MONTE CARLO ESTIMATION FOR KV CACHE EVICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Most KV-cache eviction methods ask, in effect, *which memory appeared important while reading the prompt?* We instead ask, *which memory will matter while answering?* Since decoding queries are unavailable at eviction time, prior future-aware methods rely on pseudo-responses or synthetic future-query estimates. We cast fixed-budget future-aware eviction as distributional estimation over plausible model-conditional query trajectories and introduce **LORE-KV** (***L**ookahead **O**utput-perturbation with **R**eliability-weighted **E**nsembles for **K**ey-**V**alue caches*), a training-free method that samples short autoregressive continuations from the frozen target model and uses their response-side query states to estimate prompt-token utility. Tokens are scored by projected leave-one-out attention-output deletion cost and aggregated across sampled futures with optional trajectory weighting. The temporary continuations are discarded before final decoding, requiring no auxiliary model or training. Ablations isolate the mechanism: at $B{=}128$, a single response-side continuation recovers about 89% of the gain over the prompt-window control, while additional futures provide smaller improvements. At $B{=}128$, LORE-KV raises the LongBench average on Qwen2.5-14B from 45.49 to 48.24 ($+2.75$) and the 16K RULER average on Mistral-7B from 45.20 to 51.05 ($+5.85$). Gains diminish at larger cache budgets and coexist with task-level regressions. LORE-KV incurs 1.46-$2.77\times$ AnDPro's per-sample wall-clock time as a one-time compression overhead across six dense and hybrid-attention backbones.

1 INTRODUCTION

Autoregressive large language models (LLMs) rely on key-value (KV) caching to reuse previously computed hidden states during decoding. Although KV caching accelerates generation, its memory and bandwidth costs grow with sequence length and batch size, making long-context inference increasingly expensive (Behnam et al., 2025; Li et al., 2025a; Łańcucki et al., 2025). KV-cache eviction reduces this cost by retaining only a subset of KV pairs, with existing methods estimating importance using recency or attention sinks (Xiao et al., 2024), accumulated or prompt-end attention (Zhang et al., 2023; Liu et al., 2023; Li et al., 2024), or key-space geometry (Devoto et al., 2024; Park et al., 2025).

A key challenge is that KV eviction happens before future decoding queries are known, so prompt-based scores may keep tokens that matter during prefill but not during generation. Future-aware methods reduce this mismatch using pseudo or synthetic queries (Wang et al., 2025b), but the benefit and cost of using multiple plausible futures remain unclear. Learned predictors can estimate future utility, but require model-specific training (Ahn et al., 2026; Moschella et al., 2026; Bui et al., 2026; Yang et al., 2026). Moreover, attention alone is incomplete because deletion impact also depends on the value vectors (Guo et al., 2024; Geng et al., 2025; Mai & Kim, 2026). These gaps motivate a training-free method that combines future-query estimation with output-aware token scoring.

To address these limitations, we propose **LORE-KV** (***L**ookahead **O**utput-perturbation with **R**eliability-weighted **E**nsembles for **K**ey-**V**alue caches*), a training-free method that estimates token importance using the queries the model is likely to encounter during generation. Prior future-aware methods estimate decoding demand from either a pseudo-response trajectory, as in LAQ, or synthetic queries from a fitted hidden-state distribution. LORE-KV instead samples multiple model-

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.

conditional autoregressive trajectories from the frozen target model and estimates fixed-budget token utility via reliability-weighted expected projected deletion cost over their query states. Our focus is the query reference: what multiple sampled futures add beyond one, and the explicit trade-off of additional compression-time compute for retained-cache quality. LORE-KV is training-free to deploy but incurs lookahead overhead on every request.

Contributions. Our contributions are threefold. **First**, we quantify and mechanistically characterize the *query-reference mismatch* between prompt-side and future decoding queries, and show that the benefit of future-aware eviction increases with this mismatch. **Second**, we use multiple short future trajectories to estimate which cached tokens are most important for upcoming generation, and softly weight more reliable trajectories. **Third**, we validate this mechanism through query controls and matched lookahead-token budget controls, future-query oracle analyses, and evaluations across six dense and hybrid-attention backbones, cache budgets, context lengths, and benchmarks.

2 Related Work

2.1 Prompt-side KV eviction

Most KV-cache eviction methods estimate token importance from information available during prefill. StreamingLLM preserves attention sinks and recent tokens (Xiao et al., 2024), H2O accumulates historical attention (Zhang et al., 2023), and SnapKV scores tokens using a prompt-end observation window (Li et al., 2024). Other methods improve how a fixed budget is distributed across layers or heads, including PyramidKV and Ada-KV (Cai et al., 2025b; Feng et al., 2025b), while DefensiveKV improves robustness to noisy query estimates (Feng et al., 2025a). These methods refine retention or budget allocation but remain prompt-side; LORE-KV instead estimates token utility from plausible future decoding queries.

2.2 Future-query estimation and output-aware scoring

A more recent line of work approximates the queries encountered during decoding. Expected Attention analytically estimates attention under a future-query distribution (Devoto et al., 2026), while DapQ constructs position-aware pseudo-queries (Tian et al., 2026). LAQ obtains lookahead queries from a short pseudo-response, LAQ++ combines them with prompt-local queries (Wang et al., 2025b), and GVote samples synthetic queries from a fitted hidden-state distribution to determine an adaptive cache budget (Tang et al., 2025). Learned approaches such as LookaheadKV and IndexMem predict future utility or learn compact memory mechanisms (Ahn et al., 2026; Yang et al., 2026). Complementary methods improve the eviction objective: OBCache measures attention-output perturbation (Gu et al., 2025), AnDPro uses projected attention-output information (Geng et al., 2025), and LaProx incorporates projected values and inter-head interactions (Mai & Kim, 2026). LORE-KV samples multiple model-conditional autoregressive trajectories and, under a fixed cache budget, estimates token utility via reliability-weighted expected projected deletion cost over their query states.

2.3 Orthogonal KV-cache compression

Other work reduces KV-cache cost along dimensions orthogonal to prompt-time token selection. Quantization methods such as KIVI and KVQuant reduce cache precision (Liu et al., 2024b; Hooper et al., 2024); representation-oriented methods exploit low-rank, cross-layer, or mergeable structure (Kang et al., 2024; Liu et al., 2024a); and serving or decode-time methods control cache growth during long generation (Liu et al., 2026b; Łańcucki et al., 2025; Su et al., 2026). These techniques are complementary to LORE-KV because they modify cache representation, precision, serving, or decode-time maintenance rather than the prompt-time criterion used to determine which KV entries should survive eviction. Aggressive eviction can additionally remove instructions or answer-bearing evidence (Yang et al., 2024b; Chen et al., 2026a; Garcia, 2026). Additional related work is discussed in Appendix A.

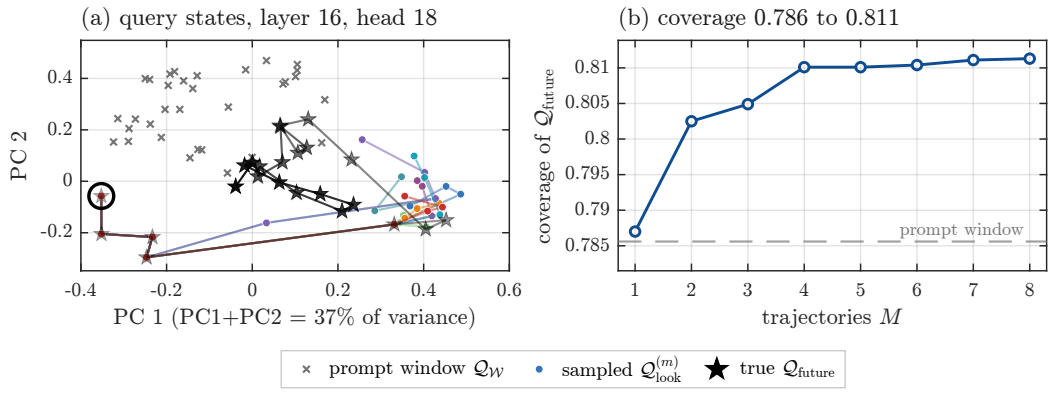


Figure 1: Monte Carlo future-query estimation: lookahead queries better match future decoding queries, with coverage saturating near $M = 4$.

3 Methodology

LORE-KV consists of three stages: (i) Monte Carlo future-query estimation from M short continuations (Section 3.2), (ii) reliability weighting using likelihood, entropy, or disagreement (Section 3.3), and (iii) projected counterfactual eviction scoring (Sections 3.4-3.5). Algorithm 1 summarizes the method, Table 4 lists the hyperparameters, and Appendices C and D provide implementation details and ablations.

3.1 Problem Setup

Let f_θ be a frozen pretrained transformer. Given a prompt $X = (x_1, \ldots, x_n)$, a prefill pass produces cached keys and values $\mathbf{K}^{l,h}$ and $\mathbf{V}^{l,h}$ for each layer l and KV head h, along with prompt-window queries $\mathcal{Q}^{l,h}_{\mathcal{W}}$. We define the prompt-end window as $\mathcal{W} = \{n - w + 1, \ldots, n\}$, where w is the number of recent prompt tokens included in $\mathcal{W}$, set to 32 by default. We optionally protect structural tokens $\mathcal{B}$. Together with the first-token attention sink (Xiao et al., 2024), the protected set is $\mathcal{P} = \{1\} \cup \mathcal{W} \cup \mathcal{B}$, leaving eviction candidates $\mathcal{C} = \{1, \ldots, n\} \setminus \mathcal{P}$. Further setup details are provided in Appendix C.

3.2 Test-Time Lookahead Query Construction

Because prompt-side queries can poorly represent those used during generation, LORE-KV uses a small test-time budget to sample M short continuations from the frozen model, providing multiple plausible response-side query trajectories for estimating future token utility:

$$Y^{(m)} \sim f_\theta(\cdot \mid X), \qquad Y^{(m)} = \{y_1^{(m)}, \ldots, y_T^{(m)}\}, \qquad m = 1, \ldots, M, \tag{1}$$

where T is small, and each continuation serves as a temporary probe of future decoding behavior. LORE-KV samples multiple autoregressive future trajectories, so the resulting query states reflect plausible response-conditioned futures before fixed-budget eviction. For trajectory m, we store the corresponding lookahead queries $\mathcal{Q}^{(m),l,h}_{\text{look}} = \{\mathbf{q}_1^{(m),l,h}, \ldots, \mathbf{q}_T^{(m),l,h}\}$ and form the scoring set $\mathcal{Q}^{l,h}_{\text{score}} = \mathcal{Q}^{l,h}_{\mathcal{W}} \cup \bigcup_{m=1}^{M} \mathcal{Q}^{(m),l,h}_{\text{look}}$. Each trajectory is evaluated under the same distribution from which it is sampled so that the reliability score remains correctly centered. Together, these trajectories provide a Monte Carlo approximation of future query states, improving coverage over prompt-only queries (Figure 1; Appendix F.2). The single-surrogate case is detailed in Appendix C.2, while Tables 13 and 13b show gains with increasing lookahead compute that eventually saturate.

3.3 Particle Reliability Weighting: Entropy and Trajectory Disagreement

Because lookahead trajectories vary in reliability, LORE-KV assigns each a weight to reduce the influence of uncertain or atypical branches. Let $u \in \{1, \ldots, T\}$ index lookahead steps, $h_u^{(m)} =$

$(X, y_{<u}^{(m)})$ denote the context of trajectory m at step u, and $q(\cdot \mid h)$ the sampling distribution. Since raw log-probability conflates token atypicality with contextual uncertainty, we separate these effects using the centered score

$$c_u^{(m)} = \log q\Big(y_u^{(m)} \mid h_u^{(m)}\Big) + H\Big(q(\cdot \mid h_u^{(m)})\Big), \tag{2}$$

for which $\mathbb{E}[c_u^{(m)} \mid h_u^{(m)}] = 0$. With $\bar{c}_m = \frac{1}{T}\sum_u c_u^{(m)}$ and $\bar{H}_m = \frac{1}{T}\sum_u H(q(\cdot \mid h_u^{(m)}))$, trajectory reliability is

$$R_m = \beta\, \bar{c}_m - \gamma\, \bar{H}_m - \lambda_D\, \hat{D}_m, \tag{3}$$

where $\bar{c}_m$ measures trajectory atypicality relative to local uncertainty, $\bar{H}_m$ captures uncertainty, and $\hat{D}_m$ measures disagreement with other trajectories. These signals are combined into a reliability score R_m, which is converted into soft trajectory weights,

$$\omega_m = \frac{\exp(R_m)}{\sum_{r=1}^{M} \exp(R_r)}, \tag{4}$$

so that more reliable trajectories receive larger weights. We use soft weighting because reliability is only approximate. *LORE-KV/E* uses entropy-based uncertainty, while *LORE-KV/D* additionally uses query-space disagreement; both share the same trajectories and differ only in their reliability coefficients. Details and settings are provided in Appendix C.3, Table 4, and Table 14.

3.4 Closed-Form Counterfactual Deletion Effect

With the scoring queries in hand, we now specify *what* is measured for each candidate token. For a scoring query $\mathbf{q}^{l,h}$ and candidate token $i \in \mathcal{C}$, the full-cache attention weight is $a_{q,i}^{l,h} = \mathrm{Softmax}_i\Big((\mathbf{q}^{l,h})^\top \mathbf{k}_i^{l,h}/\sqrt{d}\Big)$, where d is the head dimension, and the corresponding full-cache attention output is $\mathbf{o}_q^{l,h} = \sum_{j=1}^{n} a_{q,j}^{l,h}\mathbf{v}_j^{l,h}$. We write $(a_{q,i}^{l,h}, \mathbf{o}_q^{l,h}) = \mathrm{Attn}(\mathbf{q}^{l,h}, \mathbf{K}^{l,h}, \mathbf{V}^{l,h})$ for this full-cache attention weight–output pair. If token i is removed and the remaining attention weights are renormalized, the post-deletion output becomes $\mathbf{o}_{q,-i}^{l,h} = \sum_{j\neq i} \frac{a_{q,j}^{l,h}}{1-a_{q,i}^{l,h}+\epsilon}\mathbf{v}_j^{l,h} = \frac{\mathbf{o}_q^{l,h} - a_{q,i}^{l,h}\mathbf{v}_i^{l,h}}{1-a_{q,i}^{l,h}+\epsilon}$, where ϵ prevents numerical instability. The corresponding leave-one-out deletion effect is therefore

$$\Delta\mathbf{o}_{q,i}^{l,h} = \mathbf{o}_q^{l,h} - \mathbf{o}_{q,-i}^{l,h} \approx \frac{a_{q,i}^{l,h}}{1 - a_{q,i}^{l,h} + \epsilon}\Big(\mathbf{v}_i^{l,h} - \mathbf{o}_q^{l,h}\Big). \tag{5}$$

Eq. 5 shows why attention alone is insufficient: a token is important only when it receives non-negligible attention and its value meaningfully differs from the current attention output. Let $\mathbf{W}_{l,h}^O$ denote the output-projection block for head h in layer l. LORE-KV scores the residual-stream impact of deleting token i as

$$e_{q,i}^{l,h} = \Big\|\mathbf{W}_{l,h}^O \Delta\mathbf{o}_{q,i}^{l,h}\Big\|_2^2. \tag{6}$$

Unlike raw attention, this score measures the deletion effect after projection into the model stream. Further details are given in Appendix C.2. Figure 5 evaluates the resulting cache fidelity, while Figure 12 shows how the perturbation is distributed across layers and heads.

3.5 Core Score, Selection, and Compressed Cache

For each token i, we group the per-query perturbation scores into $\mathcal{E}_{i,\mathcal{W}}^{l,h} = \{e_{q,i}^{l,h} : q \in \mathcal{Q}_{\mathcal{W}}^{l,h}\}$ for the prompt window and $\mathcal{E}_{i,m}^{l,h} = \{e_{q,i}^{l,h} : q \in \mathcal{Q}_{\text{look}}^{(m),l,h}\}$ for trajectory m, then aggregate them using $\psi(\cdot)$, which reduces to the mean in the core configuration, to obtain the final token score,

$$S_i^{l,h} = \psi(\mathcal{E}_{i,\mathcal{W}}^{l,h}) + \lambda_L \sum_{m=1}^{M} \omega_m\, \psi(\mathcal{E}_{i,m}^{l,h}). \tag{7}$$

The lookahead term changes which tokens are retained: at $B = 128$, about 20% differ from the prompt-only rule (Figure 9). With uniform weights, Eq. 7 is a Monte Carlo estimate of expected

projected deletion cost, while entropy/disagreement weighting provides a soft finite-sample robustness correction. Table 14 isolates this future-aware contribution. The resulting scores are then converted into the compressed cache through two core operators:

$$\{B_{l,h}\}_h = \mathrm{Alloc}(B_l, \{E_{l,h}\}_h; \tau)\,, \qquad \mathcal{R}^{l,h} = \mathrm{Select}\big(\mathbf{S}^{l,h}, B_{l,h}\big)\,. \tag{8}$$

For the core method, Alloc uniformly distributes each layer's budget across heads, and Select keeps the top-scoring tokens per head. The final retained set is $\mathcal{R}^{l,h}_{\mathrm{final}} = \mathcal{P} \cup \mathcal{R}^{l,h}$, with $(\mathbf{K}^{l,h}_\star, \mathbf{V}^{l,h}_\star) = \mathrm{Gather}(\mathbf{K}^{l,h}, \mathbf{V}^{l,h}, \mathcal{R}^{l,h}_{\mathrm{final}})$. The lookahead continuations are then discarded and decoding uses only the compressed cache. Adaptive allocation and diversity-aware selection are ablated separately in Appendix D.

4 Experiments

We evaluate LongBench (Bai et al., 2024), RULER (Hsieh et al., 2024), and Needle-in-a-Haystack (NIAH) for long-context understanding, controlled context-length stress testing, and long-range retrieval, respectively. Mistral-7B-Instruct-v0.2 (Jiang et al., 2023) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024) support comparison with prior KV-cache methods, while Qwen2.5-14B-Instruct (Yang et al., 2024a) and Ministral-3-8B-Instruct-2512 (Liu et al., 2026a) test generalization across newer model families and scales. Qwen3.5-9B and Gemma-4-12B-it further test *hybrid-attention* backbones, in which only a minority of layers keep a growing KV cache as discussed in Appendix E.2. Shared settings, hyperparameters, and dataset details are provided in Tables 4 and 5; main results appear in Sections 4.1, 4.2, and 4.3, efficiency and mechanism analyses in Sections 4.4 and 4.5, ablations in Table 14 and Appendix D, and implementation details in Appendix E. We report LORE-KV /E and LORE-KV /D under the same pipeline, with further details in Appendices G and F.4, and evaluate all methods following the protocol of Geng et al. (2025) with matched cache budgets, backbones, and observation windows. We highlight the **best** and second-best results at each cache budget.

4.1 Evaluation on LongBench

Overall performance. Tables 1 and 6 show that LORE-KV consistently outperforms the matched baselines across both backbones and cache budgets, with the largest gains under aggressive compression and diminishing margins as the cache budget increases. At $B = 128$ on Mistral, LORE-KV /E achieves 40.28 versus 38.96 for AnDPro.

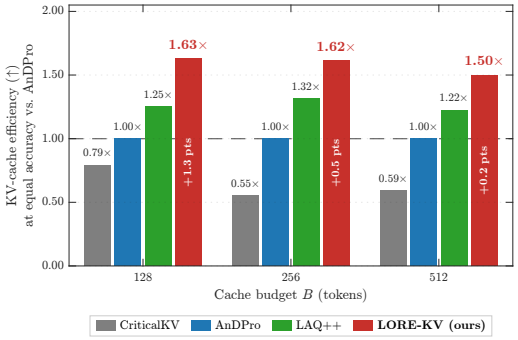


Figure 2: Equal-accuracy KV-cache efficiency on LongBench (Mistral-7B).

Across three independent seeds, LORE-KV /E obtains a mean LongBench score of 40.28 (95% CI $[39.97, 40.59]$). The lower bound remains above AnDPro (38.96), indicating that the improvement is robust to lookahead sampling variation[1]. We report published LAQ++(8) results only as a contextual comparison. For broader context, Figure 4 shows a consistently favorable accuracy-memory trade-off against OBCache (Gu et al., 2025), CAOTE (Goel et al., 2025), and Expected Attention (Devoto et al., 2026): LORE-KV preserves near-full-cache quality at substantially lower cache retention. These comparisons are contextual rather than fully controlled because the evaluation protocols differ. Importantly, the improvement is not limited to pointwise score differences: Figure 2 shows that LORE-KV reaches the same LongBench accuracy with 1.50–1.63× greater cache efficiency than AnDPro across the aggressive-compression operating points.

Where the gain comes from. Table 14 shows that LORE-KV's main gain comes from correcting the query-reference mismatch. At $B = 128$, replacing prompt-side queries with a single response-

[1]LAQ++(8) is omitted at $B = 1024$ because the original paper does not report results for this setting.

Table 1: LongBench results on Mistral-7B-Instruct-v0.2. Dataset abbreviations follow Table 5.

	Single-Doc QA			Multi-Doc QA			Summ.			Few-shot			Synth.		Code		
Method	NQA	Qasp	MF	HQA	2WQA	Musq	GovR	QMS	MN	TREC	TQA	SAMS	PC	PR	LCC	RB-P	Avg.
Full Cache	26.63	32.99	49.34	42.77	27.35	18.77	32.87	24.24	27.10	71.00	86.23	42.96	2.75	86.98	55.33	52.87	42.51
Budget $B = 128$																	
H2O	21.19	21.66	38.60	30.63	20.65	12.19	20.65	22.42	21.81	39.00	82.52	40.68	2.98	79.56	49.13	46.76	34.40
StreamingLLM	16.61	14.74	31.40	28.05	21.36	12.08	18.44	18.91	19.26	43.50	74.22	29.00	2.75	31.65	41.27	38.84	27.63
SnapKV	19.17	21.40	42.93	36.76	22.44	15.86	19.16	21.84	21.55	47.50	84.15	40.24	2.30	68.26	50.69	47.13	35.09
PyramidKV	20.16	21.77	43.55	36.78	23.12	14.39	19.53	22.03	21.47	51.00	84.62	40.24	2.79	70.77	50.57	46.53	35.58
Ada-KV	21.79	23.03	47.07	38.70	22.85	15.92	19.94	23.05	21.84	63.00	85.36	40.01	3.20	72.13	51.36	49.29	37.41
CriticalKV	21.55	23.27	46.67	38.95	24.95	16.85	20.83	23.14	22.16	65.00	85.53	40.02	3.48	77.81	53.58	50.98	38.42
AnDPro	24.84	25.14	47.33	39.79	23.99	17.34	20.40	**23.45**	21.94	67.50	85.59	40.62	3.19	76.94	53.77	51.55	38.96
LAQ++(8)	**25.62**	27.21	46.16	40.60	25.93	18.44	**21.60**	23.07	22.42	**70.00**	86.18	**42.03**	3.51	74.81	54.68	50.92	39.57
LORE-KV /E	24.05	26.85	**47.59**	**42.19**	26.39	19.20	21.48	22.46	**22.78**	69.50	**86.64**	41.08	3.44	84.37	**54.90**	**51.56**	**40.28**
LORE-KV /D	24.45	**27.82**	47.04	42.09	**26.74**	**19.34**	21.44	22.39	21.58	69.00	86.51	41.00	**3.83**	**84.62**	54.12	51.45	40.21
Budget $B = 256$																	
H2O	21.54	22.92	42.56	31.07	22.53	13.76	22.52	22.40	23.09	40.50	84.20	40.77	**3.41**	86.10	50.98	48.17	36.03
StreamingLLM	17.93	16.01	33.36	30.71	21.30	10.08	20.66	19.47	22.89	53.50	73.59	29.22	3.00	27.77	42.30	39.87	28.85
SnapKV	22.37	23.74	48.13	38.56	22.43	15.66	21.91	23.13	23.15	61.50	85.45	41.42	3.09	84.54	53.22	50.24	38.66
PyramidKV	20.09	24.00	47.33	38.24	22.48	16.02	21.40	22.45	22.63	63.00	84.93	40.98	3.40	82.48	52.78	49.36	38.22
Ada-KV	23.02	25.70	49.23	40.18	24.76	17.43	22.17	23.21	23.48	67.50	85.58	41.67	2.99	84.65	55.07	52.17	39.93
CriticalKV	23.70	26.66	49.00	40.05	25.12	17.86	21.99	23.35	23.45	68.00	86.19	42.07	2.92	85.81	53.81	50.42	40.03
AnDPro	25.71	28.17	**49.31**	40.04	25.81	18.68	21.66	23.64	23.63	**70.50**	**86.59**	**42.60**	2.86	85.62	55.74	52.87	40.84
LAQ++(8)	25.23	29.16	49.24	41.70	26.85	18.62	**23.73**	23.69	23.38	**70.50**	86.24	42.54	3.36	86.11	55.59	52.49	41.15
LORE-KV /E	25.88	**29.58**	48.10	**42.13**	26.83	**18.90**	23.23	**23.79**	**24.07**	**70.50**	86.31	41.09	3.27	89.09	56.15	53.12	**41.38**
LORE-KV /D	**26.22**	28.89	48.21	42.02	**27.42**	18.81	23.72	23.04	24.00	70.00	85.47	39.84	2.80	**89.20**	**56.58**	**53.44**	41.23
Budget $B = 512$																	
H2O	21.72	26.03	44.81	32.33	23.16	14.86	23.65	22.84	24.70	42.00	85.22	41.57	**3.40**	86.45	53.04	49.68	37.22
StreamingLLM	18.76	17.17	37.09	30.21	21.64	9.93	24.44	20.00	**25.57**	62.00	72.36	29.95	2.48	18.17	43.70	40.13	29.60
SnapKV	24.60	27.81	48.98	39.46	25.25	16.98	23.70	22.96	24.37	67.00	85.88	41.26	2.78	86.56	54.81	51.71	40.26
PyramidKV	23.23	27.94	48.87	40.50	24.36	16.74	23.22	23.16	24.37	67.00	85.73	41.74	3.16	85.67	54.16	50.34	40.01
Ada-KV	23.88	29.05	48.79	40.44	25.30	17.66	23.47	23.47	24.48	69.50	86.39	41.38	2.94	87.56	56.58	52.88	40.86
CriticalKV	23.83	28.67	49.13	40.18	26.18	18.19	23.52	23.57	24.28	70.00	86.39	42.16	2.82	87.55	56.51	52.64	40.98
AnDPro	24.87	30.99	49.58	41.34	27.01	18.44	23.40	23.76	25.01	**71.00**	**86.74**	**43.65**	2.83	86.23	**57.46**	53.61	41.62
LAQ++(8)	25.49	30.92	**49.72**	41.50	26.84	19.20	**25.67**	24.04	25.31	**71.00**	86.43	43.14	2.90	85.27	56.80	53.54	41.74
LORE-KV /E	25.58	**31.14**	48.98	**41.82**	**27.41**	19.12	25.46	23.97	25.05	**71.00**	86.49	41.92	2.95	**88.20**	56.85	**53.82**	**41.86**
LORE-KV /D	**25.68**	31.01	48.65	41.44	26.89	**19.25**	25.48	**24.67**	24.82	**71.00**	86.12	42.16	2.98	88.00	55.62	53.17	41.68
Budget $B = 1024$																	
H2O	23.90	28.62	46.46	37.03	24.74	15.04	25.30	23.11	25.92	46.00	85.93	41.80	**3.24**	86.57	54.46	51.01	38.70
StreamingLLM	19.42	21.69	41.75	32.40	22.18	11.18	**27.13**	21.09	**26.59**	67.00	71.79	30.11	2.88	16.57	44.82	39.76	31.02
SnapKV	25.47	29.57	49.33	40.90	25.53	**19.01**	25.94	23.89	26.21	69.50	86.48	42.10	2.98	**88.56**	55.57	51.92	41.44
PyramidKV	24.21	29.86	48.93	40.75	25.05	18.77	25.73	24.06	25.65	68.50	86.31	42.25	2.97	87.17	54.75	52.10	41.07
Ada-KV	25.60	31.15	49.02	41.47	27.07	18.77	25.40	23.84	25.92	70.50	86.55	43.07	2.54	87.14	57.27	**53.85**	41.82
CriticalKV	26.03	31.50	48.39	41.73	27.10	18.84	25.30	24.10	25.99	**71.00**	86.30	43.16	2.66	87.23	56.69	53.20	41.83
AnDPro	25.78	**32.54**	49.35	42.20	26.94	18.63	25.33	24.01	26.29	**71.00**	86.30	43.46	2.68	86.98	57.18	53.77	42.03
LORE-KV /E	25.98	31.81	49.26	43.96	27.17	18.91	26.98	**24.12**	26.41	**71.00**	86.60	42.30	2.68	86.49	**57.83**	53.77	42.20
LORE-KV /D	**26.58**	31.98	**49.61**	**44.57**	**28.19**	18.50	27.04	23.57	25.71	70.50	**87.00**	**44.07**	3.22	87.17	56.87	53.60	**42.39**

side rollout yields the largest improvement (35.87 → 39.81), while multiple trajectories provide a further gain to 40.28/40.21. Response-side queries drive the primary gain; multiple futures add further benefit, while deletion scoring and reliability weighting provide only minor refinements as discussed in Appendix F.4.

Lookahead geometry. Table 13 shows that extending the lookahead horizon matters more than simply adding trajectories: for example, at $M = 4$, increasing T from 2 to 16 improves performance from 38.96 to 40.34. For a compute-controlled comparison, we combine LAQ-style response-side lookahead with AnDPro scoring, so that AnDPro scores the identical $M \times T$ lookahead queries under the same cache budgets and selector. LORE-KV remains above this hybrid baseline across the matched quality-compute frontier (Table 13b), showing that its gains are not explained by additional lookahead compute alone.

Generalization across backbones. The same operating-point trend holds on Qwen2.5-14B-Instruct and Ministral-3-8B-Instruct-2512 (Tables 11 and 12). At $B = 128$, LORE-KV improves over the strongest baseline by 2.75 points on Qwen2.5 and 0.56 points on Ministral-3, while the margins become very small by $B = 1024$. The smaller gain on Ministral-3 likely reflects its long-context

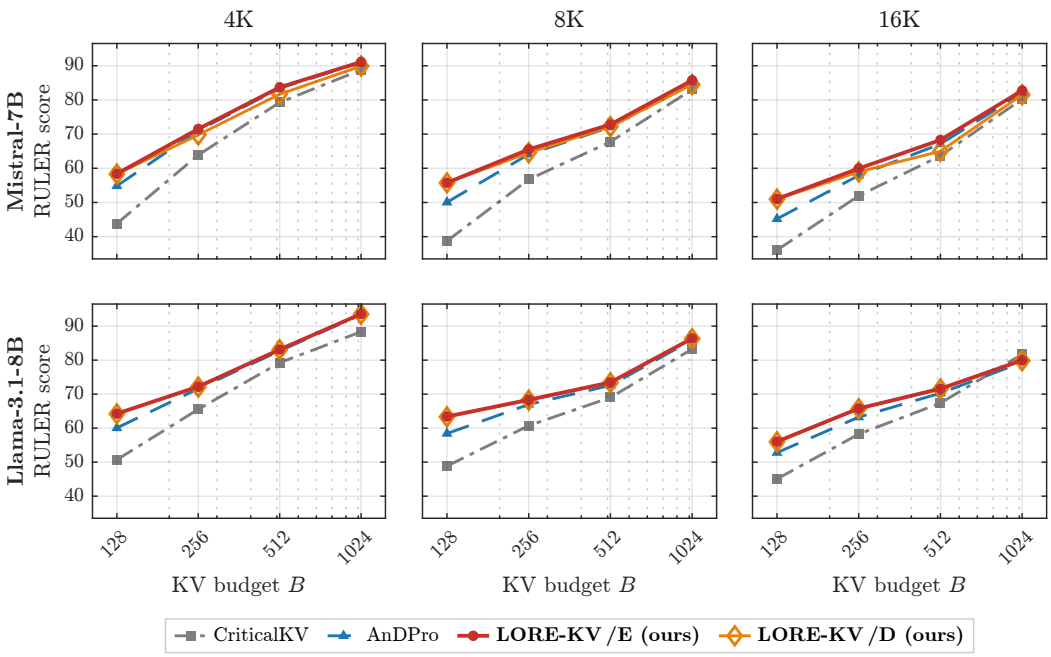


Figure 3: RULER accuracy averaged over 13 tasks at 4K, 8K, and 16K context lengths. Per-task results are given in Tables 7 and 10.

training and position-aware query scaling, which reduce the prompt-to-future query mismatch that LORE-KV is designed to correct (Appendix E.3).

Hybrid-attention backbones. LORE-KV also generalizes to Qwen3.5-9B and Gemma-4-12B-it, consistently outperforming the strongest baseline across cache budgets (Table 2), with the largest gains under aggressive compression. Detailed results are provided in Appendix E.2 and Tables 8-9.

Cache efficiency. The accuracy gains also translate into lower cache requirements. Table 16 shows that matching LORE-KV /E requires the strongest baseline to retain about 208 tokens, corresponding to a 38.5% cache saving, with a 31.1% saving still observed at $B = 512$. Figure 13 confirms the same trend across the full budget range, showing that LORE-KV improves the overall quality-memory trade-off rather than only isolated operating points.

Table 2: **LongBench on hybrid-attention backbones.** Eviction is applied only to the KV-growing layers.

	Qwen3.5-9B				Gemma-4-12B-it			
Method	B=128	256	512	1024	B=128	256	512	1024
Full Cache	52.97				27.27			
CriticalKV	46.30	48.30	49.58	50.31	19.74	21.53	22.45	23.22
AnDPro	46.89	48.72	50.08	50.42	20.21	21.92	22.73	23.47
LORE-KV /E	48.49	50.05	51.12	51.79	20.85	22.63	23.69	24.44
LORE-KV /D	**48.57**	**50.22**	**51.16**	**51.92**	**20.93**	**22.76**	**23.81**	**24.59**

4.2 Evaluation on Needle-in-a-Haystack (NIAH)

LORE-KV also improves retrieval under aggressive compression. On Llama-3.1-8B-Instruct at $B = 1024$, LORE-KV /E and LORE-KV /D tie for the best exact-match score (0.977), outperforming CriticalKV (0.955) and AnDPro (0.932), with the same trend under word recall.

As Figure 8 shows (setup in Appendix E.5), the gains come from boundary-depth needles (0% and 100%) at 16K, which the baselines miss and LORE-KV retrieves. Figure 11 further shows that LORE-KV retains answer-bearing tokens more often, linking the retrieval gains to its future-query-aware token selection.

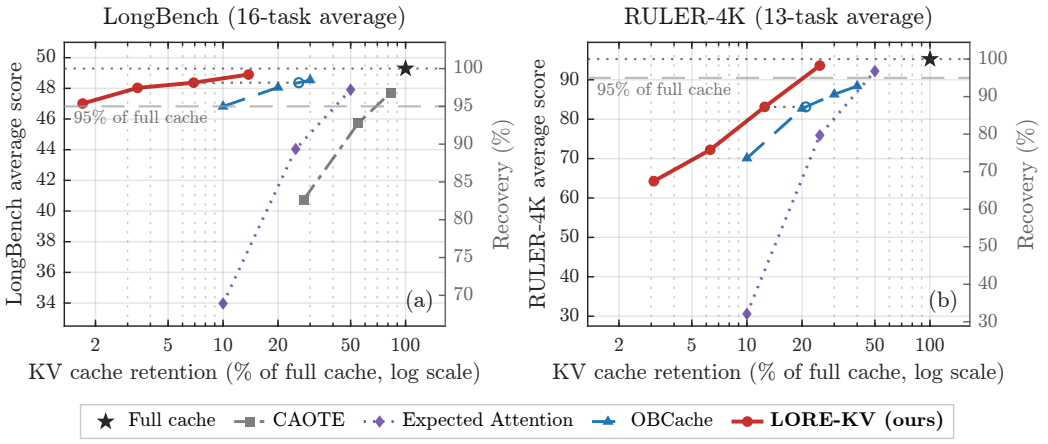


Figure 4: Accuracy-memory trade-off on Llama-3.1-8B-Instruct. Higher and further left indicates a better trade-off.

4.3 Evaluation on RULER: Sweeping Context Length Across Fixed Tasks

Context and budget scaling. RULER evaluates the same 13 tasks across 4K–16K contexts with fixed lookahead cost, where LORE-KV /E achieves the best overall average and wins 23 of 24 settings, with the largest gains on aggregation-style tasks (Figure 3; Tables 7 and 10). The advantage is largest under aggressive compression and longer contexts: on Mistral at $B = 128$, the gain over AnDPro grows from 3.55 points at 4K to 5.85 at 16K, while becoming small at $B = 1024$. With fixed lookahead cost, this shows that future-query estimation is most valuable when the cache is tight and prompt-side queries are less representative of later decoding needs (Figure 6). Table 15 further shows that multiple futures outperform a single rollout, while LORE-KV /D closely tracks LORE-KV /E.

Quality-memory trade-off. In the fixed 4K setting, Figure 4 shows that LORE-KV /E achieves higher quality at lower retention, outperforming OBCache (Gu et al., 2025) at 12.5% versus 20% retention and Expected Attention (Devoto et al., 2026) at the same 1,024-token budget. The Expected Attention comparison is contextual because it is query-agnostic by design, but follows the mismatch trend in Figure 6.

4.4 Memory and Latency

At a fixed cache budget B, all methods have the same steady-state decoding memory and per-token cost; LORE-KV adds a one-time lookahead and scoring overhead from M trajectories of length T, while reliability weighting requires no additional decoding, giving LORE-KV /E and LORE-KV /D the same cost profile, as detailed in Algorithm 1. Table 13b shows that LORE-KV remains above the strongest baseline at every matched lookahead-token budget, with computational overhead detailed in Appendix C.4. Figure 7 reports the latency-quality trade-off; stage-wise LORE-KV latency is averaged over two seeds, and because the figure and Table 3 use separate timing runs, their totals may differ slightly. With fixed (M, T), lookahead and scoring costs are largely independent of B; the lower latency at B=1024 is consistent with runtime variation. Figure 2 further shows that LORE-KV reaches the same accuracy with a smaller retained cache.

4.5 Mechanism Analysis: Locating the Gain

We analyze whether LORE-KV benefits from better alignment with future decoding queries, using retrospectively observed $\mathcal{Q}_{\text{future}}$ (Appendices B and F.1). Figure 1 shows that response-side lookahead improves LongBench coverage of $\mathcal{Q}_{\text{future}}$ from 0.785 for the prompt window to 0.802 at M=2 and 0.811 at M=8, saturating near M=4; on this geometric diagnostic a single rollout (0.786) barely separates from the prompt window, so the coverage gain comes from spanning several plausible futures rather than from one surrogate continuation. The M-ablations in Tables 14 and 15 measure the corresponding end-to-end benefit, which is already large at M=1; coverage is

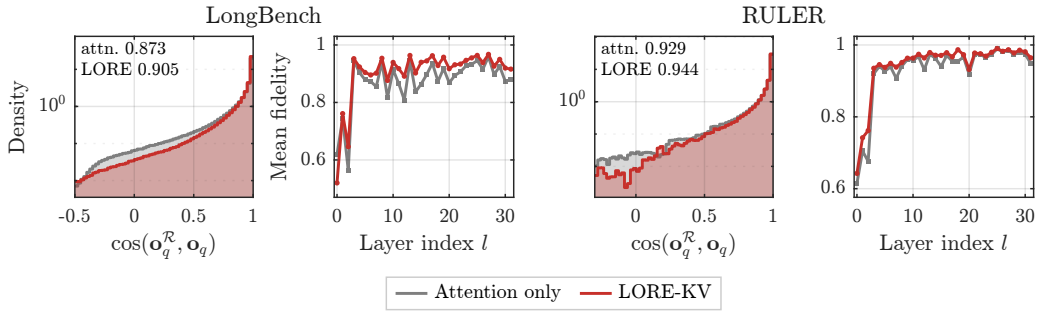


Figure 5: **Attention-output fidelity.** Similarity between retained- and full-cache outputs on future queries across layers. Mistral-7B-Instruct-v0.2, $B = 128$.

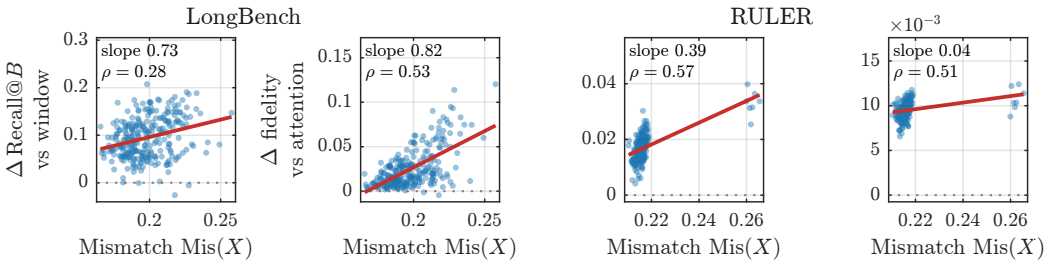


Figure 6: **Gain versus query mismatch.** Per-example gains in oracle retention and output fidelity with increasing prompt-to-future mismatch. Mistral-7B-Instruct-v0.2, $B = 128$, $M = 4$.

a nearest-neighbor statistic over query directions and therefore tracks the geometry of the estimated query set rather than the realized deletion cost (Appendix F.2). Correspondingly, output fidelity increases from 0.873 to 0.905 on LongBench and from 0.929 to 0.944 on RULER (Figure 5), with gains distributed across layers and heads (Figure 12).

LORE-KV's gains increase with prompt-to-future query mismatch. Defining $\mathrm{Mis}(X) = 1 - \mathrm{Cov}(\mathcal{Q}_{\mathrm{future}}; \{\mathcal{Q}_{\mathcal{W}}^{l,h}\})$, Figure 6 shows positive Spearman correlations with oracle-retention recall ($\rho = 0.28/0.57$) and output fidelity ($\rho = 0.53/0.51$) on LongBench/RULER, consistent with the budget- and context-length trends in Section 4.3. Figure 10 further shows that, with prefill, particles, and selection fixed, LORE-KV better matches the future-query oracle and preserves full-cache output fidelity than CriticalKV, AnDPro, and the prompt-only control. Under identical trajectories, the two reliability variants have nearly identical averages, so entropy and disagreement weighting act as robustness corrections on top of the future-query reference rather than as a second source of gain (Appendix F.4). Additional diagnostics are provided in Appendix F, Figures 9, 11, 12, and 13.

5 Conclusion

At a broader level, LORE-KV treats cache eviction as a future-use prediction problem: cached tokens should be retained based on how useful they are likely to be during generation, not only on how important they appeared during the prompt. **Limitations.** LORE-KV still incurs one-time lookahead overhead, makes irreversible eviction decisions, and does not explicitly model cross-head or cross-layer interactions. These limitations motivate adaptive lookahead, recoverable or dynamically updated caches, and globally coordinated retention. **Future work.** More broadly, this perspective points toward dynamic inference-time memory systems that adapt retention as the model's computational needs evolve. Since the benefits of future-aware eviction are strongest under tighter cache budgets, such systems could decide when lookahead is worth invoking, allocate extra lookahead only when future-query uncertainty is high, and revise cache decisions during generation. A practical extension is to use a cheap proxy for prompt-to-future mismatch to skip lookahead when prompt-side queries are already sufficiently informative. This would move beyond a fixed prompt-time eviction policy toward memory that tracks what information is likely to matter next.

AI USE STATEMENT

ChatGPT by OpenAI and Claude by Anthropic were used for minor editing, proofreading, and resolving LaTeX formatting issues in this document. Codex by OpenAI was used to assist with routine code writing, plotting, table generation, and code refactoring in the implementation of LORE-KV. The authors reviewed and verified all AI-assisted content and take full responsibility for the paper.

ETHICS STATEMENT

This work uses publicly available models and benchmarks and involves no human subjects, personally identifiable information, or new data collection.

REPRODUCIBILITY STATEMENT

LORE-KV is training-free and requires no modification to model weights. The methodology and algorithm are provided in Section 3 and Algorithm 1; hyperparameters, implementation details, evaluation settings, and ablation protocols are documented in Table 4, Appendices C.1, E.4, and F.1, and Tables 5 and 14-15. All methods are evaluated under a matched implementation, with backbone-specific verification described in Appendix E.3.

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. pp. 3119–3137, August 2024. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Payman Behnam, Yaosheng Fu, Ritchie Zhao, Po-An Tsai, Zhiding Yu, and Alexey Tumanov. RocketKV: Accelerating long-context LLM inference via two-stage KV cache compression. *arXiv preprint arXiv:2502.14051*, 2025.

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

Zefan Cai, Wen Xiao, Hanshi Sun, Cheng Luo, Yikai Zhang, Ke Wan, Yucheng Li, Yeyang Zhou, Li-Wen Chang, Jiuxiang Gu, Zhen Dong, Anima Anandkumar, Abedelkadir Asi, and Junjie Hu. R-KV: Redundancy-aware KV cache compression for reasoning models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=2jwAjomEDB.

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, 2025b. URL https://arxiv.org/abs/2406.02069.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Hung-Yueh Chiang, Yash Akhauri, Xilai Dai, Huiqiang Jiang, Yucheng Li, Luis Ceze, Kai-Chiang Wu, and Mohamed S. Abdelfattah. xKV: Cross-layer KV-cache compression via aligned singular vector extraction, 2026. URL https://arxiv.org/abs/2503.18893.

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, 2026a.

Xingchi Chen, Peiyuan Zong, Ziqiang Gao, Qing Li, Yong Jiang, Fa Zhu, and Hui Li. ContrastKV: Robust KV cache eviction via contrastive signal fusion for multi-query generalization. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9216–9229, San Diego, California, United States, July 2026b. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.417. URL https://aclanthology.org/2026.acl-long.417/.

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 the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18476–18499, 2024.

Alessio Devoto, Maximilian Jeblick, and Simon Jégou. Expected attention: KV cache compression by estimating attention from future queries distribution, 2026. URL https://openreview.net/forum?id=VmojW15eRc.

Shichen Dong, Wen Cheng, Jiayu Qin, and Wei Wang. QAQ: Quality adaptive quantization for LLM KV cache. *arXiv preprint arXiv:2403.04643*, 2024.

Yuan Feng, Haoyu Guo, JunLin Lv, S Kevin Zhou, and Xike Xie. Taming the fragility of KV cache eviction in LLM inference. *arXiv preprint arXiv:2510.13334*, 2025a.

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, 2025b. URL https://arxiv.org/abs/2407.11550.

Yuan Feng, Junlin Lv, Haoyu Guo, Yukun Cao, S Kevin Zhou, and Xike Xie. Criticalkv: Optimizing kv cache eviction from an output perturbation perspective, 2026. URL https://arxiv.org/abs/2502.03805.

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. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=FJFVmeXusW.

Gabriel Garcia. Protection is (nearly) all you need: Structural protection dominates scoring in globally capped KV eviction. *arXiv preprint arXiv:2605.18053*, 2026.

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*, volume 2024, pp. 22975–22988, 2024.

Zijie Geng, Jie Wang, Ziqi Liu, Feng Ju, Yiming Li, Xing Li, Mingxuan Yuan, Jianye HAO, Defu Lian, Enhong Chen, and Feng Wu. Accurate KV cache eviction via anchor direction projection for efficient LLM inference. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Tdl89SZItB.

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco

Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L.

Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The Llama 3 herd of models. 2024. URL https://arxiv.org/abs/2407.21783.

Yuzhe Gu, Xiyu Liang, Jiaojiao Zhao, and Enmao Diao. OBCache: Optimal brain KV cache pruning for efficient long-context LLM inference. *arXiv preprint arXiv:2510.07651*, 2025.

Zhiyu Guo, Hidetaka Kamigaito, and Taro Watanabe. Attention score is not all you need for token importance indicator in KV cache reduction: Value also matters. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21158–21166, 2024.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun S Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. *Advances in Neural Information Processing Systems*, 37:1270–1303, 2024.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models?, 2024. URL https://arxiv.org/abs/2404.06654.

Junhao Hu, Wenrui Huang, Weidong Wang, Zhenwen Li, Tiancheng Hu, Zhixia Liu, Xusheng Chen, Tao Xie, and Yizhou Shan. RaaS: Reasoning-aware attention sparsity for efficient LLM reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 2577–2590, 2025.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. 2023. URL https://arxiv.org/abs/2310.06825.

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

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=JFygzwx8SJ.

Jang-Hyun Kim, Dongyoon Han, and Sangdoo Yun. Fast KVzip: Efficient and accurate LLM inference with gated KV eviction, 2026. URL https://arxiv.org/abs/2601.17668.

Adrian Łańcucki, Konrad Staniszewski, Piotr Nawrot, and Edoardo Ponti. Inference-time hyper-scaling with KV cache compression. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=8ZiElzQxf1.

Xing Li, Zeyu Xing, Yiming Li, Linping Qu, Hui-Ling Zhen, Wulong Liu, Yiwu Yao, Sinno Jialin Pan, and Mingxuan Yuan. KVTuner: Sensitivity-aware layer-wise mixed-precision KV cache quantization for efficient and nearly lossless LLM inference. *arXiv preprint arXiv:2502.04420*, 2025a.

Xuelin Li, Xiangqi Jin, and Linfeng Zhang. GraphKV: Breaking the static selection paradigm with graph-based KV cache eviction, 2025b. URL https://arxiv.org/abs/2509.00388.

Yubo Li and Yidi Miao. CONF-KV: Confidence-aware KV cache eviction with mixed-precision storage for long-horizon LLM. *arXiv preprint arXiv:2605.24786*, 2026.

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. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. MiniCache: KV cache compression in depth dimension for large language models, 2024a. URL https://arxiv.org/abs/2405.14366.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, Baptiste Rozière, Baudouin De Monicault, Clémence Lanfranchi, Corentin Barreau, Cyprien Courtot, Daniele Grattarola, Darius Dabert, Diego de las Casas, Elliot Chane-Sane, Faruk Ahmed, Gabrielle Berrada, Gaëtan Ecrepont, Gauthier Guinet, Georgii Novikov, Guillaume Kunsch, Guillaume Lample, Guillaume Martin, Gunshi Gupta, Jan Ludziejewski, Jason Rute, Joachim Studnia, Jonas Amar, Joséphine Delas, Josselin Somerville Roberts, Karmesh Yadav, Khyathi Chandu, Kush Jain, Laurence Aitchison, Laurent Fainsin, Léonard Blier, Lingxiao Zhao, Louis Martin, Lucile Saulnier, Luyu Gao, Maarten Buyl, Margaret Jennings, Marie Pellat, Mark Prins, Mathieu Poirée, Mathilde Guillaumin, Matthieu Dinot, Matthieu Futeral, Maxime Darrin, Maximilian Augustin, Mia Chiquier, Michel Schimpf, Nathan Grinsztajn, Neha Gupta, Nikhil Raghuraman, Olivier Bousquet, Olivier Duchenne, Patricia Wang, Patrick von Platen, Paul Jacob, Paul Wambergue, Paula Kurylowicz, Pavankumar Reddy Muddireddy, Philomène Chagniot, Pierre Stock, Pravesh Agrawal, Quentin Torroba, Romain Sauvestre, Roman Soletskyi, Rupert Menneer, Sagar Vaze, Samuel Barry, Sanchit Gandhi, Siddhant Waghjale, Siddharth Gandhi, Soham Ghosh, Srijan Mishra, Sumukh Aithal, Szymon Antoniak, Teven Le Scao, Théo Cachet, Theo Simon Sorg, Thibaut Lavril, Thiziri Nait Saada, Thomas Chabal, Thomas Foubert, Thomas Robert, Thomas Wang, Tim Lawson, Tom Bewley, Tom Bewley, Tom Edwards, Umar Jamil, Umberto Tomasini, Valeriia Nemychnikova, Van Phung, Vincent Maladière, Virgile Richard, Wassim Bouaziz, Wen-Ding Li, William Marshall, Xinghui Li, Xinyu Yang, Yassine El Ouahidi, Yihan Wang, Yunhao Tang, and Zaccharie Ramzi. Ministral 3, 2026a. URL https://arxiv.org/abs/2601.08584.

Tengxuan Liu, Shiyao Li, Jiayi Yang, Tianchen Zhao, Feng Zhou, Xiaohui Song, Guohao Dai, Shengen Yan, Huazhong Yang, and Yu Wang. PM-KVQ: Progressive mixed-precision KV cache quantization for long-CoT LLMs. *arXiv preprint arXiv:2505.18610*, 2025.

Zedong Liu, Xinyang Ma, Dejun Luo, Hairui Zhao, Bing Lu, Wenjing Huang, Yida Gu, Xingchen Liu, Zheng Wei, Jinyang Liu, Dingwen Tao, and Guangming Tan. KVServe: Service-aware KV cache compression for communication-efficient disaggregated LLM serving, 2026b. URL https://arxiv.org/abs/2605.13734.

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. *Advances in Neural Information Processing Systems*, 36:52342–52364, 2023.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. *arXiv preprint arXiv:2402.02750*, 2024b.

Da Ma, Lu Chen, Situo Zhang, Yuxun Miao, Su Zhu, Zhi Chen, Hongshen Xu, Hanqi Li, Shuai Fan, Lei Pan, and Kai Yu. Compressing KV cache for long-context LLM inference with inter-layer attention similarity. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 16407–16411, 2026. doi: 10.1109/ICASSP55912.2026.11464826.

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

Luca Moschella, Laura Manduchi, and Ozan Sener. Learning to evict from Key-Value cache. *arXiv preprint arXiv:2602.10238*, 2026.

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.

Junyoung Park, Dalton Jones, Matthew J Morse, Raghavv Goel, Mingu Lee, and Christopher Lott. KeyDiff: Key similarity-based KV cache eviction for long-context LLM inference in resource-constrained environments. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=uBaFH7aQnC.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=EQgEMAD4kv.

Akshat Ramachandran, Marina Neseem, Charbel Sakr, Rangharajan Venkatesan, Brucek Khailany, and Tushar Krishna. ThinKV: Thought-adaptive KV cache compression for efficient reasoning models. *arXiv preprint arXiv:2510.01290*, 2025.

Akshat Sharma, Hangliang Ding, Jianping Li, Neel Dani, and Minjia Zhang. MiniKV: Pushing the limits of LLM inference via 2-bit layer-discriminative KV cache. *arXiv preprint arXiv:2411.18077*, 2024.

Yiqun Shen, Song Yuan, Zhengze Zhang, Xiaoliang Wang, Daxin Jiang, and Nguyen Cam-Tu. LAVa: Layer-wise KV cache eviction with dynamic budget allocation, 2025. URL https://arxiv.org/abs/2509.09754.

Konrad Staniszewski and Adrian Łańcucki. KV cache transform coding for compact storage in LLM inference. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=aNVKROYpLB.

Yi Su, Zhenxu Tian, Dan Qiao, Yuechi Zhou, Juntao Li, and Min Zhang. LongFlow: Efficient KV cache compression for reasoning models. *arXiv preprint arXiv:2603.11504*, 2026.

Zunhai Su, Zhe Chen, Wang Shen, Hanyu Wei, Linge Li, Huangqi Yu, and Kehong Yuan. RotateKV: Accurate and robust 2-bit KV cache quantization for LLMs via outlier-aware adaptive rotations. *arXiv preprint arXiv:2501.16383*, 2025.

Chenxia Tang, Jianchun Liu, Hongli Xu, and Liusheng Huang. Adaptive KV-cache compression without manually setting budget. *arXiv preprint arXiv:2509.03136*, 2025.

Ziyao Tang, Pengkun Jiao, Xinhang Chen, Wei Liu, Shiyong Li, and Jingjing Chen. Predicting future utility: Global combinatorial optimization for task-agnostic KV cache eviction. *arXiv preprint arXiv:2602.08585*, 2026.

Zhenxu Tian, Yi Su, Juntao Li, and Min Zhang. Where matters more than what: Decoding-aligned KV cache compression via position-aware pseudo-queries, 2026. URL https://arxiv.org/abs/2603.11564.

Guangtao Wang, Shubhangi Upasani, Chen Wu, Darshan Gandhi, Jonathan Li, Changran Hu, Bo Li, and Urmish Thakker. LLMs know what to drop: Self-attention guided KV cache eviction for efficient long-context inference. *arXiv preprint arXiv:2503.08879*, 2025a.

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 Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 34158–34174, Suzhou, China, November 2025b. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1732. URL https://aclanthology.org/2025.emnlp-main.1732/.

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, 2024. URL https://arxiv.org/abs/2407.08454.

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

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2024a. URL https://arxiv.org/abs/2412.15115.

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

Xintong Yang, Hao Gu, Binxing Xu, Lujun Li, Bei Liu, Jiacheng Liu, Qiyuan Zhu, Sirui Han, and Yike Guo. IndexMem: Learned KV-cache eviction with latent memory for long-context LLM inference. *arXiv preprint arXiv:2605.25475*, 2026.

Haoyue Zhang, Hualei Zhang, Xiaosong Ma, Jie Zhang, and Song Guo. LazyEviction: Lagged KV eviction with attention pattern observation for efficient long reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36335–36352, 2026a.

Tao Zhang, Ziqian Zeng, Hao Peng, Huiping Zhuang, and Cen Chen. MixKVQ: Query-aware mixed-precision KV cache quantization for long-context reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7189–7204, 2026b.

Yanqi Zhang, Yuwei Hu, Runyuan Zhao, John C. S. Lui, and Haibo Chen. DiffKV: Differentiated memory management for large language models with parallel KV compaction. In *Proceedings of the ACM SIGOPS 31st Symposium on Operating Systems Principles*, SOSP '25, pp. 431–445, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400718700. doi: 10.1145/3731569.3764810. URL https://doi.org/10.1145/3731569.3764810.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models, 2023. URL https://arxiv.org/abs/2306.14048.

Xiabin Zhou, Wenbin Wang, Minyan Zeng, Jiaxian Guo, Xuebo Liu, Li Shen, Min Zhang, and Liang Ding. DynamicKV: Task-aware adaptive KV cache compression for long context LLMs. *arXiv preprint arXiv:2412.14838*, 2024.

Appendix Contents

A Extended Related Work **18**

A.1 Prompt-side eviction and budget allocation 18

A.2 Future-aware and output-aware eviction 18

A.3 Quantization and representation compression 18

A.4 Decode-time and serving-oriented memory management 18

B Background: The KV Cache and the Eviction Problem **18**

C Method Details **19**

C.1 Implementation: Protected Set, Grouped-Query Attention, Fused Kernels 19

C.2 Relation to Prior Surrogate-Query and Output-Aware Methods 19

C.3 Reliability Weighting: Derivations, Calibration, and Edge Cases 19

C.4 Computational Overhead 21

D Plug-In Components and the Ablation Protocol **22**

E Experimental Setup and Additional Results **26**

E.1 Protocol, Datasets and Baselines 26

E.2 Hybrid-Attention Backbones: Where Eviction Applies 27

E.3 LongBench Results on Additional Backbones 29

E.4 RULER: Protocol and Per-Task Results 30

E.5 NIAH Grid 32

F Mechanism Analysis **32**

F.1 Measurement Protocol 32

F.2 Future-Query Coverage 34

F.3 Retention-Order Shift 34

F.4 Reliability Weighting: Robustness Beyond the Query Reference 35

F.5 Retained-Token Composition 36

F.6 Layer- and Head-Level Structure 37

F.7 Budget Equivalence in Behavioral Fidelity 37

G The Two Reliability Variants in the $(\beta, \gamma, \lambda_D)$ Plane **37**

A Extended Related Work

A.1 Prompt-side eviction and budget allocation

Beyond the representative methods discussed in Section 2.1, Scissorhands uses accumulated attention statistics (Liu et al., 2023), TOVA uses current-query attention (Oren et al., 2024), FastGen applies head-specific retention patterns (Ge et al., 2024), and CAKE improves layer-wise budget allocation (Qin et al., 2025). HeadKV, DynamicKV, SAGE-KV, LAVa, and LU-KV further adapt retention across heads, layers, or token groups (Fu et al., 2025; Zhou et al., 2024; Wang et al., 2025a; Shen et al., 2025; Tang et al., 2026). Although these approaches differ substantially in scoring and allocation, their eviction signals are primarily derived from queries or statistics available on the prompt side.

A.2 Future-aware and output-aware eviction

Several additional methods address robustness to unknown future queries. KVP and globally calibrated retention methods learn predictors of future token utility (Moschella et al., 2026; Bui et al., 2026), whereas KVzip, FastKVzip, and ContrastKV seek query-agnostic or multi-query robustness (Kim et al., 2025; Chen et al., 2026b; Kim et al., 2026). CriticalKV incorporates value information into token importance (Feng et al., 2026), while CAOTE scores tokens according to attention-output error (Goel et al., 2025). These methods motivate the two axes considered by LORE-KV: the query states against which importance is evaluated and the output distortion induced by deletion, but generally address one axis independently or replace explicit response-side sampling with learned, analytical, or query-agnostic proxies.

A.3 Quantization and representation compression

KV-cache quantization methods include QAQ, MiKV, MiniKV, RotateKV, KVTuner, MixKVQ, PM-KVQ, and DiffKV (Dong et al., 2024; Yang et al., 2024b; Sharma et al., 2024; Su et al., 2025; Li et al., 2025a; Zhang et al., 2026b; Liu et al., 2025; Zhang et al., 2025). GEAR, KVMerger, xKV, inter-layer compression, and transform-coding approaches instead exploit low-rank, mergeable, cross-layer, or coded cache representations (Kang et al., 2024; Wang et al., 2024; Chang et al., 2026; Ma et al., 2026; Staniszewski & Łańcucki, 2026).

A.4 Decode-time and serving-oriented memory management

KVServe targets communication-efficient KV-cache serving (Liu et al., 2026b). DMS, ThinKV, RaaS, LazyEviction, R-KV, RocketKV, LongFlow, and CONF-KV instead control cache growth or retention during extended decoding (Łańcucki et al., 2025; Ramachandran et al., 2025; Hu et al., 2025; Zhang et al., 2026a; Cai et al., 2025a; Behnam et al., 2025; Su et al., 2026; Li & Miao, 2026). These methods address a different stage of the memory-management problem and can in principle be combined with a future-aware prompt-time eviction criterion such as LORE-KV.

B Background: The KV Cache and the Eviction Problem

KV caching and memory cost. For L layers, H KV heads, head dimension d, sequence length n, batch size b, and p bytes per element, the KV cache requires $\mathrm{Mem}_{\mathrm{KV}} = 2bLHdnp$ bytes, where the factor 2 accounts for keys and values; hence, its memory grows linearly with context length. For example, a 70B-class model with $L = 80$, $H = 8$, $d = 128$, $n = 131{,}072$ (128K tokens), $b = 1$, and $p = 2$ bytes (BF16) requires approximately 42.9 GB of KV-cache memory, about 31% of the roughly 140 GB required by its BF16 model weights. Given a cache budget $B \ll n$, let $\mathcal{R}^{l,h} \subseteq \{1, \ldots, n\}$ denote the positions retained by layer l and KV head h, with $|\mathcal{R}^{l,h}| \leq B_{l,h}$. Defining $\delta(\mathbf{q}, \mathcal{R}) = \mathrm{dist}(\mathrm{output}(\mathbf{q}; \text{full cache}), \mathrm{output}(\mathbf{q}; \text{retained cache}))$ as the distortion induced by eviction for a future query $\mathbf{q}$, the ideal retained cache is $\mathcal{R}^\star = \arg\min_{\{\mathcal{R}^{l,h}:|\mathcal{R}^{l,h}|\leq B_{l,h}\}} \mathbb{E}_{\mathbf{q}\sim\mathcal{Q}_{\mathrm{future}}}[\delta(\mathbf{q}, \mathcal{R})]$. Thus, eviction seeks to retain the KV entries that best preserve full-cache behavior over future decoding queries; Section 4.5 evaluates this approximation against the actual future queries observed after generation.

C METHOD DETAILS

This section provides the implementation and computational details of LORE-KV, including protected-token handling, grouped-query attention, reliability weighting, and efficient perturbation scoring. Algorithm 1 summarizes the full LORE-KV pipeline. Starting from the prefilled KV cache, LORE-KV samples short lookahead continuations to obtain plausible future queries, reliability-weights these trajectories, scores candidate KV entries by their projected deletion effect, and retains the highest-scoring entries under the cache budget.

C.1 IMPLEMENTATION: PROTECTED SET, GROUPED-QUERY ATTENTION, FUSED KERNELS

Protected positions. The optional set $\mathcal{B}$ contains prompt-structure markers, chat-template delimiters (`<|system|>`, `<|user|>`, `<|assistant|>`, `[INST]`, `[/INST]`), tool-call markers, and document separators identified directly from the tokenized prompt and always retained; if none are present, $\mathcal{B} = \emptyset$. Following prior work (Xiao et al., 2024; Li et al., 2024), this prevents instructions and prompt boundaries from being removed under aggressive eviction.

Grouped-query attention. Under GQA/MQA, multiple query heads share each KV head. In our implementation, we therefore use all query heads associated with KV head h, denoted by $\mathcal{G}(h)$, to score each cached pair $(\mathbf{k}_i^{l,h}, \mathbf{v}_i^{l,h})$ by aggregating their deletion effects from Eq. 5. Throughout, H denotes the number of KV heads, and we assign cache budgets at the KV-head level.

Fused attention kernels. Since kernels such as FlashAttention do not materialize the full $n \times n$ attention matrix required by perturbation scoring, we explicitly compute attention only for the scoring-query set $\mathcal{Q}_{\text{score}}^{l,h}$, whose size $|\mathcal{Q}_{\text{score}}^{l,h}| = w + MT \ll n$, against the cached keys. The fused kernel therefore remains unchanged for the rest of prefill and final decoding, making eviction scoring a small side computation that never requires full-matrix attention on the main inference path.

C.2 RELATION TO PRIOR SURROGATE-QUERY AND OUTPUT-AWARE METHODS

The $M{=}1$ setting already captures most of LORE-KV's gain, demonstrating that correcting the prompt-to-future query mismatch is the primary driver of performance. Increasing to $M{>}1$ further improves robustness by covering multiple plausible futures and reducing sensitivity to any single rollout through reliability-weighted aggregation, while $\lambda_L{=}0$ provides the prompt-only control. Together, these results show that LORE-KV is both effective with minimal lookahead and increasingly robust as future-query coverage expands.

C.3 RELIABILITY WEIGHTING: DERIVATIONS, CALIBRATION, AND EDGE CASES

This section gives the statistical interpretation and implementation details of the reliability weighting in Section 3.3.

Centered typicality and scale. For a sampled token $y_u \sim q(\cdot \mid h_u)$, the centered score of Eq. 2 satisfies

$$\mathbb{E}[c_u \mid h_u] = \sum_y q(y \mid h_u) \log q(y \mid h_u) + H(q(\cdot \mid h_u)) = 0.$$

Hence $\mathbb{E}[\bar{c}_m] = 0$, so $\bar{c}_m$ measures how typical a sampled trajectory is relative to the model's local uncertainty. Positive values indicate that the sampled tokens are more likely than expected, while $\bar{H}_m$ separately captures how uncertain the model is along that trajectory.

The identity

$$\frac{1}{T}\sum_{u=1}^{T} \log q(y_u^{(m)} \mid h_u^{(m)}) = \bar{c}_m - \bar{H}_m,$$

also makes the parameterization of Eq. 3 transparent. Uniform weighting corresponds to $(\beta, \gamma, \lambda_D) = (0, 0, 0)$, centered-typicality-only weighting to $(1, 0, 0)$, and raw log-probability weighting to $(1, 1, 0)$. The reported reliability settings are given in Table 4.

Because $\{c_u\}$ are martingale differences, for $u < v$,

$$\mathbb{E}[c_u c_v] = \mathbb{E}[c_u \, \mathbb{E}[c_v \mid h_v]] = 0.$$

Algorithm 1 LORE-KV eviction. Lookahead trajectories are used only for scoring and discarded before decoding, leaving model weights and decoding unchanged. Scoring uses only $w + MT \ll n$ queries.

Require: frozen LLM f_θ; prompt $X = (x_1, \dots, x_n)$; window $\mathcal{W}$ of size w; particles M, horizon T; layer budgets $\{B_l\}_{l=1}^{L}$; reliability coefficients $(\beta, \gamma, \lambda_D)$, mixing weight λ_L, guard ϵ; plug-in switches $(\alpha, \kappa, \tau, \lambda_k, \lambda_v)$, all 0 for the core method (Table 4)
Ensure: compressed KV cache $(\mathbf{K}^\star, \mathbf{V}^\star)$

$\big(\{\mathbf{K}^{l,h}, \mathbf{V}^{l,h}\}, \{\mathcal{Q}_{\mathcal{W}}^{l,h}\}\big) \leftarrow \mathrm{Prefill}(f_\theta, X)$
$\mathcal{P} \leftarrow \{1\} \cup \mathcal{W} \cup \mathcal{B}$; $\mathcal{C} \leftarrow \{1, \dots, n\} \setminus \mathcal{P}$ ▷ *protected and candidate positions*
Stage 1: Monte Carlo future-query estimation *(Sec. 3.2)*
for $m = 1, \dots, M$ **in parallel do**
  $Y^{(m)} \sim f_\theta(\cdot \mid X)$ with $|Y^{(m)}| = T$ ▷ *one plausible continuation*
  cache $\mathcal{Q}_{\mathrm{look}}^{(m),l,h}$ for all (l, h), and $\{c_u^{(m)}, H_u^{(m)}\}_{u=1}^{T}$ ▷ *kept for scoring and weighting*
end for
$\mathcal{Q}_{\mathrm{score}}^{l,h} \leftarrow \mathcal{Q}_{\mathcal{W}}^{l,h} \cup \bigcup_{m=1}^{M} \mathcal{Q}_{\mathrm{look}}^{(m),l,h}$ ▷ $w + MT$ *scoring queries in total*
Stage 2: particle reliability weighting *(Sec. 3.3)*
$\hat{D}_m \leftarrow \big(D_m - \bar{D}\big) / \mathrm{sd}(D)$ for $M \geq 3$, else $\hat{D}_m \leftarrow 0$ ▷ *drift from the ensemble*
$\bar{c}_m \leftarrow \frac{1}{T} \sum_u c_u^{(m)}$, $\bar{H}_m \leftarrow \frac{1}{T} \sum_u H_u^{(m)}$ ▷ *per-step means, not sequence totals*
$R_m \leftarrow \beta\, \bar{c}_m - \gamma\, \bar{H}_m - \lambda_D\, \hat{D}_m$; $\boldsymbol{\omega} \leftarrow \mathrm{softmax}\big((R_m)_{m=1}^{M}\big)$ ▷ *soft trajectory weights*
Stage 3: projected counterfactual scoring and selection *(Secs. 3.4–3.5)*
for layer $l = 1, \dots, L$ **do**
  for KV head $h = 1, \dots, H$ **do** ▷ *under GQA, over the query group* $\mathcal{G}(h)$
    for candidate $i \in \mathcal{C}$ and scoring query $\mathbf{q} \in \mathcal{Q}_{\mathrm{score}}^{l,h}$ **do**
      $(a_{q,i}^{l,h}, \mathbf{o}_q^{l,h}) \leftarrow \mathrm{Attn}(\mathbf{q}, \mathbf{K}^{l,h}, \mathbf{V}^{l,h})$ ▷ *full-cache weight and output*
      $\Delta\mathbf{o}_{q,i}^{l,h} \leftarrow \big[a_{q,i}^{l,h} / (1 - a_{q,i}^{l,h} + \epsilon)\big] \big(\mathbf{v}_i^{l,h} - \mathbf{o}_q^{l,h}\big)$ ▷ *output shift if i is evicted*
      $e_{q,i}^{l,h} \leftarrow \big\| \mathbf{W}_{l,h}^{O}\, \Delta\mathbf{o}_{q,i}^{l,h} \big\|_2^2$ ▷ *that shift, in the residual stream*
    end for
    $S_i^{l,h} \leftarrow \psi(\mathcal{E}_{i,\mathcal{W}}^{l,h}) + \lambda_L \sum_{m=1}^{M} \omega_m\, \psi(\mathcal{E}_{i,m}^{l,h}) \quad \forall i \in \mathcal{C}$ ▷ *combine prompt and weighted lookahead importance*
  end for
  $E_{l,h} \leftarrow \sum_{i \in \mathcal{C}} S_i^{l,h}$; $\{B_{l,h}\}_{h=1}^{H} \leftarrow \mathrm{Alloc}\big(B_l, \{E_{l,h}\}_{h=1}^{H}; \tau\big)$ ▷ *budget split across heads*
  for KV head $h = 1, \dots, H$ **do**
    $\mathcal{R}_{\mathrm{final}}^{l,h} \leftarrow \mathcal{P} \cup \mathrm{Select}\big(\mathbf{S}^{l,h}, B_{l,h}\big)$ ▷ *keep the top-scoring candidates*
    $(\mathbf{K}_\star^{l,h}, \mathbf{V}_\star^{l,h}) \leftarrow \mathrm{Gather}\big(\mathbf{K}^{l,h}, \mathbf{V}^{l,h}, \mathcal{R}_{\mathrm{final}}^{l,h}\big)$ ▷ *build the compressed cache*
  end for
end for
discard all lookahead states $\{\mathcal{Q}_{\mathrm{look}}^{(m),l,h}\}_{m=1}^{M}$
return $(\mathbf{K}^\star, \mathbf{V}^\star)$ ▷ *decoding uses the compressed cache only*

For a fixed-length trajectory of length T,

$$\mathrm{Var}(\bar{c}_m) = \frac{1}{T^2} \sum_{u=1}^{T} \mathbb{E}[V_u], \qquad V_u = \mathrm{Var}_{y \sim q(\cdot \mid h_u)}[\log q(y \mid h_u)], \tag{9}$$

where V_u is the varentropy. For a trajectory that terminates after $T_m < T$ steps, T is replaced by T_m in the corresponding per-trajectory averages. The optional normalization

$$\tilde{c}_m = \frac{\bar{c}_m}{\sqrt{\bar{V}_m / T_m}}, \qquad \bar{V}_m = \frac{1}{T_m} \sum_{u=1}^{T_m} V_u^{(m)},$$

normalizes $\bar{c}_m$ by its trajectory-level standard-error scale; it is evaluated only as an ablation and is not part of the default method.

Query-space disagreement. Predictive entropy is defined in token-probability space, whereas eviction ultimately depends on query directions. We therefore measure whether trajectory m follows an atypical query path relative to the other sampled futures. Let H_q denote the number of query heads

and $T_{mr} = \min(T_m, T_r)$ the common length of trajectories m and r. We define

$$D_m = \frac{1}{M-1} \sum_{r \neq m} \frac{1}{T_{mr} L H_q} \sum_{u=1}^{T_{mr}} \sum_{l=1}^{L} \sum_{g=1}^{H_q} \left[1 - \cos\left(\mathbf{q}_u^{(m),l,g}, \mathbf{q}_u^{(r),l,g} \right) \right]. \tag{10}$$

Queries compared at step u occupy the same absolute position $n+u$ and therefore share the same rotary phase. We standardize D_m across particles as

$$\hat{D}_m = \begin{cases} (D_m - \bar{D})/\operatorname{sd}(D), & M \geq 3 \text{ and } \operatorname{sd}(D) > 0, \\ 0, & \text{otherwise}, \end{cases} \qquad \bar{D} = \frac{1}{M} \sum_{m=1}^{M} D_m.$$

This definition covers both the $M \leq 2$ degeneracy and the rare case in which all particles have identical disagreement scores. The computation reuses the stored lookahead queries and therefore requires no additional decoding. Since $\bar{H}_m$ is measured in nats whereas $\hat{D}_m$ is standardized, their coefficients are calibrated on their realized across-particle scales rather than compared by raw magnitude.

Estimator interpretation. No separate softmax temperature is needed: multiplying Eq. 3 by a concentration parameter η is equivalent to rescaling $(\beta, \gamma, \lambda_D)$. With uniform weights, the lookahead term is the ordinary Monte Carlo estimate of the expected deletion effect under $p_\theta(Y \mid X)$. When $\lambda_D = 0$, the normalized weighted estimate is a self-normalized importance estimator whose large-M target is

$$\pi(Y) \propto p_\theta(Y \mid X) \exp\left(\beta \bar{c}(Y) - \gamma \bar{H}(Y) \right).$$

For any trajectory statistic f, a first-order expansion around $(\beta, \gamma) = (0, 0)$ gives

$$\mathbb{E}_\pi[f] = \mathbb{E}_{p_\theta}[f] + \beta \operatorname{Cov}_{p_\theta}(f, \bar{c}) - \gamma \operatorname{Cov}_{p_\theta}(f, \bar{H}) + O\left(\|(\beta, \gamma)\|^2 \right).$$

Centering therefore makes the typicality channel zero-mean under the sampling distribution, but does not make the finite-M weighted estimator unbiased. When $\lambda_D > 0$, $\hat{D}_m$ depends on the entire particle ensemble, so the resulting rule is an interacting-particle estimator rather than finite-M importance weighting for a fixed single-trajectory target.

Failure modes and safeguards. Reliability weighting is introduced as a robustness safeguard because future-query estimation introduces an additional source of uncertainty: sampled continuations can be plausible yet unrepresentative of the queries encountered during actual decoding. Existing KV-cache methods typically rely on prompt-side or single-surrogate query references without explicitly assessing their reliability. LORE-KV addresses this gap by aggregating multiple plausible futures and softly reweighting them using complementary entropy and query-space disagreement signals, while λ_L limits the influence of imperfect lookahead. This reliability-aware formulation makes future-aware eviction robust to sampling variability rather than assuming that every predicted future is equally trustworthy. Appendix F.4 quantifies how these reliability signals relate to oracle particle utility, and separates that effect from the coverage gain of Figure 1.

C.4 Computational Overhead

LORE-KV restricts scoring to at most $w + MT \ll n$ prompt-window and lookahead queries, keeping the additional computation localized. The entropy statistics for LORE-KV /E are obtained directly from the sampling logits, while LORE-KV /D adds only pairwise cosine disagreement over already-materialized query states, requiring $O(M^2 L T H_q d)$ arithmetic and no additional model decoding. Table 3 shows that LORE-KV /E incurs approximately $1.46\times$–$2.77\times$ the per-sample wall-clock time of AnDPro across the evaluated settings; LORE-KV /D has the same lookahead decoding cost with only the additional query-space reduction. Since (M, T) is fixed, this one-time overhead is independent of the retained cache budget B, explaining its non-monotonic variation with B. Importantly, the robustness of the resulting gains is examined through matched lookahead-token budget controls and analyses across cache budgets, context lengths, and prompt-to-future mismatch in Sections 4.1-4.5, while the single-surrogate score-level control is reported in Appendix F.3. Figure 7 separates LORE-KV's one-time lookahead/scoring overhead from the shared decoding cost and shows that this extra compute consistently translates into higher quality at matched cache budgets.

Table 3: Average inference time per sample (seconds). Lower is better. LORE-KV uses fixed lookahead compute, so non-monotonic latency across budgets, including at B=1024, is consistent with runtime variation.

Backbone	Dataset	Method	128	256	512	1024
Mistral	LongBench	AnDPro	7.8799	7.4885	7.4355	7.5338
		CriticalKV	**6.6867**	**6.3991**	**6.3982**	**6.5276**
		LORE-KV /E	17.5342	19.5496	19.6767	17.7930
	RULER	AnDPro	6.2360	4.5775	5.9707	3.1746
		CriticalKV	**6.1907**	**4.1849**	**4.9899**	**2.4039**
		LORE-KV /E	13.6877	11.1720	14.5537	7.2315
	NIAH	AnDPro	2.9842	2.8559	2.8779	2.9169
		CriticalKV	**2.3514**	**2.1441**	**2.1109**	**2.1961**
		LORE-KV /E	7.3873	7.4716	7.4072	7.3407
Llama	LongBench	AnDPro	8.3345	8.2447	8.1550	8.0648
		CriticalKV	**7.0724**	**7.0455**	**7.0165**	**6.9867**
		LORE-KV /E	12.1815	14.4038	15.8882	13.1342
	RULER	AnDPro	4.3311	4.0226	3.7141	3.4060
		CriticalKV	**4.2994**	**3.6776**	**3.1042**	**2.5791**
		LORE-KV /E	6.6410	6.8587	6.3249	5.4202
	NIAH	AnDPro	2.6046	2.5137	2.5269	2.6172
		CriticalKV	**1.9267**	**1.9222**	**1.9450**	**1.8561**
		LORE-KV /E	7.0298	6.9553	7.0076	6.9816

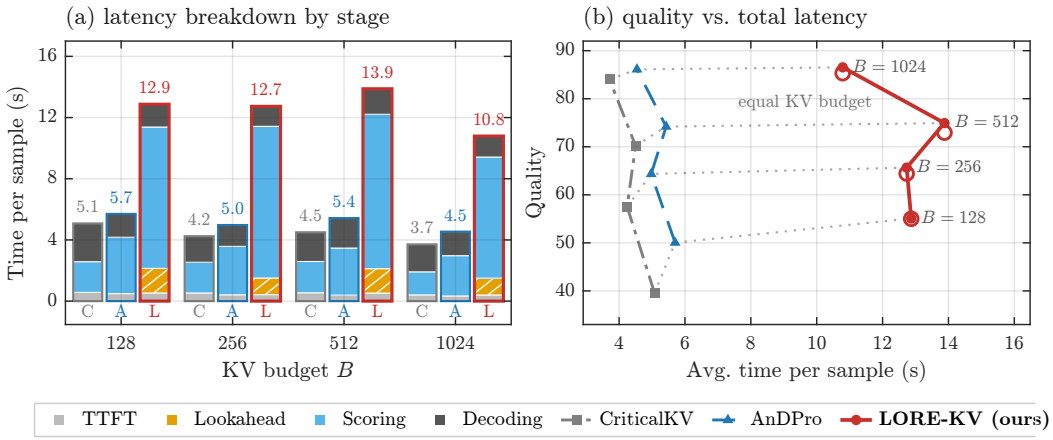


Figure 7: **Latency-quality trade-off.** (a) Stage-wise latency averaged over two seeds and (b) quality versus total latency across cache budgets on Mistral-7B-Instruct-v0.2, averaged over LongBench, RULER, and NIAH.

D Plug-In Components and the Ablation Protocol

We evaluate three optional plug-ins that modify different stages of LORE-KV: (A) score aggregation, (B) head-budget allocation, and (C) token selection. Results are reported in Tables 14 and 15. All three are disabled in the core method, so the main results use Eq. 7 with uniform head budgets and plain top-$B_{l,h}$ selection. This separation lets us isolate whether gains come from better future-query scoring or from additional allocation and selection mechanisms.

Where each plug-in acts. The three components modify different parts of the pipeline. *(A) Robust aggregation* changes the per-token aggregation function ψ in Eq. 7, but each token still receives one scalar score and the top-$B_{l,h}$ rule is unchanged. *(B) Adaptive head budgets* keep the scores and

Table 4: LORE-KV hyperparameters and default settings.

Symbol	Meaning	Default
(a) Core method		
w	prompt-end observation-window size	32
M	number of lookahead particles	4
T	lookahead particle length	8
λ_L	lookahead mixing coefficient in Eq. 7	1.0
ϵ	renormalization guard in Eq. 5	10^{-6}
	chunk size (adjacent-token grouping)	4
(b) Plug-in switches (Appendix D)		
κ, α	robust aggregation	0, 0
τ	adaptive head budgets	0 (ablation: 1)
λ_k, λ_v	diversity selection	0 (ablation: 0.1)
(c) Reliability settings in Eq. 3		
LORE-KV /E	log-probability + entropy penalty	(1, 1.3, 0)
LORE-KV /D	log-probability + disagreement penalty	(1, 1, 0.3)
Log-probability only	control	(1, 1, 0)
Entropy + disagreement	control	(1, 1.3, 0.3)

Table 5: LongBench English datasets, the abbreviations used in all LongBench results tables, metrics, average input lengths, and sample counts (Bai et al., 2024).

Task	Abbr.	Task type	Metric	Avg. len	Language	#Samples
NarrativeQA	NQA	Single-Doc. QA	F1	18,409	EN	200
Qasper	Qasp	Single-Doc. QA	F1	3,619	EN	200
MultiFieldQA-en	MF	Single-Doc. QA	F1	4,559	EN	150
HotpotQA	HQA	Multi-Doc. QA	F1	9,151	EN	200
2WikiMultihopQA	2WQA	Multi-Doc. QA	F1	4,887	EN	200
MuSiQue	Musq	Multi-Doc. QA	F1	11,214	EN	200
GovReport	GovR	Summarization	Rouge-L	8,734	EN	200
QMSum	QMS	Summarization	Rouge-L	10,614	EN	200
MultiNews	MN	Summarization	Rouge-L	2,113	EN	200
TREC	TREC	Few-shot Learning	Accuracy	5,177	EN	200
TriviaQA	TQA	Few-shot Learning	F1	8,209	EN	200
SAMSum	SAMS	Few-shot Learning	Rouge-L	6,258	EN	200
PassageCount	PC	Synthetic	Accuracy	11,141	EN	200
PassageRetrieval-en	PR	Synthetic	Accuracy	9,289	EN	200
LCC	LCC	Code	Edit Sim	1,235	Python/C#/Java	500
RepoBench-P	RB-P	Code	Edit Sim	4,206	Python/Java	500

ranking fixed but change how the layer budget is distributed across heads through Alloc in Eq. 8. *(C) Diversity-aware selection* replaces Select with a set-dependent greedy rule, since a token's effective utility depends on which tokens have already been retained. Thus, A changes the score statistic, B changes the cutoff, and C changes the selection rule. The core configuration is recovered with $\alpha = 0$, $\tau = 0$, and $\lambda_k = \lambda_v = 0$.

A. Robust aggregation (α, κ). A mean can underweight tokens that are important for only a small subset of future queries. Motivated by DefensiveKV (Feng et al., 2025a), we therefore define

$$\mathrm{TopAvg}_\kappa(\mathcal{E}) = \frac{1}{K_\kappa} \sum_{e \in \mathrm{TopK}(\mathcal{E}, K_\kappa)} e, \qquad K_\kappa = \lceil \kappa |\mathcal{E}| \rceil,$$

and use

$$\psi(\mathcal{E}) = \alpha\, \mathrm{TopAvg}_\kappa(\mathcal{E}) + (1-\alpha)\, \mathrm{Mean}(\mathcal{E}), \qquad \alpha \in [0,1].$$

Setting $\alpha = 0$ exactly recovers the core mean aggregation, while larger α gives more weight to high-importance query subsets and reduces sensitivity to an unrepresentative particle. Unless stated

Table 6: LongBench results on Llama-3.1-8B-Instruct. Dataset abbreviations follow Table 5.

	Single-Doc QA			Multi-Doc QA			Summ.			Few-shot			Synth.		Code		
Method	NQA	Qasp	MF	HQA	2WQA	Musq	GovR	QMS	MN	TREC	TQA	SAMS	PC	PR	LCC	RB-P	Avg.
Full Cache	30.12	46.60	56.41	58.10	49.01	32.52	34.04	25.37	27.06	73.00	91.90	43.46	7.06	100.00	62.13	51.80	49.29
Budget $B = 128$																	
SnapKV	25.54	31.40	50.99	55.17	45.50	27.88	20.77	23.41	21.05	47.50	90.46	40.76	8.00	99.50	56.88	46.50	43.21
PyramidKV	27.83	32.98	51.45	56.72	44.67	30.54	21.89	22.90	21.67	63.00	90.45	40.06	8.00	**100.00**	55.37	43.79	44.46
Ada-KV	25.06	34.50	52.27	56.29	47.57	29.06	21.17	23.36	21.51	62.00	**92.17**	40.84	8.00	99.50	58.45	48.72	45.03
CriticalKV	26.00	34.96	51.79	56.01	46.53	28.95	21.31	23.14	21.33	62.00	89.51	40.74	7.59	97.89	57.85	48.28	44.62
AnDPro	25.25	35.85	52.85	54.35	44.41	27.51	22.82	23.60	22.20	69.00	92.00	41.50	7.02	**100.00**	**62.26**	**56.34**	46.06
LaProx	25.78	31.38	**52.97**	55.17	44.95	29.46	21.66	23.66	21.47	59.00	89.94	41.09	**8.89**	99.50	61.91	56.16	45.19
LAQ++(8)	**28.65**	10.65	26.04	24.23	21.56	15.67	23.50	23.74	22.75	**72.00**	91.95	**42.19**	8.37	94.81	62.18	50.95	38.70
LORE-KV /E	27.38	**37.66**	51.14	57.37	48.47	30.55	**23.77**	23.60	23.14	68.00	90.46	41.77	7.34	97.90	58.99	50.15	46.11
LORE-KV /D	27.33	37.50	51.81	**57.73**	**48.72**	**30.63**	23.12	**24.25**	**23.62**	69.50	90.11	41.69	7.98	97.02	59.09	50.17	**46.27**
Budget $B = 256$																	
SnapKV	26.92	38.19	52.62	57.03	46.75	29.63	23.07	23.92	22.82	58.50	91.83	40.81	7.75	99.50	59.95	48.63	45.50
PyramidKV	28.46	39.36	54.30	56.77	45.50	30.74	23.84	23.88	22.91	69.00	91.09	40.74	7.88	99.50	56.94	45.18	46.01
Ada-KV	27.04	39.24	54.41	57.36	47.53	31.35	23.06	23.81	23.43	69.00	**92.50**	41.23	7.67	**100.00**	61.45	49.52	46.79
CriticalKV	25.45	39.60	52.90	55.72	46.05	29.97	23.10	23.77	22.77	68.00	90.48	41.08	7.48	97.55	60.19	48.25	45.77
AnDPro	27.23	42.01	**54.62**	**57.87**	47.43	31.92	23.53	24.22	23.54	71.50	92.37	41.47	7.66	99.81	62.45	51.67	47.46
LaProx	28.02	38.56	53.32	55.23	45.72	30.83	24.84	23.78	23.92	67.50	92.12	42.40	**8.56**	99.50	63.57	**57.61**	47.22
LAQ++(8)	**30.25**	12.43	26.63	25.77	22.83	18.45	25.07	23.67	23.75	**72.00**	91.97	**43.17**	6.93	94.59	**64.07**	53.38	39.69
LORE-KV /E	28.26	**44.01**	53.43	57.73	48.78	32.30	**25.63**	**24.93**	24.10	71.00	91.03	39.98	7.39	98.52	61.70	51.28	**47.50**
LORE-KV /D	28.33	42.69	52.66	57.75	**48.86**	**32.34**	24.95	24.87	**24.16**	70.00	91.09	40.95	7.77	99.73	61.61	51.84	47.48
Budget $B = 512$																	
SnapKV	28.40	41.52	55.40	57.66	47.48	30.95	24.92	24.46	24.54	68.50	92.33	41.98	7.33	99.50	61.25	50.39	47.29
PyramidKV	29.15	43.04	55.55	57.71	47.67	31.27	26.05	24.08	24.54	70.50	**92.67**	41.52	7.33	**100.00**	59.96	47.51	47.41
Ada-KV	26.92	42.68	55.20	57.47	48.06	31.27	24.95	24.17	24.40	71.00	92.14	42.35	7.33	**100.00**	62.26	50.62	47.55
CriticalKV	26.00	42.53	53.90	56.62	46.76	31.13	24.26	23.76	23.81	70.00	89.56	40.75	6.92	97.20	60.51	49.74	46.47
AnDPro	27.97	**44.15**	55.49	58.16	48.19	31.92	24.80	24.50	24.77	**72.50**	91.76	41.99	7.31	99.78	62.54	52.00	47.99
LaProx	29.82	43.46	54.33	55.44	46.16	30.83	27.08	24.10	**25.30**	70.50	91.89	42.48	8.40	99.50	64.10	**58.16**	48.23
LAQ++(8)	29.64	13.22	26.79	27.58	23.49	18.63	26.94	24.04	25.21	**72.50**	92.25	**42.83**	**8.43**	96.25	**65.00**	53.37	40.39
LORE-KV /E	29.77	43.61	55.22	**58.32**	48.68	**32.73**	27.40	24.96	25.02	**72.50**	91.50	40.50	7.46	99.52	60.38	51.25	48.05
LORE-KV /D	**30.91**	43.68	**55.88**	57.58	**49.38**	32.71	**28.22**	**25.34**	24.94	**72.50**	91.22	40.87	7.18	99.46	61.02	51.29	**48.26**
Budget $B = 1024$																	
SnapKV	28.32	45.37	**56.81**	58.16	48.28	31.89	26.91	24.58	25.82	70.50	92.22	43.08	6.96	**100.00**	62.46	51.50	48.30
PyramidKV	29.32	44.57	56.14	58.39	48.36	31.93	28.20	23.78	26.00	70.50	**92.41**	42.46	6.88	**100.00**	61.33	49.12	48.09
Ada-KV	28.52	**46.01**	55.72	57.85	48.10	32.02	26.98	24.72	25.96	72.00	91.97	42.55	6.96	**100.00**	61.98	51.52	48.30
CriticalKV	27.27	44.74	54.24	56.17	46.95	31.65	26.35	23.91	24.93	70.00	89.08	41.00	6.70	96.86	59.85	49.75	46.84
AnDPro	**31.36**	43.42	55.10	55.05	45.93	29.79	28.46	24.70	26.56	72.50	91.42	**43.57**	7.39	99.50	64.86	**59.01**	48.66
LaProx	31.08	44.45	55.64	55.82	45.98	31.96	**29.22**	24.42	26.36	71.50	91.75	43.32	**8.23**	99.50	**65.01**	58.69	**48.94**
LORE-KV /E	29.99	44.83	56.19	58.60	**49.01**	**33.02**	28.90	**25.49**	26.38	73.00	92.35	42.24	7.50	99.50	61.31	51.60	48.74
LORE-KV /D	31.19	44.92	56.02	**58.96**	48.91	32.79	28.99	25.42	**26.65**	**73.50**	92.23	42.40	7.20	98.75	60.55	52.03	48.78

otherwise, we use $\alpha = 0.5$ with κ from Table 4, giving an equal mixture of the top-κ statistic and the mean. Only the scalar token score changes; ranking and top-$B_{l,h}$ selection remain unchanged.

B. Adaptive head-wise budgets (τ). The core method divides each layer budget uniformly across KV heads. Following Ada-KV (Feng et al., 2025b) and the layer-wise allocation principle of PyramidKV (Cai et al., 2025b), we instead allow heads with larger future-query perturbation mass

$$E_{l,h} = \sum_{i \in \mathcal{C}} S_i^{l,h}$$

to receive more capacity:

$$B_{l,h} = \left\lfloor B_l \frac{(E_{l,h} + \epsilon)^\tau}{\sum_{h'=1}^{H} (E_{l,h'} + \epsilon)^\tau} \right\rfloor . \tag{11}$$

Here $\tau \geq 0$ controls how strongly the budget is shifted toward high-impact heads; $\tau = 0$ recovers the uniform allocation $\lfloor B_l / H \rfloor$. The within-head scores and their ordering are unchanged; only the retention cutoff changes. Figure 12 motivates this ablation by showing that both $E_{l,h}$ and the benefit of future-query scoring vary substantially across layers and heads.

C. Diversity-aware selection (λ_k, λ_v). Plain top-$B_{l,h}$ selection may retain several near-duplicate keys or values. As an ablation, we use a greedy redundancy-aware rule inspired by GraphKV (Li

Table 7: RULER per-task results with Mistral-7B-Instruct-v0.2. CWE/FWE denote common-/frequent-word extraction; MK/MV/MQ/S denote multi-key/multi-value/multi-query/single-needle variants; VT denotes variable tracking.

B	Method	CWE	FWE	MK1	MK2	MK3	MQ	MV	S1	S2	S3	QA-1	QA-2	VT	Avg.
Context length = 4K															
128	CriticalKV	9.40	40.20	89.20	70.40	0.00	0.80	17.40	89.20	96.00	0.00	82.20	59.40	15.60	43.83
128	AnDPro	17.20	44.20	98.60	87.60	**0.40**	**37.40**	**46.40**	99.60	**99.80**	0.00	82.60	61.20	38.00	54.85
128	**LORE-KV /E**	**29.62**	**61.27**	**99.80**	99.60	0.00	19.40	36.50	**100.00**	**99.80**	**4.00**	84.00	61.60	**63.56**	**58.40**
128	**LORE-KV /D**	29.18	60.80	**99.80**	**99.80**	0.00	19.00	36.30	**100.00**	**99.80**	3.80	**84.20**	**62.00**	63.40	58.31
256	CriticalKV	24.20	70.80	93.20	85.00	17.20	70.00	60.20	93.20	97.40	9.00	83.20	60.80	66.40	63.89
256	AnDPro	29.20	73.40	99.40	96.20	**17.40**	**93.80**	**79.00**	**100.00**	**99.80**	9.00	83.40	62.00	81.00	71.05
256	**LORE-KV /E**	35.20	82.40	98.60	98.60	15.80	81.60	71.80	98.60	98.40	10.00	82.80	60.80	**95.00**	**71.51**
256	**LORE-KV /D**	**41.52**	**84.07**	**99.60**	**100.00**	4.80	49.15	59.25	**100.00**	**99.80**	**32.80**	**84.20**	**62.40**	89.44	69.77
512	CriticalKV	43.60	84.00	95.80	92.80	69.20	83.60	78.60	96.00	98.40	54.80	84.20	61.60	87.80	79.26
512	AnDPro	46.40	85.40	**99.40**	99.40	**69.40**	**97.40**	**89.60**	**100.00**	**99.80**	54.80	84.40	**62.20**	96.20	83.42
512	**LORE-KV /E**	50.80	**91.40**	**99.40**	99.60	68.80	91.00	86.00	99.60	99.40	55.80	84.40	62.00	**99.80**	**83.69**
512	**LORE-KV /D**	**60.22**	89.53	**99.40**	**100.00**	35.40	81.45	78.65	**100.00**	**99.80**	**72.40**	**85.40**	61.80	97.72	81.67
1024	CriticalKV	76.40	89.00	97.60	96.60	**89.20**	90.20	84.60	98.00	99.00	93.80	**85.00**	61.20	93.60	88.78
1024	AnDPro	78.00	89.80	**99.40**	**100.00**	**89.20**	**97.40**	**90.40**	**100.00**	**99.80**	93.80	**85.00**	**61.60**	98.20	90.97
1024	**LORE-KV /E**	80.60	**93.20**	**99.40**	99.80	89.00	93.80	88.40	99.80	99.60	**94.40**	**85.00**	61.40	**100.00**	**91.11**
1024	**LORE-KV /D**	**87.72**	91.53	99.20	**100.00**	73.20	94.00	86.70	**100.00**	**99.80**	91.60	84.80	61.00	98.88	89.88
Context length = 8K															
128	CriticalKV	3.20	26.40	75.20	59.60	0.00	0.00	6.00	92.60	88.40	0.00	75.40	58.80	18.20	38.75
128	AnDPro	7.40	29.60	94.60	82.80	**0.40**	**19.60**	35.40	**100.00**	97.20	0.00	74.40	59.40	50.00	50.06
128	**LORE-KV /E**	**19.62**	**57.73**	96.40	98.40	0.00	12.90	35.20	**100.00**	**100.00**	**1.20**	**78.20**	61.40	**63.92**	**55.77**
128	**LORE-KV /D**	19.16	56.47	**96.80**	**98.60**	0.00	13.10	**35.95**	**100.00**	**100.00**	0.80	77.80	**62.00**	**63.92**	55.74
256	CriticalKV	11.40	46.40	83.80	77.20	4.00	68.80	52.20	95.20	94.00	2.60	**77.20**	59.80	65.00	56.74
256	AnDPro	14.20	48.40	96.40	92.20	**4.20**	**81.60**	**71.20**	**100.00**	99.80	2.60	76.60	60.20	85.60	64.08
256	**LORE-KV /E**	19.80	63.80	95.60	**98.20**	2.20	75.80	69.40	98.20	98.20	1.60	77.00	59.60	**92.40**	**65.52**
256	**LORE-KV /D**	**29.12**	**69.60**	**96.80**	**98.20**	1.40	34.15	55.60	**100.00**	**100.00**	**25.00**	77.00	**61.40**	91.00	64.56
512	CriticalKV	19.80	67.00	88.80	88.60	26.60	86.60	72.40	97.20	96.60	19.60	**78.20**	57.80	81.60	67.75
512	AnDPro	21.40	68.20	96.20	97.40	**26.80**	**94.00**	**83.60**	**100.00**	**100.00**	19.60	77.80	58.00	93.80	72.06
512	**LORE-KV /E**	24.80	**77.20**	95.80	**99.00**	25.60	90.80	82.60	99.00	99.00	19.00	**78.20**	57.80	**97.80**	**72.82**
512	**LORE-KV /D**	**37.22**	73.07	**96.40**	98.60	9.40	65.15	72.25	**100.00**	**100.00**	**54.80**	76.60	**60.40**	94.48	72.18
1024	CriticalKV	64.60	80.40	94.00	94.40	**68.00**	91.80	81.60	98.60	98.20	**76.20**	**81.80**	59.40	90.60	83.05
1024	AnDPro	65.40	81.20	**97.80**	99.00	**68.00**	**95.80**	**87.40**	**100.00**	**100.00**	**76.20**	81.60	**59.60**	97.00	85.31
1024	**LORE-KV /E**	67.60	**86.40**	97.60	99.40	67.40	94.00	87.00	99.40	99.40	75.80	**81.80**	59.40	**99.40**	**85.74**
1024	**LORE-KV /D**	**74.72**	84.73	97.40	**99.60**	51.60	94.20	85.30	99.60	99.60	73.00	81.60	59.00	98.28	84.51
Context length = 16K															
128	CriticalKV	0.00	44.20	62.00	50.20	**0.00**	**0.00**	0.00	92.60	83.80	0.00	73.00	51.40	11.80	36.08
128	AnDPro	0.00	46.40	82.00	73.80	**0.00**	**0.00**	**24.00**	**100.00**	92.80	0.00	72.20	52.40	44.00	45.20
128	**LORE-KV /E**	**12.00**	**71.60**	83.60	88.40	**0.00**	**0.00**	21.20	**100.00**	**94.80**	**1.80**	**75.40**	54.00	**60.80**	**51.05**
128	**LORE-KV /D**	11.54	70.73	**83.80**	**88.60**	**0.00**	**0.00**	21.45	**100.00**	**94.80**	1.40	75.20	**54.40**	60.72	50.97
256	CriticalKV	5.60	63.80	70.80	67.80	**1.40**	**49.60**	44.20	95.20	89.60	1.00	75.00	52.60	58.80	51.95
256	AnDPro	5.60	65.40	83.80	83.20	**1.40**	**49.60**	**60.00**	**100.00**	95.40	1.00	74.40	53.20	79.80	57.91
256	**LORE-KV /E**	11.80	79.40	83.80	91.00	0.20	42.60	57.20	99.00	95.60	1.20	75.40	53.20	**88.80**	**59.94**
256	**LORE-KV /D**	**19.62**	**83.13**	**84.80**	**91.60**	0.00	5.55	44.00	**100.00**	**97.20**	**24.20**	**76.00**	**54.80**	85.32	58.94
512	CriticalKV	14.60	80.80	78.60	81.20	**24.60**	**68.40**	65.40	97.20	93.00	18.40	**76.40**	52.00	76.80	63.65
512	AnDPro	14.60	81.80	86.20	90.20	**24.60**	**68.40**	**74.60**	**100.00**	96.40	18.40	76.00	52.40	89.00	67.12
512	**LORE-KV /E**	18.40	**90.20**	86.00	**94.60**	23.60	64.60	73.20	99.20	96.40	18.20	**76.40**	52.20	**94.60**	**68.28**
512	**LORE-KV /D**	**22.04**	83.93	**89.00**	88.00	2.80	46.60	60.70	**100.00**	**99.40**	**34.20**	76.00	**53.00**	89.04	64.98
1024	CriticalKV	61.00	89.40	87.20	89.60	**66.60**	**79.00**	76.80	98.60	95.80	**75.40**	**80.60**	55.80	87.40	80.25
1024	AnDPro	61.00	89.80	**91.20**	94.40	**66.60**	**79.00**	**81.60**	**100.00**	97.60	**75.40**	80.40	**56.00**	93.80	82.06
1024	**LORE-KV /E**	63.20	**94.60**	**91.20**	97.00	66.20	76.80	80.60	99.60	97.60	**75.40**	**80.60**	**56.00**	**97.00**	**82.75**
1024	**LORE-KV /D**	**70.32**	92.93	91.00	**97.20**	50.40	77.00	78.90	99.80	**97.80**	72.60	80.40	55.60	95.88	81.53

et al., 2025b). Starting from $\mathcal{R}^{l,h} = \emptyset$, we repeatedly select

$$i^\star = \arg\max_{i \in \mathcal{C} \setminus \mathcal{R}^{l,h}} \left[S_i^{l,h} - \lambda_k \max_{j \in \mathcal{R}^{l,h}} \cos(\mathbf{k}_i^{l,h}, \mathbf{k}_j^{l,h}) - \lambda_v \max_{j \in \mathcal{R}^{l,h}} \cos(\mathbf{v}_i^{l,h}, \mathbf{v}_j^{l,h}) \right]$$

until $|\mathcal{R}^{l,h}| = B_{l,h}$. Setting $\lambda_k = \lambda_v = 0$ recovers plain top-$B_{l,h}$ selection.

Unlike robust aggregation, this rule is set-dependent because a token's effective utility changes with the already-selected set. A possible failure mode is suppressing repeated but genuinely useful evidence, such as multiple passages mentioning the same entity. We limit this risk in three ways: the

redundancy penalty is applied on top of the future-query perturbation score, λ_k and λ_v are kept small so that redundancy mainly breaks ties among similarly important tokens, and the perturbation score itself already discounts some redundancy because deleting a token that is well compensated by a near-copy produces a small $\Delta\mathbf{o}_{q,i}^{l,h}$.

Why the core method disables all three plug-ins. We set $\alpha = \tau = \lambda_k = \lambda_v = 0$ for every headline result so that improvements can be attributed directly to the query reference and deletion score rather than to a tuned combination of auxiliary mechanisms. The plug-in rows in Tables 14 and 15 therefore act as robustness checks: each modifies one pipeline stage using an idea from prior work, while the overall ranking against the baselines is preserved and the changes remain small relative to the core gain.

Ablations of the core method. We separately ablate three parts of the core formulation.

(i) Lookahead geometry. We vary $M \in \{0, 1, 2, 4, 8\}$ at the default horizon $T = 8$ in Table 14 and evaluate the full $M \times T$ grid, with $T \in \{2, 4, 8, 16, 32\}$, in Table 13. The $M = 0$ condition uses only the prompt window and is therefore independent of T. All configurations are evaluated on LongBench at $B = 128$ with uniform head budgets and top-K selection. Since lookahead sampling is stochastic, we additionally repeat the default configuration ($M = 4$, $T = 8$, $B = 128$) with three independent sampling seeds. Across these runs, LORE-KV /E achieves a mean LongBench score of 40.28 (95% CI $[39.97, 40.59]$), compared with 38.96 for AnDPro and 35.87 without lookahead ($M = 0$). The consistent performance across seeds indicates that this improvement is not attributable to a particular sampled set of lookahead trajectories. For the matched lookahead-token budget comparison in Table 13b, AnDPro's criterion scores the same prompt window and M trajectories of length T, aggregated uniformly without reliability weighting, under the same budgets and selector. We measure compute as the number of lookahead tokens $M \times T$, which sets both the lookahead decoding FLOPs and the number of extra scoring queries ($w + MT$). The comparison is conservative at the cell level.

(ii) Reliability weighting. We vary the $(\beta, \gamma, \lambda_D)$ channels of Eq. 3 while keeping all other components fixed. The reported LORE-KV /E and LORE-KV /D settings are described in Appendix G; log-probability-only and entropy-plus-disagreement variants serve as controls. All configurations use uniform head budgets and top-K selection and re-weight nested prefixes of one 8-particle draw per prompt, so they have matched lookahead decoding compute.

(iii) Perturbation objective. Under the same lookahead queries, we replace Eq. 6 with accumulated attention mass, the unprojected perturbation norm $\|\Delta\mathbf{o}_{q,i}^{l,h}\|_2^2$, or AnDPro's scoring criterion. These matched lookahead-token budget controls are reported in Table 14 and discussed in Section 4.1. At $B = 128$, changing the deletion objective changes the LongBench average by at most 0.09 points, indicating that the query reference has a substantially larger effect than the exact deletion criterion.

Overall interpretation. The ablations consistently localize the main source of improvement. Changing aggregation, head-budget allocation, selection, or even the deletion objective under fixed lookahead queries produces only small changes. In contrast, replacing prompt-side queries with response-side lookahead queries produces the dominant gain. Thus, LORE-KV's performance is driven primarily by correcting the query-reference mismatch rather than by a tuned combination of aggregation, allocation, or selection heuristics.

E Experimental Setup and Additional Results

E.1 Protocol, Datasets and Baselines

Table 5 gives the LongBench task metadata used in these tests, together with the dataset abbreviations used in every LongBench results table.

Baselines and sourcing. LongBench results with Mistral follow the protocol of Geng et al. (2025). CriticalKV, AnDPro, and both LORE-KV variants are re-run under one implementation for RULER, NIAH, and LongBench with Llama. The LAQ++(8) values are reproduced from Wang et al. (2025b) and are shown only as an external reference because they were obtained under that work's evaluation protocol. LORE-KV samples M=4 trajectories from the same distribution in one batch for T=8 autoregressive steps, eight sequential steps but MT=32 decoded token positions on the full

cache, so we do not treat LAQ++ as budget-matched. LAQ++(8) values are available, and therefore reported, only at $B \in \{128, 256, 512\}$. GVote (Tang et al., 2025) is closely related through Monte Carlo-style future-query estimation, but its published formulation uses synthetic-query voting to set an adaptive per-request cache budget, whereas LORE-KV evaluates token utility under a fixed budget. Because we could not identify an official public implementation, we do not report or claim a protocol-matched advantage over GVote. Our controlled evidence instead relies on baselines re-run in a common harness and the AnDPro control with matched queries and lookahead-token budget in Table 13b.

Backbones and prompt formats. The four dense-attention backbones are Mistral-7B-Instruct-v0.2, Llama-3.1-8B-Instruct, Qwen2.5-14B-Instruct and Ministral-3-8B-Instruct-2512. Two further backbones, Qwen3.5-9B and Gemma-4-12B-it, use hybrid attention and are described in Appendix E.2; both are evaluated with their own chat templates. Prompts follow each model's own convention and are otherwise identical across methods, so the comparison within a backbone is unaffected: Mistral takes the raw LongBench prompt, matching Geng et al. (2025); Llama-3.1 and Qwen2.5 use their chat templates; and Ministral-3 uses Mistral's `[INST]` $\cdots$ `[/INST]` format rather than its packaged template, which would prepend a several-hundred-token assistant system prompt to every long-context input. Qwen2.5-14B-Instruct is also the one backbone of materially different scale, at roughly twice the parameters of the other three. Ministral-3-8B-Instruct-2512 is released as a vision-language checkpoint; we evaluate its text tower, extracted without modifying any weights.

Implementation. Our implementation follows the protocol of Geng et al. (2025) and uses a single NVIDIA H100 80 GB GPU. All methods use the same retained cache size at a fixed budget, and adjacent-token grouping follows SnapKV and Ada-KV. Unless stated otherwise, LORE-KV uses the core configuration in Table 4, with all plug-ins disabled to isolate the score in Eq. 7. The reliability coefficients are the only difference between the two variants.

Lookahead particles are sampled at temperature 1 without top-k or top-p truncation, so the sampling distribution is exactly p_θ and Eq. 2 remains correctly centered. We compute c_u, H_u, and V_u from the same logits used for sampling, with masked reductions to avoid undefined $0 \cdot (-\infty)$ terms under truncated support.

Per-step statistics require only MT floats per prompt. Query-space disagreement is computed across all decoder layers from the stored lookahead queries before cache compression releases them. If a particle ends early, D_m is computed only over the shared valid steps, while $\bar{c}_m$ and $\bar{H}_m$ are averaged independently over each particle's valid steps.

E.2 Hybrid-Attention Backbones: Where Eviction Applies

Dense versus hybrid backbones. The four dense backbones use full attention in every layer, so their KV caches grow linearly with context length, and LORE-KV, CriticalKV, and AnDPro prune every layer to the same budget. Qwen3.5-9B and Gemma-4-12B-it are *hybrid*: only 8/32 and 8/48 layers, respectively, use full attention, while the remaining layers maintain $O(1)$ state through linear-attention recurrence in Qwen3.5 or a bounded sliding window (W=1024) in Gemma-4-12B-it. These fixed-size layers do not require eviction.

Implication for LORE-KV. LORE-KV is therefore applied only to the full-attention subset, where KV memory grows with context. Although this involves fewer layers, they contain most of the cache memory at long context: at 64K, they account for 98.2% of Qwen3.5's KV memory and 75.8% of Gemma-4-12B-it's. Thus, hybridization changes the number of compressible layers, but not the core compression problem. Budget, observation window, and chunk size remain identical across methods, ensuring a matched comparison.

Hybrid-specific implementation. Qwen3.5 represents full and linear-attention layers as separate module classes, so eviction is naturally restricted to full-attention layers. Gemma-4-12B-it uses a shared attention class and requires explicit layer selection. We also preserve Gemma-4-12B-it's native K=V sharing on full-attention layers (`attention_k_eq_v`), so the reported budgets follow the model's original cache layout.

Lookahead cost. LORE-KV clones the live cache once per sampled particle for future-query rollouts. On hybrid models, this clone is also restricted to full-attention layers, reducing absolute cost while preserving the relative overhead over non-lookahead baselines. Because these layers still con-

Table 8: LongBench results on **Qwen3.5-9B**. Eviction is applied to the full-attention layers, which hold 98.2% of the cache at 64K context as discussed in Appendix E.2.

	Single-Doc QA			Multi-Doc QA			Summ.			Few-shot			Synth.		Code		
Method	NQA	Qasp	MF	HQA	2WQA	Musq	GovR	QMS	MN	TREC	TQA	SAMS	PC	PR	LCC	RB-P	Avg.
Full Cache	33.87	51.42	58.96	66.31	65.74	48.93	33.28	25.14	26.87	76.00	91.36	44.03	12.42	100.00	57.94	55.21	52.97
Budget $B = 128$																	
CriticalKV	31.92	41.19	52.98	61.28	60.21	46.08	22.53	20.81	18.11	62.50	87.60	30.94	10.00	98.00	49.81	46.83	46.30
AnDPro	31.86	42.28	54.21	60.61	60.60	43.45	22.39	21.12	18.34	68.50	88.25	32.01	9.50	99.50	50.38	47.24	46.89
LORE-KV /E	31.93	47.92	54.58	61.84	**61.51**	**46.24**	**24.30**	21.34	18.69	71.50	**89.06**	**34.22**	**11.50**	**100.00**	**53.13**	48.16	48.49
LORE-KV /D	**32.03**	**48.66**	**54.93**	**62.25**	61.09	46.06	24.09	**21.58**	**18.81**	**72.50**	**89.06**	33.72	11.00	**100.00**	52.67	**48.72**	**48.57**
Budget $B = 256$																	
CriticalKV	32.71	44.37	55.12	62.86	61.94	46.83	25.16	22.47	20.38	66.00	88.70	33.26	**12.38**	98.62	52.34	49.71	48.30
AnDPro	32.41	45.28	55.69	62.38	62.17	45.62	25.36	22.71	20.51	70.50	89.44	34.18	10.47	99.74	52.86	50.13	48.72
LORE-KV /E	32.87	48.94	56.43	63.66	**63.28**	**47.54**	26.84	22.96	**21.32**	73.00	90.12	**35.47**	11.86	**100.00**	**55.05**	51.38	50.05
LORE-KV /D	**33.16**	**49.53**	**56.87**	**64.28**	62.87	47.43	**27.31**	**23.11**	21.18	**73.50**	**90.47**	35.04	12.14	**100.00**	54.71	**51.86**	**50.22**
Budget $B = 512$																	
CriticalKV	32.46	46.72	56.31	64.17	63.42	47.24	27.86	23.14	22.36	69.50	88.92	34.62	11.16	99.24	54.28	51.94	49.58
AnDPro	32.86	47.63	56.18	63.82	63.74	46.18	27.91	23.37	22.30	72.50	89.51	**37.14**	10.92	99.86	54.83	52.47	50.08
LORE-KV /E	33.42	50.16	57.68	63.29	**64.86**	**47.28**	29.73	**23.84**	**23.41**	74.50	90.83	36.72	12.28	**100.00**	**56.68**	53.16	51.12
LORE-KV /D	**33.71**	**50.78**	**58.14**	**65.72**	62.51	47.11	**30.18**	22.86	22.76	**75.00**	**91.04**	36.28	**12.53**	**100.00**	56.23	**53.68**	**51.16**
Budget $B = 1024$																	
CriticalKV	33.19	48.16	56.72	64.83	63.86	**47.51**	29.64	23.47	**25.18**	71.00	88.55	35.16	11.47	99.06	54.86	52.38	50.31
AnDPro	33.06	49.04	55.94	64.51	64.12	46.74	27.54	23.68	23.73	73.50	89.12	35.94	11.83	99.62	55.41	52.96	50.42
LORE-KV /E	33.41	50.87	58.34	65.72	**65.31**	**47.51**	31.86	24.31	24.93	75.50	91.12	**37.16**	12.16	**100.00**	**56.42**	54.08	51.79
LORE-KV /D	**33.62**	**51.24**	**58.71**	**66.14**	64.81	47.36	**32.43**	**24.41**	24.62	**76.00**	**91.28**	36.83	**12.68**	**100.00**	56.01	**54.62**	**51.92**

Table 9: LongBench results on **Gemma-4-12B-it**. Eviction is applied to the full-attention layers, which hold 75.8% of the cache at 64K context as discussed in Appendix E.2.

	Single-Doc QA			Multi-Doc QA			Summ.			Few-shot			Synth.		Code		
Method	NQA	Qasp	MF	HQA	2WQA	Musq	GovR	QMS	MN	TREC	TQA	SAMS	PC	PR	LCC	RB-P	Avg.
Full Cache	8.24	14.86	27.13	26.42	26.91	11.37	21.64	20.48	23.16	30.50	71.42	9.83	4.12	48.67	47.31	44.28	27.27
Budget $B = 128$																	
CriticalKV	4.12	6.28	19.84	18.92	19.43	7.16	14.37	16.58	16.84	19.50	61.24	5.03	3.18	29.46	38.72	35.16	19.74
AnDPro	4.30	6.53	20.16	19.48	19.86	7.42	14.82	16.94	17.28	20.00	61.87	5.34	3.46	30.83	39.28	35.74	20.21
LORE-KV /E	4.21	9.18	20.51	19.61	**20.67**	**7.80**	15.26	**17.33**	17.92	**20.50**	62.94	**5.57**	3.27	**32.18**	**40.16**	36.42	20.85
LORE-KV /D	**4.47**	**9.71**	**20.56**	**20.28**	19.67	7.49	**15.73**	17.06	**18.34**	**20.50**	**63.78**	5.25	**3.61**	31.74	39.83	**36.87**	**20.93**
Budget $B = 256$																	
CriticalKV	4.91	7.96	21.37	20.46	21.08	8.13	16.24	**18.04**	18.56	22.50	63.86	6.18	3.47	33.62	40.83	37.24	21.53
AnDPro	5.12	8.41	21.74	20.83	21.46	8.47	16.71	17.38	18.92	**23.50**	64.61	6.47	**3.97**	34.18	41.26	37.68	21.92
LORE-KV /E	5.42	10.93	22.08	21.47	**22.31**	**8.94**	17.42	17.86	**19.63**	23.00	65.34	**6.79**	3.68	**36.47**	42.08	38.70	22.63
LORE-KV /D	**5.74**	**11.26**	**22.43**	**21.87**	22.07	8.72	**17.83**	17.64	19.28	**23.50**	**65.87**	6.62	3.83	35.92	**42.47**	**39.07**	**22.76**
Budget $B = 512$																	
CriticalKV	4.74	9.39	22.08	21.36	22.14	8.76	17.86	17.53	19.74	23.00	65.12	6.84	3.71	36.28	42.17	38.43	22.45
AnDPro	**5.64**	8.16	22.41	21.78	22.46	9.12	18.34	17.81	20.16	24.00	63.96	**7.72**	3.62	36.94	42.63	38.92	22.73
LORE-KV /E	5.09	12.48	22.91	**22.51**	**23.28**	**9.68**	19.27	**18.42**	**20.87**	25.00	66.83	6.48	3.54	**39.16**	43.52	39.98	23.69
LORE-KV /D	5.35	**12.81**	**23.37**	21.53	22.96	9.41	**19.72**	18.16	20.43	**25.50**	**67.42**	7.34	**4.02**	38.63	**43.94**	**40.38**	**23.81**
Budget $B = 1024$																	
CriticalKV	5.63	10.26	22.63	21.94	22.48	9.03	19.14	17.86	**22.16**	23.00	66.04	7.26	3.41	38.72	42.86	39.17	23.22
AnDPro	5.86	10.58	22.87	22.16	22.29	9.34	19.68	18.07	21.24	24.00	65.51	7.48	**4.27**	39.28	43.24	39.63	23.47
LORE-KV /E	6.24	13.16	22.64	23.18	**23.86**	**9.92**	20.83	**18.63**	21.94	26.00	67.16	**7.71**	3.96	**41.37**	44.18	40.26	24.44
LORE-KV /D	**6.47**	**13.58**	**23.74**	**23.49**	23.41	9.67	**21.16**	18.34	21.52	**26.50**	**67.83**	7.63	3.94	40.84	**44.57**	**40.72**	**24.59**

tain most of the cache memory, the clone can remain memory-intensive at very long context, particularly for Gemma-4-12B-it. We therefore treat this as a practical constraint of lookahead-based eviction on hybrid backbones rather than a limitation of the scoring rule.

Table 10: RULER per-task results with Llama-3.1-8B-Instruct. Abbreviations follow Table 7.

B	Method	CWE	FWE	MK1	MK2	MK3	MQ	MV	S1	S2	S3	QA-1	QA-2	VT	Avg.
Context length = 4K															
128	CriticalKV	9.40	22.00	91.20	78.60	0.00	55.80	59.40	91.20	96.40	0.00	86.20	58.00	11.20	50.72
128	AnDPro	16.00	25.40	99.20	93.20	0.20	**86.80**	**84.00**	**100.00**	99.60	0.00	86.60	59.60	30.20	60.06
128	**LORE-KV /E**	29.82	46.67	**100.00**	**95.80**	0.00	51.55	66.95	**100.00**	**99.80**	**25.40**	**88.20**	62.00	69.36	**64.27**
128	**LORE-KV /D**	**30.12**	**47.12**	99.80	95.60	**0.40**	50.37	66.55	99.20	99.40	**25.40**	88.00	**62.20**	**69.79**	64.15
256	CriticalKV	23.80	56.80	94.20	90.00	16.80	79.40	83.60	93.80	97.20	8.80	85.80	59.40	62.60	65.55
256	AnDPro	28.00	59.00	**99.40**	99.60	**17.00**	**99.60**	**99.60**	**99.60**	**99.20**	8.80	**86.00**	**60.40**	74.80	71.62
256	**LORE-KV /E**	**35.00**	**69.00**	98.60	**100.00**	15.60	86.80	92.00	98.20	97.80	**10.20**	85.60	59.40	90.60	**72.22**
256	**LORE-KV /D**	34.46	68.80	98.60	99.80	15.40	85.47	92.10	98.00	98.00	10.00	85.40	60.20	**90.68**	72.07
512	CriticalKV	43.00	74.20	96.00	93.60	68.60	87.40	89.80	95.80	97.60	54.20	85.40	60.20	84.40	79.25
512	AnDPro	45.40	75.40	98.80	99.00	**68.80**	**99.00**	**99.00**	**99.00**	**98.60**	54.20	85.60	60.80	91.60	82.71
512	**LORE-KV /E**	**49.80**	81.40	99.00	**99.40**	68.40	92.60	95.40	98.80	98.40	**55.40**	**85.80**	60.80	**95.40**	**83.12**
512	**LORE-KV /D**	49.76	**82.18**	**99.20**	99.20	68.20	92.10	95.34	98.40	98.20	55.20	**85.80**	**61.20**	94.77	83.04
1024	CriticalKV	75.40	82.40	96.80	96.00	88.00	92.60	93.80	97.00	98.00	92.60	85.00	60.00	91.00	88.35
1024	AnDPro	86.00	89.60	**99.80**	**100.00**	**98.40**	**99.80**	**99.80**	**100.00**	**100.00**	95.60	**87.80**	**62.60**	94.60	93.38
1024	**LORE-KV /E**	88.80	93.40	**99.80**	99.80	98.20	96.20	97.80	99.80	99.80	**96.40**	**87.80**	62.40	96.60	93.60
1024	**LORE-KV /D**	**88.81**	**94.74**	99.40	**100.00**	**98.40**	95.81	97.77	99.40	**100.00**	96.20	87.60	**62.60**	**97.40**	**93.70**
Context length = 8K															
128	CriticalKV	8.20	13.80	82.80	73.80	0.00	68.20	58.60	93.60	92.00	0.00	81.80	59.40	2.80	48.85
128	AnDPro	11.80	16.60	99.20	93.60	0.00	**84.80**	83.40	99.80	99.40	0.00	81.00	60.00	29.80	58.42
128	**LORE-KV /E**	33.20	**39.00**	**100.00**	**100.00**	8.20	36.60	**87.20**	**100.00**	**100.00**	**9.40**	**94.40**	**70.80**	45.80	**63.43**
128	**LORE-KV /D**	**33.75**	38.05	99.20	99.80	**8.40**	37.35	86.45	99.40	99.40	9.00	94.20	**70.80**	**46.83**	63.28
256	CriticalKV	15.00	36.80	88.60	86.80	3.60	88.60	83.40	95.40	94.60	**2.40**	**81.80**	59.80	52.80	60.74
256	AnDPro	17.20	38.60	**99.40**	99.60	**3.80**	**99.60**	**99.60**	**99.40**	**99.60**	**2.40**	81.20	**60.20**	70.20	66.98
256	**LORE-KV /E**	**23.80**	55.80	98.60	**100.00**	2.00	93.60	97.80	97.60	98.00	1.60	**81.80**	59.80	78.00	68.34
256	**LORE-KV /D**	23.76	**55.83**	98.60	99.80	2.40	93.94	99.08	98.00	98.20	1.40	81.20	59.40	**78.39**	**68.46**
512	CriticalKV	22.00	60.20	91.40	91.80	26.00	92.80	89.60	96.80	96.40	**19.40**	80.60	57.40	72.80	69.02
512	AnDPro	23.40	61.20	**97.60**	99.00	**26.20**	**99.00**	**99.00**	**99.00**	**99.00**	**19.40**	80.20	**57.60**	83.00	72.58
512	**LORE-KV /E**	**27.00**	70.60	97.20	**100.00**	25.20	95.80	98.20	98.20	98.20	19.00	**80.80**	**57.60**	87.20	**73.46**
512	**LORE-KV /D**	26.59	**70.65**	97.40	99.80	25.40	95.42	98.46	98.60	97.80	18.80	**80.80**	57.40	**87.46**	73.43
1024	CriticalKV	**65.20**	75.80	94.80	94.80	66.80	95.40	93.80	97.60	97.20	75.20	82.60	58.80	84.40	83.26
1024	AnDPro	45.00	81.60	**100.00**	99.80	76.40	**99.60**	**99.80**	**100.00**	**100.00**	**77.20**	83.40	**61.80**	92.60	85.94
1024	**LORE-KV /E**	47.20	87.20	99.80	**100.00**	75.80	97.80	99.20	99.40	99.40	76.80	**83.80**	61.60	**95.20**	**86.40**
1024	**LORE-KV /D**	46.69	**87.62**	99.20	99.20	**76.60**	97.98	98.01	99.60	99.60	**77.20**	**83.80**	61.40	95.08	86.31
Context length = 16K															
128	CriticalKV	2.80	30.00	68.60	63.20	0.00	61.20	49.00	93.80	86.80	0.00	78.80	51.60	0.00	45.06
128	AnDPro	2.80	32.00	85.60	83.20	0.00	61.20	69.60	**100.00**	94.40	0.00	78.20	52.40	27.00	52.80
128	**LORE-KV /E**	**4.20**	34.60	**90.80**	**88.40**	**1.20**	**65.40**	**74.00**	**100.00**	**100.00**	**1.20**	**83.20**	56.00	**29.60**	**56.05**
128	**LORE-KV /D**	3.65	**35.13**	90.40	88.00	0.80	65.38	73.86	99.80	**100.00**	1.00	83.00	**56.40**	29.34	55.90
256	CriticalKV	7.60	53.00	75.00	77.00	**1.20**	**95.00**	80.40	95.60	91.20	1.00	79.00	52.20	49.00	58.25
256	AnDPro	7.60	54.20	86.00	90.00	**1.20**	**95.00**	**93.60**	**99.60**	96.20	1.00	78.40	52.80	66.60	63.25
256	**LORE-KV /E**	14.80	**70.20**	**86.20**	99.00	0.00	88.00	91.00	98.60	**96.60**	**1.40**	**79.80**	53.00	76.80	65.80
256	**LORE-KV /D**	**15.95**	69.67	86.00	**99.40**	0.60	87.16	91.80	99.40	96.40	**1.40**	79.40	**53.40**	**77.12**	**65.98**
512	CriticalKV	16.00	73.20	80.80	86.60	**24.20**	**98.20**	88.80	96.80	93.60	18.20	78.60	51.40	69.60	67.38
512	AnDPro	16.00	74.00	**87.20**	94.20	**24.20**	**98.20**	**96.40**	**99.00**	96.40	18.20	78.20	**51.80**	79.80	70.28
512	**LORE-KV /E**	20.00	**82.60**	**87.20**	**98.80**	23.20	94.40	95.20	98.40	96.60	18.20	**78.80**	**51.80**	**85.60**	**71.60**
512	**LORE-KV /D**	**20.22**	82.50	87.00	98.40	23.00	94.49	94.81	**99.00**	**97.20**	**18.40**	**78.80**	51.20	85.19	71.55
1024	CriticalKV	**61.20**	**84.20**	87.60	92.00	**65.40**	96.20	90.00	97.60	95.20	**74.40**	**81.40**	55.00	82.20	**81.72**
1024	AnDPro	24.80	75.80	**99.80**	99.40	57.80	**99.40**	**98.80**	**100.00**	**100.00**	51.00	80.80	55.80	88.40	79.37
1024	**LORE-KV /E**	27.20	81.00	**99.80**	**100.00**	57.40	97.00	98.00	99.60	**100.00**	51.00	81.20	55.80	91.80	79.98
1024	**LORE-KV /D**	27.48	81.22	99.60	99.80	57.80	97.81	97.71	99.40	**100.00**	50.60	81.20	**56.20**	**93.36**	80.17

