DISTANCE-KV: EXPLOITING RELATIVE DISTANCE FOR EFFICIENT LONG-CONTEXT INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

The memory usage and decoding latency of LLM inference grow rapidly with context length. To reduce these costs, key-value (KV) cache compression methods selectively retain cached states based on token importance or differences in attention patterns across heads. However, we discover that retrieval capability varies substantially with relative distance, even within the same attention head. To exploit this structure, we introduce Distance-KV, which learns a static KV retention pattern over the joint space of layers, attention heads, and relative distances. The pattern is learned offline with the language model frozen and reused across inputs to prune and compact the KV cache without online importance scoring. Across three backbone models and four long-context benchmarks, Distance-KV consistently achieves the best overall performance among competing KV cache compression methods, exceeding the strongest compression baseline by up to 9.3 points on RULER at 128K. On Llama-3.1-8B-Instruct at 128K, Distance-KV reduces KV cache memory by 65.4% and achieves a 1.66× decoding speedup relative to Dense. Together, these results identify relative distance as an important structural dimension for understanding how LLMs retrieve information over long contexts and for designing more efficient inference methods.

1 INTRODUCTION

Extending the context length enables large language models (LLMs) to process long documents, code repositories, and multi-turn interactions, but the resulting growth of the key-value (KV) cache substantially increases memory use and decoding latency (Gemini Team, 2025; Yang et al., 2025). To reduce this overhead, prior work has explored KV retention at multiple granularities, including individual tokens, layers, and attention heads (Zhang et al., 2023; Cai et al., 2025; Ge et al., 2024). In particular, work on retrieval heads identifies a small subset of attention heads as critical for retrieving relevant information from long contexts (Wu et al., 2025). Existing head-level methods assign each head a single score, label, or sparse attention rule, without explicitly modeling how its retrieval capability varies with relative distance (Xiao et al., 2025; Fu et al., 2024).

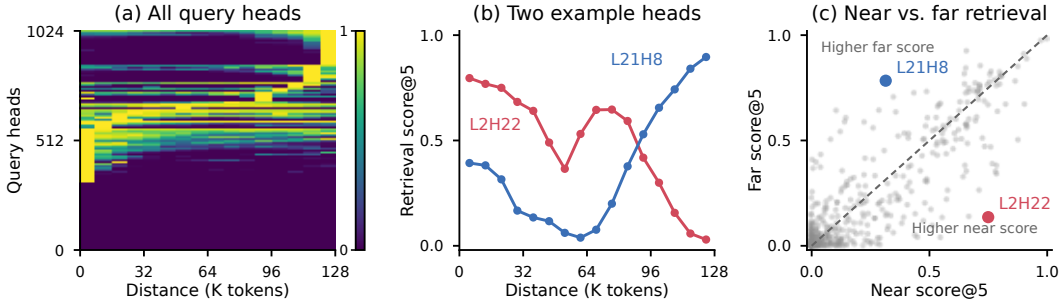


Figure 1: **Retrieval capability varies across attention heads and distances.** Results are shown for Llama-3.1-8B-Instruct. **(a)** Per-head-normalized retrieval score@5 for all query heads, sorted by peak distance. **(b)** Raw retrieval score@5 profiles of two example heads. **(c)** Mean retrieval score@5 at near and far distances for every query head; the diagonal indicates equal scores.

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

Figure 1 reveals a limitation of this head-level characterization: an attention head's retrieval capability can vary markedly across relative distances (see Section 3.1 for further analysis). Consequently, head-level allocation cannot distinguish between distance ranges where the same head exhibits strong and weak retrieval capability. As a result, retaining an entire head may consume cache budget at distances where it contributes little, whereas compressing it may discard useful states at other distances. This within-head variation suggests that relative distance is not merely a positional coordinate but a key dimension for characterizing long-context retrieval.

Building on this observation, we propose Distance-KV, which learns a static KV retention pattern over the joint space of layers, query heads, and relative distances. This finer granularity allows the same query head to retain KV states over useful distance ranges while compressing the rest. For each model and cache budget, Distance-KV optimizes the pattern offline using synthetic retrieval examples while keeping the language model frozen. The resulting pattern is independent of the content of both the context and the query and can therefore be reused across inputs and successive queries. At inference time, the learned pattern is mapped to token positions and applied once to prune and compact the KV cache, requiring no online importance scoring or optimization.

We evaluate Distance-KV on controlled retrieval, downstream long-context tasks, robustness across context lengths, and shared-context multi-turn reuse using three backbone models. Across these settings, Distance-KV consistently leads the competing compression methods and improves over the strongest compression baseline by up to 9.3 points on RULER at 128K. The reduction in retained KV states also translates into practical efficiency: on Llama-3.1-8B-Instruct at 128K, