09636*, 2024.

Xuanfan Ni, Liyan Xu, Chenyang Lyu, Longyue Wang, Mo Yu, Lemao Liu, Fandong Meng, Jie Zhou, and Piji Li. ReFreeKV: Towards threshold-free KV cache compression. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 30827–30841, 2026. doi: 10.18653/v1/2026.findings-acl.1541. URL https://aclanthology.org/2026.findings-acl.1541/.

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

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

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context LLM inference. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 47901–47911, 2024.

Zhongwei Wan, Xinjian Wu, Yu Zhang, Yi Xin, Chaofan Tao, Zhihong Zhu, Xin Wang, Siqi Luo, Jing Xiong, Longyue Wang, and Mi Zhang. D_2O: Dynamic discriminative operations for efficient long-context inference of large language models. In *International Conference on Learning Representations*, 2025.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*, 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.

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

Yuxin Zhang, Yuxuan Du, Gen Luo, Yunshan Zhong, Zhenyu Zhang, Shiwei Liu, and Rongrong Ji. CaM: Cache merging for memory-efficient LLMs inference. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 58840–58850, 2024.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Meizhi Zhong, Xikai Liu, Chen Zhang, Yikun Lei, Yan Gao, Yao Hu, Kehai Chen, and Min Zhang. ZigZagKV: Dynamic KV cache compression for long-context modeling based on layer uncertainty, 2024.

A Notation

Table 1: Notation.

Symbol	Description
T, B, B_ℓ	Context length, KV budget, per-layer budget
P, W	Protected sink and recent-window sizes (4 and 512)
ρ	Overflow ratio; experiments use 0.1
E, H	Uplink doubling cap and base downlink halvings (3 and 6)
K	Feasibility-probe budget per eviction (K=1+E+$H \leq 10$)
a_i	Accumulated, head-averaged, min–max normalized attention
v_i	Information-value score (Section 4.3)
$V_i^{(\ell)}$	Transformer value vector at layer ℓ
μ, $r_i(\mu)$, $S(\mu)$	Trade-off, combined score, top-B set by $r_i(\mu)$
S^*	Attention-greedy keep set (= $S(0)$)
δ	NDCG floor; production $\delta \in \{1.00, 0.95, 0.90, 0.85\}$
$\mathcal{M}(S^*)$	Keep-region Gini (dose of correction)
$B_{\max}$	Overflow watermark $\lfloor (1+\rho)B \rfloor$
FRI	Full-Referenced Index vs Full KV
d	Greedy damage $\max(0, 1 - \mathrm{FRI}_{\mathrm{prompt}}(\mathrm{wrap}))$

B Proof of Lemma 5.1

The argument is the swap comparison specialized to set coverage (3). At $\mu = 0$, $r_i(0) = a_i$ and $S(0) = S^*$, so $\mathrm{NDCG}(0) = 1$. Let $\mu_1 < \mu_2$ and take $j \in S(\mu_2) \setminus S(\mu_1)$, $k \in S(\mu_1) \setminus S(\mu_2)$. Top-$B$ optimality gives $a_j + \mu_2 v_j \geq a_k + \mu_2 v_k$ and $a_k + \mu_1 v_k \geq a_j + \mu_1 v_j$. Adding yields $(\mu_2 - \mu_1)(v_j - v_k) \geq 0$, hence $v_j \geq v_k$; substituting back, $a_k \geq a_j$. Equal-sized difference sets then imply $\sum_{S(\mu_2)} a_i \leq \sum_{S(\mu_1)} a_i$.

C Protocol Details

Prefill is processed in chunks of 1024 tokens (Qwen3 long prompts use 256 where needed for memory); per-layer streaming hooks accumulate key-side attention without materializing full attention matrices. After prefill every arm compresses each layer once from T to B. PyramidKV uses the published per-layer budget formula with β=20 and the same total budget, sink and window as the other arms. SnapKV applies kernel-5 average pooling to accumulated attention before top-k. Ada-SnapKV computes SnapKV-style pooled per-head scores and Ada-KV head budgets, collapsed to one keep set per layer in the streaming stack; RankKV takes that set as S^*. When the decode guard triggers, the sample is regenerated under the bare wrap of its own cell. NDCG@B is recorded at the prefill compression event (layer mean per prompt) and never averaged over decode micro-events.

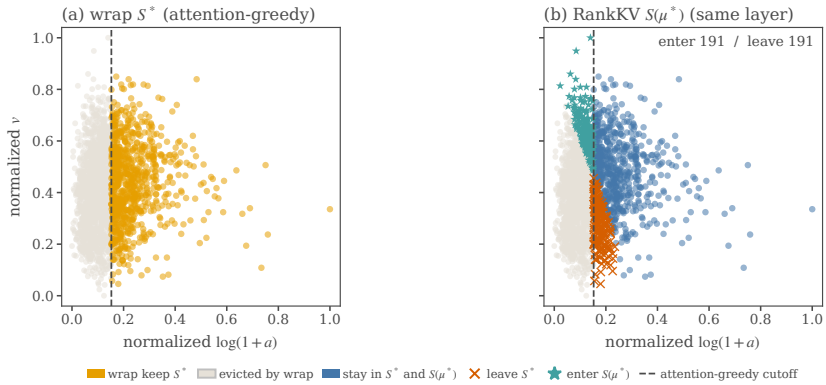

Figure 7: Same-layer wrap vs RankKV keep set (Llama-3.1-8B, one QMSum prompt, layer 7, B/T=0.5, δ=0.95). (a) Attention-greedy wrap S^*: a single a-cutoff (dashed) keeps the right-hand tokens, independent of v. (b) RankKV $S(\mu^*)$ on the same layer: 191 high-v tokens enter from below the cutoff and 191 low-v tokens leave; $\mathrm{NDCG}(S(\mu^*), S^*)$=0.950. δ=1 is the identity $S(\mu^*)$=S^* (μ=0), not a second layer.

Table 2: Per-task Random FRI seed range. Scene-level $\mathrm{FRI}_{\mathrm{global}}$; range = max−min. Pooled-mean seed std is much smaller (0.005/0.003) by cancellation. Both Random and $\mu\to\infty$ are wrap-agnostic.

Model	Task	FRI min	FRI max	range
Llama-3.1-8B	QMSum	0.871	0.921	0.050
	GovReport	0.879	0.899	0.020
	arXiv	0.886	0.909	0.022
Qwen2.5-14B	QMSum	0.957	0.990	0.033
	GovReport	0.910	0.938	0.028
	arXiv	0.934	0.956	0.022
Qwen3-32B	QMSum	0.958	1.001	0.043
	GovReport	0.995	1.017	0.022
	axXiv	0.961	1.023	0.062

D WRAP SUMMARIES

Each table is one host compressor. Metrics are columns; models are row-groups (Llama-3.1-8B, Qwen2.5-14B, Qwen3-32B). Under each model the methods are Full KV, wrap, wrap+RankKV, Random, $\mu\to\infty$ in that order. A number is the unweighted mean over that model's three tasks when all three cells exist; cells are not weighted by subset size n. Random and $\mu\to\infty$ are wrap-agnostic shared controls, so SnapKV, PyramidKV and Ada-SnapKV reuse the same trailing rows as Table 3. ΔFRI is the unweighted mean of that model's three cube values in Figure 4 (wrap+RankKV vs wrap), printed only on the wrap+RankKV row. FRI is not a cross-model index. QMSum n=124, GovReport n=194, arXiv n=400. The same per-prompt almost-none / severe bins on SnapKV, PyramidKV and Ada-SnapKV: Figure 8.

Table 3: Official H_2O wrap. Qwen2.5-14B Random is the unweighted 3-task mean (arXiv 8 seeds 42–49). Unweighted task mean per model. ΔFRI only on wrap+RankKV. NS = NativeScore, FR = FactRecall, U = UnsupportedRate, Len = mean output tokens. Bold: best budgeted arm per metric within a model (Full KV excluded).

Model	Method	NS↑	FR↑	U↓	Rep4↓	Len	FRI	ΔFRI
Llama-3.1-8B	Full KV	0.146	0.301	0.273	0.286	490	1.000	—
	H_2O	0.133	0.281	**0.289**	0.349	490	0.938	—
	H_2O+RK	**0.140**	**0.296**	0.298	**0.314**	487	**0.968**	+0.030
	Random	0.126	0.265	0.310	0.367	496	0.895	—
	$\mu\to\infty$	0.127	0.291	0.296	0.372	492	0.931	—
Qwen2.5-14B	Full KV	0.122	0.305	0.280	0.315	500	1.000	—
	H_2O	0.118	0.293	0.286	0.326	504	0.969	—
	H_2O+RK	**0.124**	**0.305**	**0.283**	0.298	502	**1.002**	+0.033
	Random	0.116	0.281	0.303	0.310	505	0.945	—
	$\mu\to\infty$	0.122	0.293	0.286	**0.276**	500	0.987	—
Qwen3-32B	Full KV	0.126	0.310	0.317	0.337	484	1.000	—
	H_2O	0.123	0.299	0.315	0.345	486	0.980	—
	H_2O+RK	**0.133**	0.308	0.319	**0.291**	485	1.013	+0.033
	Random	0.119	0.355	0.393	0.410	524	0.992	—
	$\mu\to\infty$	**0.133**	**0.320**	**0.313**	0.293	483	**1.027**	—

Table 4: SnapKV wrap. Random and $\mu\to\infty$ are the shared wrap-agnostic controls (same numbers as Table 3), not remeasured per wrap. Unweighted task mean per model. ΔFRI only on wrap+RankKV. NS = NativeScore, FR = FactRecall, U = UnsupportedRate, Len = mean output tokens. Bold: best budgeted arm per metric within a model (Full KV excluded).

Model	Method	NS↑	FR↑	U↓	Rep4↓	Len	FRI	ΔFRI
Llama-3.1-8B	Full KV	0.146	0.301	0.273	0.286	490	1.000	—
	SnapKV	0.133	0.279	0.307	0.351	488	0.929	—
	SnapKV+RK	**0.139**	**0.297**	0.301	**0.319**	488	**0.966**	+0.037
	Random	0.126	0.265	0.310	0.367	496	0.895	—
	$\mu\to\infty$	0.127	0.291	**0.296**	0.372	492	0.931	—
Qwen2.5-14B	Full KV	0.122	0.305	0.280	0.315	500	1.000	—
	SnapKV	0.120	0.299	0.289	0.319	503	0.979	—
	SnapKV+RK	0.120	**0.304**	**0.281**	0.314	503	**0.990**	+0.011
	Random	0.116	0.281	0.303	0.310	505	0.945	—
	$\mu\to\infty$	**0.122**	0.293	0.286	**0.276**	500	0.987	—
Qwen3-32B	Full KV	0.126	0.310	0.317	0.337	484	1.000	—
	SnapKV	0.128	0.303	0.323	0.331	487	0.990	—
	SnapKV+RK	0.128	0.310	0.326	0.315	485	0.998	+0.009
	Random	0.119	0.355	0.393	0.410	524	0.992	—
	$\mu\to\infty$	**0.133**	**0.320**	**0.313**	**0.293**	483	**1.027**	—

Table 5: PyramidKV wrap. Random and $\mu \to \infty$ are the shared wrap-agnostic controls (same numbers as Table 3), not remeasured per wrap. Unweighted task mean per model. ΔFRI only on wrap+RankKV. NS = NativeScore, FR = FactRecall, U = UnsupportedRate, Len = mean output tokens. Bold: best budgeted arm per metric within a model (Full KV excluded).

Model	Method	NS↑	FR↑	U↓	Rep4↓	Len	FRI	ΔFRI
Llama-3.1-8B	Full KV	0.146	0.301	0.273	0.286	490	1.000	—
	PyramidKV	0.129	0.278	0.302	0.367	492	0.918	—
	PyramidKV+RK	**0.134**	0.290	**0.296**	**0.337**	489	**0.947**	+0.029
	Random	0.126	0.265	0.310	0.367	496	0.895	—
	$\mu \to \infty$	0.127	**0.291**	**0.296**	0.372	492	0.931	—
Qwen2.5-14B	Full KV	0.122	0.305	0.280	0.315	500	1.000	—
	PyramidKV	0.115	0.288	**0.285**	0.344	503	0.954	—
	PyramidKV+RK	0.114	0.291	0.286	0.340	501	0.958	+0.004
	Random	0.116	0.281	0.303	0.310	505	0.945	—
	$\mu \to \infty$	**0.122**	**0.293**	0.286	**0.276**	500	**0.987**	—
Qwen3-32B	Full KV	0.126	0.310	0.317	0.337	484	1.000	—
	PyramidKV	0.120	0.287	0.324	0.331	490	0.952	—
	PyramidKV+RK	0.121	0.286	0.344	0.332	487	0.943	−0.010
	Random	0.119	0.355	0.393	0.410	524	0.992	—
	$\mu \to \infty$	**0.133**	**0.320**	**0.313**	**0.293**	483	**1.027**	—

Table 6: Ada-SnapKV wrap. Random and $\mu \to \infty$ are the shared wrap-agnostic controls (same numbers as Table 3), not remeasured per wrap. Unweighted task mean per model. ΔFRI only on wrap+RankKV. NS = NativeScore, FR = FactRecall, U = UnsupportedRate, Len = mean output tokens. Bold: best budgeted arm per metric within a model (Full KV excluded).

Model	Method	NS↑	FR↑	U↓	Rep4↓	Len	FRI	ΔFRI
Llama-3.1-8B	Full KV	0.146	0.301	0.273	0.286	490	1.000	—
	Ada-SnapKV	0.129	0.269	0.291	0.358	482	0.912	—
	Ada-SnapKV+RK	**0.132**	0.281	**0.288**	**0.349**	489	**0.938**	+0.027
	Random	0.126	0.265	0.310	0.367	496	0.895	—
	$\mu \to \infty$	0.127	**0.291**	0.296	0.372	492	0.931	—
Qwen2.5-14B	Full KV	0.122	0.305	0.280	0.315	500	1.000	—
	Ada-SnapKV	0.115	0.290	0.303	0.348	504	0.947	—
	Ada-SnapKV+RK	0.116	**0.297**	0.288	0.318	500	0.971	+0.024
	Random	0.116	0.281	0.303	0.310	505	0.945	—
	$\mu \to \infty$	**0.122**	0.293	**0.286**	**0.276**	500	**0.987**	—
Qwen3-32B	Full KV	0.126	0.310	0.317	0.337	484	1.000	—
	Ada-SnapKV	0.124	0.295	0.320	0.336	486	0.972	—
	Ada-SnapKV+RK	0.125	0.308	0.314	0.324	483	0.995	+0.023
	Random	0.119	0.355	0.393	0.410	524	0.992	—
	$\mu \to \infty$	**0.133**	**0.320**	**0.313**	**0.293**	483	**1.027**	—

E Mechanism Ablations

Shuffle, guard-off and polarity reverse are sealed on the H_2O host only (mechanism checks on that wrap's scenes)—not SnapKV / PyramidKV / Ada-SnapKV, and not a second 36-cell cube. Those other wraps were not re-run under these ablations, so we do not pool them or claim transfer.

Table 7: Permute production v on nine H_2O cells only (3 models $\times$ 3 tasks; same a, G4 polarity, Algorithm 1). SnapKV / PyramidKV / Ada-SnapKV were not re-run, so they are not pooled. Δ is versus H_2O. Shuffled v is not a better signal; the only shuffle>production cell is Qwen3-32B$\times$arXiv ($+0.003$), both near zero.

Model	Task	Δ prod.	Δ shuffle	shuffle $-$ prod.
Llama-3.1-8B	QMSum	$+0.027$	$+0.026$	-0.002
Llama-3.1-8B	GovReport	$+0.030$	$+0.005$	-0.025
Llama-3.1-8B	arXiv	$+0.034$	$+0.012$	-0.022
Qwen2.5-14B	QMSum	$+0.030$	-0.008	-0.039
Qwen2.5-14B	GovReport	$+0.047$	$+0.031$	-0.016
Qwen2.5-14B	arXiv	$+0.021$	$+0.005$	-0.016
Qwen3-32B	QMSum	$+0.053$	$+0.039$	-0.014
Qwen3-32B	GovReport	$+0.046$	$+0.035$	-0.011
Qwen3-32B	arXiv	0.000	$+0.003$	$+0.003$

Table 8: Decode-Rep4 restart on/off versus the official H_2O wrap, all nine scenes. Wrap FRI and production Δ are from the cube. Δ is versus wrap. Guard-off fallback is 0 on every sealed cell.

Model	Task	wrap FRI	Δ off	Δ +guard	fb +g
Llama-3.1-8B	QMSum	0.947	$+\mathbf{0.028}$	$+0.027$	27/124
	GovReport	0.936	$+0.012$	$+\mathbf{0.030}$	26/194
	arXiv	0.931	$+\mathbf{0.034}$	$+\mathbf{0.034}$	1/400
Qwen2.5-14B	QMSum	1.011	$+0.025$	$+\mathbf{0.030}$	7/124
	GovReport	0.942	$+0.009$	$+\mathbf{0.047}$	91/194
	arXiv	0.955	$+\mathbf{0.021}$	$+\mathbf{0.021}$	1/400
Qwen3-32B	QMSum	0.987	$+0.023$	$+\mathbf{0.053}$	55/124
	GovReport	0.981	$+0.026$	$+\mathbf{0.046}$	49/194
	arXiv	0.971	$+\mathbf{0.007}$	0.000	4/400

Table 9: Canonical vs reversed Gini$\rightarrow$ δ on Llama-3.1-8B$\times$$H_2O$ (QMSum, GovReport). Δ is versus this ablation's H_2O arm (QMSum FRI 0.954, n=119; GovReport FRI 0.909, n=194).

Task	n	H_2O FRI	G4 Δ	rev Δ	rev$-$G4
QMSum	119	0.954	$+0.025$	-0.005	-0.030
GovReport	194	0.909	$+0.056$	$+0.034$	-0.022

F Decode Guard

The guard monitors Rep4 at $\lfloor L_{\max}/2 \rfloor$ generated tokens and at end-of-sequence. A length-only schedule shared by all datasets and models is

$$\tau(L) = \mathrm{clip}\left(\tau_0 + \kappa \log_2 \frac{L}{L_{\text{ref}}},\ \tau_0,\ \tau_{\max}\right), \tag{6}$$

with τ_0=0.3, L_{ref}=128, $\tau_{\max}$=0.65 and κ=0.15. A triggered sample is regenerated from scratch under the bare wrap.

G Redistributive Bins on SnapKV, PyramidKV and Ada-SnapKV

Figure 5 is official H_2O. Figure 8 repeats that recipe on SnapKV, PyramidKV and Ada-SnapKV (top to bottom). For each model we pool QMSum, GovReport and arXiv; we never average FRI across Llama and Qwen. Damage $d = \max(0, 1 - \mathrm{FRI}_{\text{prompt}}(\text{wrap}))$ versus Full KV; bins d<2% (almost-none) and $d\geq$10% (severe), the same edges as Figures 1 and 5. Bars are wrap vs RankKV on NativeScore r_A, FactRecall r_B and unsupported-rate recovery r_U; Δ is mean per-prompt FRI change (RankKV $-$ wrap) in that bin. Finite pairs require the Full-KV denom gate and strictly positive r_A, r_B, r_U on both arms. Full KV is the same-model expansion Full (cited from that model's H_2O run when the wrap CSV has none).

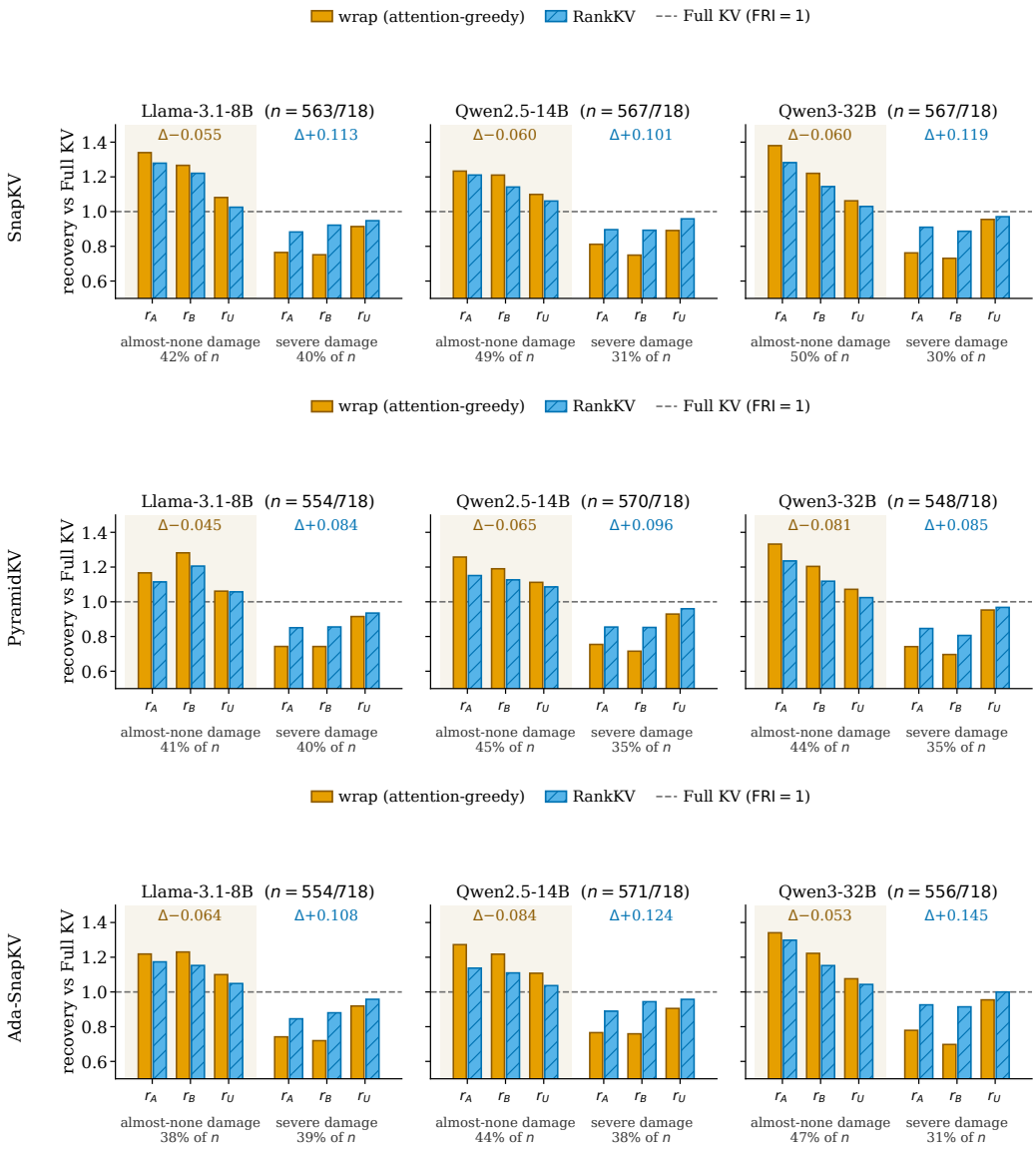

Figure 8: Same recipe as Figure 5 on SnapKV, PyramidKV and Ada-SnapKV (top to bottom). One panel per model; each panel pools that model's three tasks. Cream band = almost-none damage (d<2%); solid orange = wrap; hatched blue = RankKV. y-axis: recovery vs Full KV.