E.3 LongBench Results on Additional Backbones

To test whether LORE-KV generalizes beyond Mistral-7B-Instruct-v0.2, we evaluate Llama-3.1-8B-Instruct, Qwen2.5-14B-Instruct, and Ministral-3-8B-Instruct-2512. We follow the same evaluation protocol as Appendix E.1 and keep the budgets, observation window, chunk size, and core configuration of Table 4 fixed. Qwen2.5 prompts are truncated to 31,500 tokens to respect its context limit; the other backbones are evaluated without this truncation. For Ministral-3, our implementation reproduces the reference model exactly under uncompressed inference, including identical logits and top-1 predictions.

Table 11: LongBench results on Qwen2.5-14B-Instruct. Dataset abbreviations follow Table 5.

	Single-Doc QA			Multi-Doc QA			Summ.			Few-shot			Synth.		Code		
Method	NQA	Qasp	MF	HQA	2WQA	Musq	GovR	QMS	MN	TREC	TQA	SAMS	PC	PR	LCC	RB-P	Avg.
Full Cache	29.23	45.81	52.89	62.41	57.46	38.20	29.66	23.78	21.90	76.50	90.23	47.27	8.56	98.75	60.37	46.84	49.37
Budget $B = 128$																	
CriticalKV	26.28	30.80	41.67	58.54	53.39	35.81	19.23	21.29	16.34	63.50	86.20	44.64	8.67	92.33	55.82	40.96	43.47
AnDPro	24.83	37.09	46.49	60.61	54.50	35.18	19.33	21.31	16.93	71.50	88.27	45.11	8.21	97.25	57.72	43.47	45.49
LORE-KV /E	29.10	**38.64**	50.07	63.07	**58.70**	**36.91**	**23.55**	23.21	20.29	76.50	89.06	**46.01**	**10.32**	**97.58**	**60.53**	46.98	48.16
LORE-KV /D	**29.32**	38.28	**51.55**	**63.12**	58.28	36.82	22.66	**23.68**	**21.29**	**77.00**	**89.13**	45.72	9.94	96.53	60.34	**48.22**	**48.24**
Budget $B = 256$																	
CriticalKV	26.54	37.16	47.35	59.80	54.37	37.73	21.37	22.25	18.26	71.00	88.25	45.93	7.67	98.75	58.76	45.03	46.26
AnDPro	27.13	**41.92**	**51.26**	61.70	55.49	37.71	21.90	21.74	18.55	75.50	89.34	45.82	9.04	98.33	61.10	46.63	47.70
LORE-KV /E	28.93	41.41	50.55	62.85	59.10	**38.03**	**25.43**	23.55	**21.42**	**76.50**	89.58	46.47	11.75	98.75	**61.30**	47.60	48.95
LORE-KV /D	**28.96**	40.99	50.74	**64.77**	**59.43**	38.01	25.23	**23.89**	20.31	75.50	**90.36**	**47.25**	**12.48**	**99.51**	60.49	**47.81**	**49.11**
Budget $B = 512$																	
CriticalKV	26.88	41.79	50.08	62.01	55.80	37.00	23.40	22.78	19.42	74.00	89.56	46.62	8.40	98.50	59.80	46.20	47.64
AnDPro	28.50	44.45	52.25	61.47	57.20	39.11	23.50	22.73	20.33	76.50	89.74	46.69	9.88	98.25	61.25	47.16	48.69
LORE-KV /E	**29.00**	44.87	52.62	62.45	**58.15**	**39.30**	**25.85**	**23.45**	21.10	76.50	**89.92**	**46.95**	**10.25**	98.75	61.50	47.60	49.27
LORE-KV /D	28.86	**45.51**	**53.35**	**62.81**	56.82	38.81	25.03	23.21	**21.65**	**77.50**	89.89	46.07	10.13	**98.76**	**62.74**	**48.10**	**49.33**
Budget $B = 1024$																	
CriticalKV	28.00	43.69	52.00	61.50	56.80	38.00	26.00	23.10	20.50	75.50	89.90	47.00	8.50	**98.75**	60.20	46.60	48.50
AnDPro	**29.65**	45.31	53.53	62.50	57.66	38.46	25.34	23.24	21.09	**76.50**	90.14	47.40	9.20	**98.75**	61.50	47.70	49.25
LORE-KV /E	29.40	**45.45**	53.70	62.65	57.90	**38.60**	26.10	23.40	21.25	**76.50**	**90.18**	**47.50**	9.50	**98.75**	**61.70**	**47.80**	49.40
LORE-KV /D	29.27	45.27	**53.87**	**63.39**	**59.07**	37.76	**26.45**	**24.48**	**22.57**	75.50	89.46	46.55	**9.66**	98.72	61.53	47.51	**49.44**

Tables 6, 11, and 12 show the same overall trend as Section 4.1: LORE-KV is most beneficial under aggressive compression, while the margin decreases as the cache budget grows. On Llama, the largest improvements at B=128 occur mainly on multi-hop and long-document QA, while code completion and a few retrieval/few-shot tasks are less consistent; the corresponding Mistral results are given in Table 1.

The trend also transfers to the two newer backbones. At B=128, LORE-KV improves over AnDPro by approximately 2.75 points on Qwen2.5-14B and 0.56 points on Ministral-3, as shown in Tables 11 and 12, respectively, with the gap becoming small by B=1024. This operating-point dependence is consistent with the proposed mechanism: future-query estimation matters most when eviction is aggressive. The smaller gain on Ministral-3 in Table 12 is also consistent with its long-context training and position-dependent query scaling, which reduce the prompt-to-generation query mismatch that LORE-KV is designed to correct. On Qwen2.5, a few compressed settings in Table 11 slightly exceed the full-cache score; we treat these small differences as possible distractor filtering or run variation rather than evidence that compression is intrinsically better than full attention. Across all three additional backbones, Tables 6, 11, and 12 show that the LORE-KV /E and LORE-KV /D variants remain close, supporting the conclusion that the main gain comes from the future-query reference rather than a particular reliability parameterization.

Qwen3.5-9B and Gemma-4-12B-it show consistent gains across the hybrid-attention setting (Tables 8, 9, and 2). LORE-KV remains strongest under aggressive compression, with notable gains on tasks requiring answer-time evidence retrieval. As the budget increases, Qwen3.5 approaches full-cache quality, while Gemma-4-12B-it maintains a more stable margin because its bounded sliding-window layers limit how quickly additional budget reduces pressure on the full-attention cache.

E.4 RULER: Protocol and Per-Task Results

Tables 7 and 10 report the full RULER results across 13 tasks, context lengths of 4K–16K, and $B \in \{128, 256, 512, 1024\}$. Following the official RULER protocol (Hsieh et al., 2024), we evaluate 500 examples per task. Multi-reference synthetic tasks use `string_match_all`, which assigns fractional credit across reference answers, while QA uses `string_match_part`; scores are averaged over examples and rounded to two decimals, allowing finer granularity than the 0.2-point step of binary accuracy. The overall trend matches the LongBench results: LORE-KV provides its largest gains under tight cache budgets, particularly on aggregation-heavy tasks such as variable

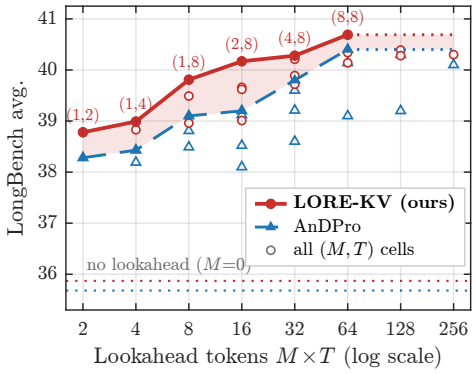
Table 12: LongBench results on Ministral-3-8B-Instruct-2512. Dataset abbreviations follow Table 5.

	Single-Doc QA			Multi-Doc QA			Summ.			Few-shot			Synth.		Code		
Method	NQA	Qasp	MF	HQA	2WQA	Musq	GovR	QMS	MN	TREC	TQA	SAMS	PC	PR	LCC	RB-P	Avg.
Full Cache	28.31	45.48	46.07	61.03	56.85	40.13	22.79	16.82	19.75	74.00	91.38	44.51	14.50	100.00	69.26	63.86	49.67
Budget $B = 128$																	
CriticalKV	25.35	31.84	40.00	55.37	52.91	34.22	18.67	16.83	15.83	58.00	90.64	38.89	**14.50**	99.25	64.39	59.94	44.79
AnDPro	25.78	37.46	42.15	59.06	54.72	37.54	18.87	16.62	16.32	70.50	90.81	39.11	13.50	99.50	64.98	60.86	46.74
LORE-KV /E	**26.38**	**38.66**	42.15	60.56	**56.02**	**38.74**	**19.67**	16.82	**16.72**	**71.00**	91.11	39.41	14.00	**99.60**	65.18	60.76	**47.30**
LORE-KV /D	26.15	37.96	**42.96**	**61.01**	55.38	37.43	19.11	**18.13**	16.01	70.50	**91.62**	**39.45**	14.14	99.08	**65.68**	**61.23**	47.24
Budget $B = 256$																	
CriticalKV	25.44	36.12	40.93	58.49	54.76	37.51	20.10	16.82	17.27	69.50	90.63	40.88	13.50	**100.00**	66.45	61.31	46.86
AnDPro	25.86	39.78	43.75	60.30	55.25	39.14	20.05	17.10	17.50	72.50	91.54	41.80	14.50	**100.00**	67.63	**62.55**	48.08
LORE-KV /E	26.26	**40.58**	43.65	61.10	**55.85**	39.64	**20.65**	**17.25**	17.75	72.50	91.64	41.65	14.70	**100.00**	67.73	**62.55**	48.34
LORE-KV /D	**26.68**	40.28	**45.30**	**61.51**	54.03	**40.97**	19.98	16.74	**17.90**	**73.00**	**91.76**	**42.06**	**15.93**	99.19	**67.98**	62.54	**48.49**
Budget $B = 512$																	
CriticalKV	26.15	41.00	43.22	60.11	55.65	37.63	20.87	17.48	18.31	73.50	91.29	41.62	**16.00**	**100.00**	68.48	63.49	48.43
AnDPro	26.20	42.12	**45.39**	60.79	56.81	39.91	20.69	17.48	18.36	73.50	91.63	42.81	14.50	99.50	68.81	64.01	48.91
LORE-KV /E	26.45	**42.47**	45.34	61.14	**57.06**	40.21	20.99	**17.56**	18.46	73.50	**91.68**	42.71	14.60	99.60	**68.89**	64.01	**49.04**
LORE-KV /D	**26.97**	41.36	44.92	**61.96**	55.14	**40.51**	**21.41**	16.98	**18.63**	**74.00**	91.41	**43.47**	13.11	99.90	68.36	**64.13**	48.89
Budget $B = 1024$																	
CriticalKV	27.78	43.72	45.60	**61.47**	56.30	39.21	21.57	**17.31**	19.34	73.50	**91.38**	42.45	**14.50**	**100.00**	69.85	63.57	49.22
AnDPro	28.11	45.07	46.82	60.53	58.02	40.32	21.39	16.88	19.33	**74.00**	**91.38**	42.94	**14.50**	**100.00**	69.88	**64.82**	49.62
LORE-KV /E	28.19	**45.17**	46.79	60.65	58.10	40.42	21.47	16.90	19.35	**74.00**	**91.38**	42.86	**14.50**	**100.00**	**69.90**	64.77	49.65
LORE-KV /D	**28.58**	44.95	**48.04**	59.93	**58.34**	**40.80**	**21.76**	17.05	**19.63**	73.50	91.37	**43.51**	14.45	99.75	69.43	64.45	**49.72**

Table 13: **Lookahead geometry and compute scaling.** LongBench on Mistral-7B-Instruct-v0.2 at B=128: **(a)** LORE-KV /E across trajectory count M and horizon T; **(b)** quality frontier under a matched lookahead-token budget versus AnDPro.

M	T=2	T=4	T=8	T=16	T=32
0	35.87 (independent of T)				
1	38.78	38.99	39.81	39.62	39.89
2	38.83	39.49	40.17	40.21	40.14
4	38.96	39.66	40.28	40.34	40.28
8	39.01	39.72	**40.69**	40.39	40.30

(a) LORE-KV /E over M and T



(b) Pareto frontier at matched lookahead-token budget

tracking (VT), frequent-word extraction (FWE), and common-word extraction (CWE), while multi-query retrieval (MQ) and multi-key retrieval (MK3) remain more challenging. The matched RULER ablations fix $T = 8$ and vary the remaining components. As one anchor point, at 4K and $B = 128$, increasing the number of lookahead particles from $M = 1$ to $M = 4$ improves the average from 56.90 to 58.40. The same overall trend persists at larger budgets, whereas reliability weighting, adaptive head budgets, and diversity-aware selection produce comparatively small changes. Together, these results reinforce that the primary improvement comes from estimating the future-query reference rather than from the optional plug-ins. The complementary LongBench (M, T) sweep is reported in Table 13.

Table 14: LongBench ablations (16-dataset average) with Mistral-7B-Instruct-v0.2.

Axis	Variant	B=128	B=256	B=512	B=1024
Core	**LORE-KV /E core**	40.28	41.38	41.86	42.20
	LORE-KV /D core	40.21	41.23	41.68	42.39
Lookahead budget	M=0 (prompt window only; no lookahead term)	35.87	37.50	38.70	39.50
	M=1 (single surrogate response)	39.81	40.85	41.56	42.03
	M=2	40.17	40.18	41.10	41.22
	M=8	40.69	40.97	41.52	41.65
Reliability weighting	Log-probability only	40.29	41.39	41.30	41.40
	Entropy + disagreement	40.33	41.40	41.84	42.17
Perturbation objective	Projected norm, $\|\mathbf{W}^O \Delta\mathbf{o}\|_2^2$	40.28	41.38	41.86	42.20
	Accumulated attention mass	40.19	41.27	41.65	42.13
	Unprojected norm, $\|\Delta\mathbf{o}\|_2^2$	40.22	41.22	41.65	42.13
	AnDPro scoring with the same continuations	*40.24*	*41.30*	*41.74*	*42.15*
Plug-ins	+ Adaptive head budgets	40.28	41.35	41.72	41.72
	+ Adaptive budgets and diversity	40.30	40.33	41.50	41.80
	+ Robust Aggregation	40.34	41.41	41.85	42.17

Table 15: RULER ablations with Mistral-7B-Instruct-v0.2 (13-task averages).

				B=128			B=256		
M	Reliability weighting	Head budgets	Selection	4K	8K	16K	4K	8K	16K
AnDPro (strongest baseline)				54.85	50.06	45.20	71.05	64.08	57.91
1	unit weight	uniform	top-K	56.90	54.29	52.15	68.45	62.77	57.70
		adaptive	top-K	56.92	54.31	52.18	68.50	62.81	57.75
		adaptive	diversity	56.95	54.34	**52.20**	68.55	62.85	57.78
4	Log-probability only	uniform	top-K	58.39	55.72	50.98	70.08	64.45	58.85
		adaptive	top-K	58.28	55.68	50.95	70.04	64.42	58.82
		adaptive	diversity	58.31	55.70	50.97	70.07	64.45	58.85
4	**LORE-KV /E**	adaptive	top-K	58.37	55.74	51.02	71.45	65.46	59.88
		uniform	**top-K**	**58.40**	**55.77**	51.05	**71.51**	**65.52**	**59.94**
4	**LORE-KV /D**	adaptive	top-K	58.28	55.70	50.96	69.72	64.50	58.92
		adaptive	diversity	58.31	55.72	50.98	69.75	64.53	58.95
		uniform	**top-K**	58.31	55.74	50.97	69.77	64.56	58.94

E.5 NIAH Grid

Figure 8 reports exact match and word recall for Llama-3.1-8B-Instruct at $B = 1024$ over context lengths of 4K–32K and needle depths of 0%–100% in 10% steps, with one prompt per cell. All methods use the same budget, observation window, and protected token set. Results are discussed in Section 4.2.

F Mechanism Analysis

F.1 Measurement Protocol

All mechanism analyses use the same states and scoring code as the deployed eviction pipeline, with additional instrumentation used only for analysis. For each prompt, we capture the post-RoPE prompt-window queries, the full prefill $\mathbf{K}^{l,h}, \mathbf{V}^{l,h}$, and the M lookahead trajectories together with $c_u^{(m)}$, $H_u^{(m)}$, $V_u^{(m)}$, $D_m, \hat{D}_m$, and ω_m. We also greedy-decode once with the *full* cache to obtain the future queries $\mathcal{Q}_{\text{future}}$, which serve only as an analysis-time oracle. These full-cache states are

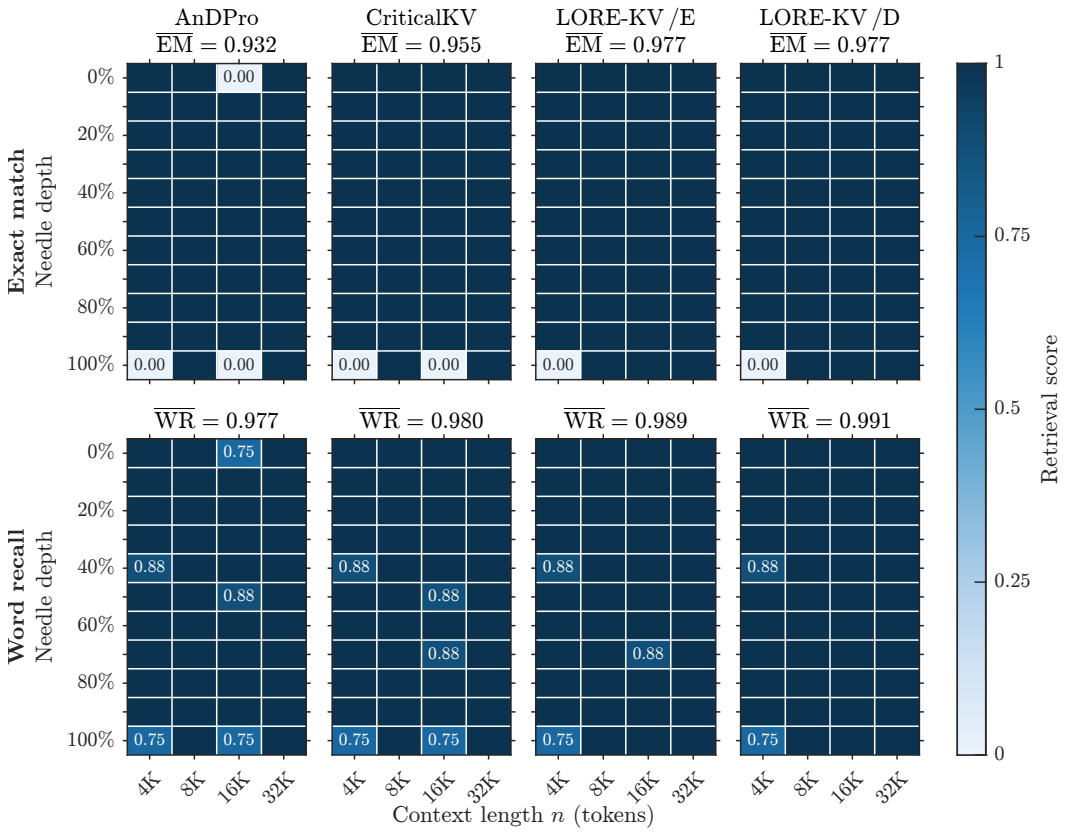


Figure 8: **NIAH retrieval.** Exact-match and word-recall scores across needle depths and 4K–32K contexts on Llama-3.1-8B-Instruct at B=1024. Panel means are shown above each plot.

not retained during normal LORE-KV inference and are excluded from the reported cache budget. The reliability channels of Eq. 3 are stored separately, including disagreement when $\lambda_D = 0$, and per-step trajectory statistics require only MT floats per prompt.

For token-level analysis, we use the same output-perturbation criterion as LORE-KV:

$$s_{q,i}^{l,h} = \left| \frac{a_{q,i}^{l,h}}{1 - a_{q,i}^{l,h} + \epsilon} \right| \left\| \mathbf{W}_{l,h}^{O} \mathbf{v}_{i}^{l,h} - \mathbf{W}_{l,h}^{O} \mathbf{o}_{q}^{l,h} \right\|_2 . \tag{12}$$

Scores are aggregated using Eq. 7, with all criteria sharing the same captured states and selector. **Attention** uses SnapKV-style prompt-window attention; **AnDPro window** uses prompt-side projected perturbation; **LORE-KV** uses the projected perturbation score with LORE-KV /E weighting; and **Single-lookahead** uses one unweighted trajectory as a score-level proxy for Lookahead Q-Cache. These controlled comparisons measure how the query reference changes token ranking (Figure 9), agreement with the future-query oracle (Figure 10), retained evidence (Figure 11), and behavioral fidelity across cache budgets (Figure 13). Protected positions $\mathcal{P}$ are always retained, and all remaining tokens use the same uniform per-head top-$\min(B, n)$ selector, so differences are attributable to the scoring/query criterion rather than downstream selection.

Equation 12 uses magnitude rather than the squared magnitude of Eq. 6; this preserves single-query rankings but changes the energy scale in Figure 12. For these analyses, chunking, adaptive head budgets, and diversity-aware selection are disabled, and B includes the $\sim$33 protected positions. Figure 12 additionally sums energy over all positions and reports GQA-expanded attention heads, while Figure 11 identifies answer tokens by exact token-id match to the gold answer; this may underestimate absolute evidence retention but affects all criteria equally.

We use three complementary diagnostics. Query coverage measures how closely an estimated query set matches the true future queries:

$$\mathrm{Cov}\left(\mathcal{Q}_{\mathrm{future}}; \{\hat{\mathcal{Q}}_s\}, \boldsymbol{\pi}\right) = \frac{1}{H}\sum_h \frac{1}{T_{\mathrm{f}}}\sum_t \sum_s \pi_s \max_{\hat{\mathbf{q}}\in\hat{\mathcal{Q}}_s} \cos\left(\mathbf{q}_{t,h}^{\mathrm{future}}, \hat{\mathbf{q}}\right). \tag{13}$$

The oracle score $S^\star$ applies Eq. 7 directly to $\mathcal{Q}_{\mathrm{future}}$, and retention agreement is measured by

$$\mathrm{Recall}\,@B(S, S^\star) = \frac{|\mathrm{top}_B(S) \cap \mathrm{top}_B(S^\star)|}{B}. \tag{14}$$

Finally, behavioral fidelity measures how closely the compressed cache reproduces the full-cache attention output:

$$F(B) = \mathbb{E}_{\mathbf{q}\sim\mathcal{Q}_{\mathrm{future}}} \cos\left(\mathbf{o}_q^{\mathcal{R}(B)}, \mathbf{o}_q\right). \tag{15}$$

Queries are unit-normalized for the cosine-based metrics, and the prompt-to-future mismatch $\mathrm{Mis}(X)$ is defined in Section 4.5. Unless stated otherwise, all analyses use Mistral-7B-Instruct-v0.2 on LongBench and 4K RULER with the core LORE-KV /E configuration.

F.2 Future-Query Coverage

Figure 1 examines whether sampled lookahead queries better represent the queries encountered during generation. We compare the prompt-window queries $\mathcal{Q}_{\mathcal{W}}^{l,h}$, the sampled lookahead queries $\mathcal{Q}_{\mathrm{look}}^{(m),l,h}$, and the true future queries $\mathcal{Q}_{\mathrm{future}}$, where $\mathcal{Q}_{\mathrm{future}}$ is obtained only retrospectively from full-cache decoding and is never available to the eviction rule.

Panel (a) visualizes these three query sets for one representative layer and KV head (layer 16, head 18) after projection onto the first two principal components. The sampled trajectories move toward the region occupied by the true future queries, illustrating the prompt-to-future query mismatch that motivates LORE-KV.

Panel (b) quantifies this effect using the coverage metric in Eq. 13. The prompt-window reference achieves 0.785 coverage, while adding sampled trajectories progressively increases coverage of $\mathcal{Q}_{\mathrm{future}}$, with most of the gain obtained by approximately $M{=}4$. The sweep uses nested prefixes of the same eight sampled trajectories for each prompt, so changes reflect increased future-query coverage rather than different random draws.

Coverage measures geometric support in query space rather than downstream eviction quality. Consequently, the small change from $M{=}0$ to $M{=}1$ ($0.785 \to 0.786$) does not contradict the substantial end-to-end improvement from a single response-side rollout in Tables 14 and 15: one trajectory can already provide a more relevant query reference, while additional trajectories improve coverage of alternative plausible futures. Together, these results support the central design of LORE-KV: estimating token utility over multiple plausible response-side query trajectories provides a better reference for eviction than relying on prompt-side queries alone.

F.3 Retention-Order Shift

Figure 9 tests whether lookahead changes token selection rather than simply rescaling prompt-side scores. At $B{=}128$, attention and LORE-KV /E retain only 0.82 of the same tokens on LongBench and 0.86 on RULER, despite high overall Spearman rank correlation ($\rho{=}0.94$ and 0.87). Thus, lookahead preserves much of the global ranking while changing the tokens near the retention boundary. In contrast, single-lookahead and LORE-KV /E are much more similar (0.97/0.92 top-B overlap on LongBench/RULER), showing that most of the ranking shift already comes from replacing prompt-side queries with response-side queries.

Figure 10 compares each criterion with the future-query oracle under the same selection rule. At $B{=}128$ on LongBench, LORE-KV /E increases oracle-retention agreement from 0.50 for the window-only control to 0.63, while also reducing full-cache output error from 0.177 to 0.138. The same advantage persists across budgets and on RULER, showing that response-side lookahead yields a token ranking that better matches future decoding needs.

These controlled comparisons isolate the source of the gain: changing the query reference toward plausible future decoding states produces substantially better agreement with the future-query oracle, whereas prompt-side attention- and value-based criteria remain much closer to one another.

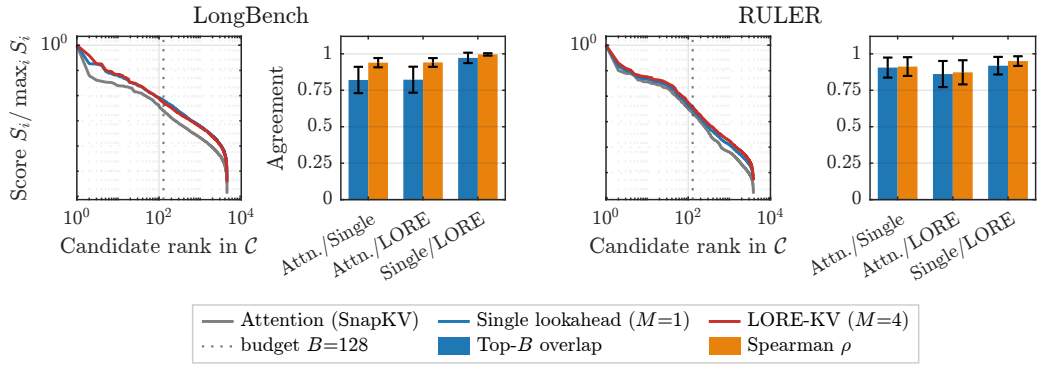


Figure 9: **Retention-order shift.** Candidate-score profiles, top-B overlap, and Spearman correlation across eviction criteria. Mistral-7B-Instruct-v0.2, $B = 128$, $M = 4$.

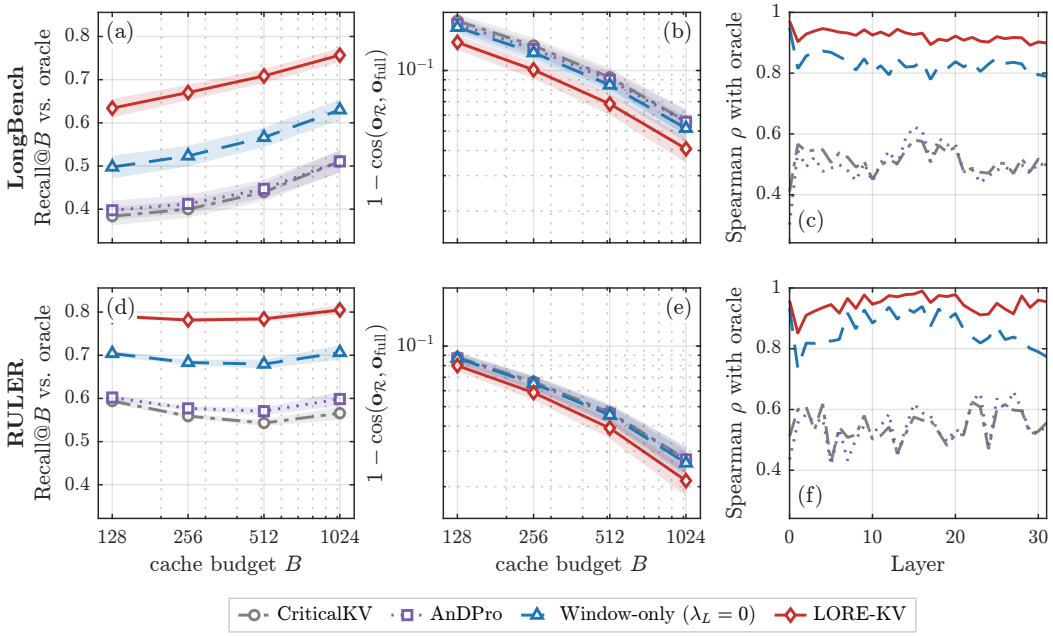


Figure 10: **Criterion comparison. (a,d)** Recall@B against the future-query oracle. **(b,e)** Attention-output error (lower is better). **(c,f)** Layer-wise Spearman correlation with the oracle.

This supports the central hypothesis of LORE-KV that *which queries define token importance* is more important than further refining the prompt-side scoring rule.

F.4 Reliability Weighting: Robustness Beyond the Query Reference

Reliability weighting acts only on the already-sampled trajectories and therefore adds no extra test-time decoding. Trajectory disagreement $\hat{D}_m$ provides an additional reliability signal across sampled futures. Its role is deliberately conservative: the reliability signals are only weakly correlated with oracle retention utility Recall@B (Eq. 14), particularly on RULER, and hard-filtering low-reliability trajectories does not reduce the residual query-coverage error $1 - \text{Cov}$. We use reliability as a soft correction through ω_m in Eq. 3, reducing the influence of uncertain or atypical trajectories without discarding plausible futures.

Importantly, the main robustness gain comes from *covering multiple plausible futures*. Figure 1(b) shows that future-query coverage increases from 0.785 for the prompt-window reference to 0.802 at M=2 and 0.811 at M=8, with gains saturating near M=4. This shows that sampling multiple

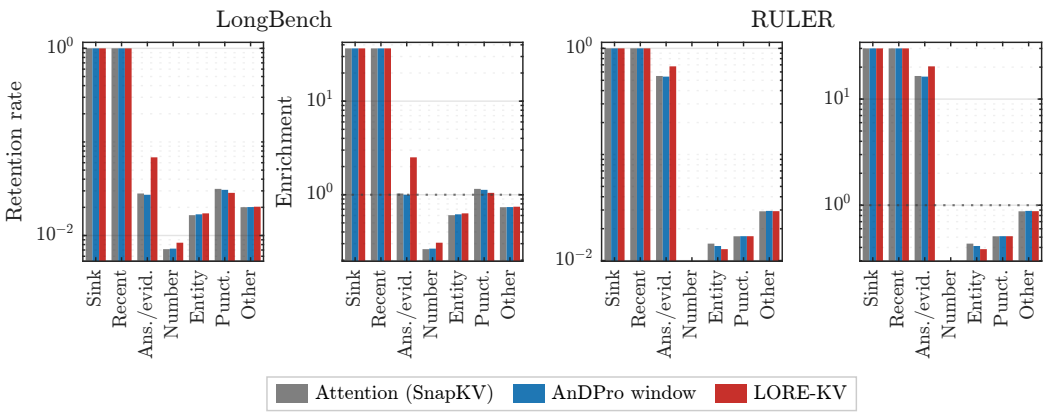


Figure 11: **Retention by token class.** Mean retained fraction (left) and class enrichment $P(c \mid \mathcal{R})/P(c \mid X)$ (right).

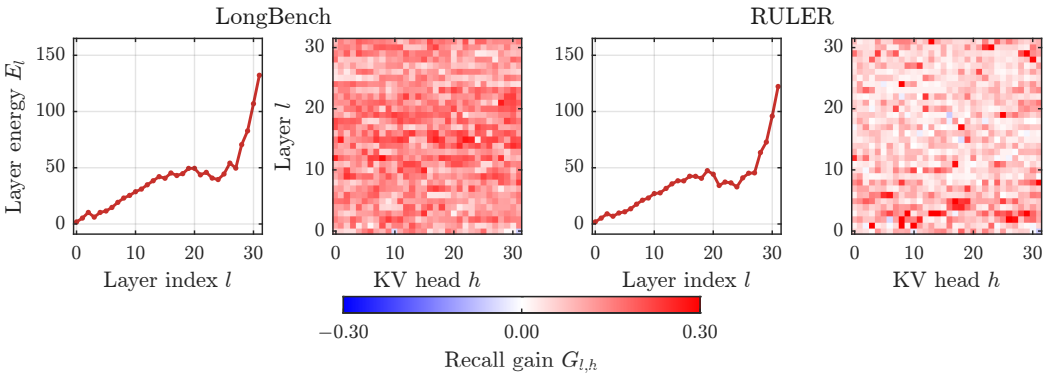


Figure 12: **Future-query signal across layers and heads.** Layer perturbation energy (left) and retention-recall gain from multi-lookahead queries (right). Mistral-7B-Instruct-v0.2, $B = 128$, $M = 4$.

trajectories provides broader coverage of the future-query space than relying on a single prompt-side reference. Reliability weighting then acts as a complementary soft correction, reducing the influence of uncertain or atypical trajectories without discarding plausible futures.

The oracle diagnostics in Figure 10 reinforce this interpretation. At B=128 on LongBench, Recall@B against the future-query oracle increases from 0.39 for the prompt-side criteria (CriticalKV and AnDPro) and 0.50 for the λ_L=0 window-only control to 0.63 for LORE-KV /E. The same ordering persists across budgets and on RULER. Similarly, the attention-output error $1 - \cos(\mathbf{o}_{\mathcal{R}}, \mathbf{o}_{\text{full}})$ decreases from 0.177 to 0.138 at B=128, while LORE-KV /E maintains substantially higher layer-wise agreement with the oracle (Spearman $\rho \approx$ 0.90-0.95) than either the window-only control ($\approx$ 0.80-0.85) or prompt-side criteria ($\approx$ 0.45-0.60). Finally, LORE-KV /E and LORE-KV /D remain within 0.19 LongBench points at every budget (Table 14). Together, these results show that LORE-KV is robust primarily because it estimates utility over multiple plausible future queries; entropy and disagreement weighting provide a secondary safeguard against noisy or atypical samples rather than a separate source of performance.

F.5 Retained-Token Composition

Figure 11 examines which token classes are preserved after compression. The protected sink and recent-token classes are identical across methods, while the main difference is in answer-bearing evidence. On LongBench, LORE-KV /E increases answer/evidence enrichment from approximately 1× for the prompt-side baselines to 2.51×, with the corresponding retention rate rising from about

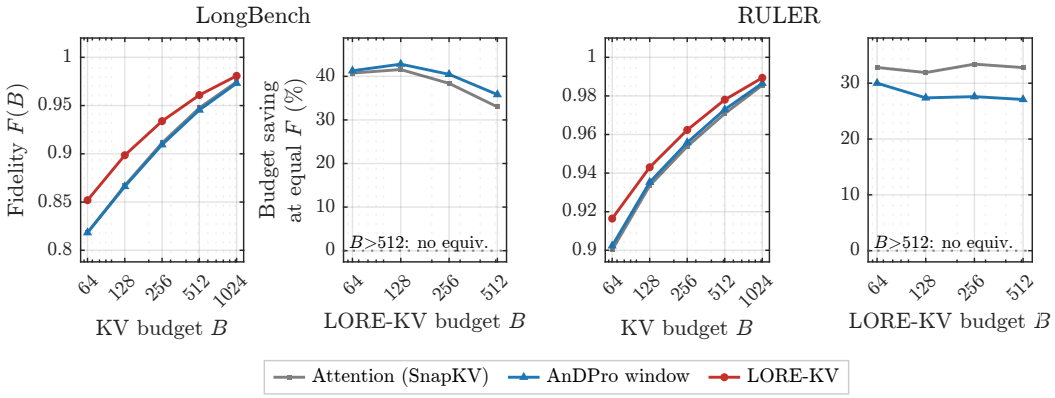


Figure 13: **Budget equivalence under behavioral fidelity.** Fidelity across cache budgets (left) and baseline cache savings required to match LORE-KV /E (right).

0.028 to 0.069. The same trend holds on RULER, where answer/evidence enrichment increases to 20.3 compared with 16.5-16.3 for the baselines. Other token classes change only modestly. This consistent shift across benchmarks shows that future-query lookahead does not indiscriminately retain more tokens; it preferentially preserves evidence that is relevant to subsequent decoding, providing a direct mechanism for the downstream gains reported in Section 4.2.

F.6 Layer- and Head-Level Structure

Figure 12 shows that the future-query signal is distributed across the network rather than concentrated in a single component. Perturbation energy generally increases with depth, while replacing window-only queries with multi-lookahead queries improves retention recall for most heads, with the largest gains concentrated in a subset of middle-depth heads. The same qualitative pattern appears on both LongBench and RULER, supporting the robustness of the lookahead signal across tasks. This non-uniformity is consistent with head specialization (Feng et al., 2025b; Fu et al., 2025) and motivates the adaptive allocation in Eq. 11; however, its modest additional gain indicates that the core LORE-KV /E score already captures most of the benefit.

F.7 Budget Equivalence in Behavioral Fidelity

Figure 13 complements Table 16 by measuring cache quality continuously with the behavioral fidelity $F(B)$ of Eq. 15. Across the full budget range, LORE-KV /E consistently preserves full-cache behavior better than the prompt-side baselines on both LongBench and RULER, with the largest advantage at tighter budgets and the same ordering maintained as B increases.

Table 16: Baseline budgets matching LORE-KV /E on LongBench (Mistral-7B-Instruct-v0.2).

LORE-KV /E budget	128	256	512	1024
Ada-KV	∼332	∼745	>1024	>1024
CriticalKV	∼307	∼709	>1024	>1024
AnDPro	**∼208**	**∼414**	**∼743**	>1024
Saving (%)	38.5	38.1	31.1	> 0

Matching the fidelity of LORE-KV /E requires approximately 33-43% more retained budget on LongBench and 27-34% on RULER. This shows that the advantage is not tied to a single operating point, but persists across compression levels. These fidelity results are consistent with the task-level performance trends reported for LongBench in Table 1 and for RULER in Figure 3 and Tables 7 and 10. If a baseline does not reach the target fidelity within the tested budgets, we report only a lower bound on the required budget saving.

G The Two Reliability Variants in the $(\beta, \gamma, \lambda_D)$ Plane

The LORE-KV /E and LORE-KV /D variants differ only in the reliability coefficients defined in Table 4, with end-to-end comparisons reported in Tables 1, 6, 14, and 15, and in Figure 3. At B=128,

both variants re-weight the same nested prefixes from a single eight-particle draw per prompt, keeping the sampled trajectories and test-time compute fixed.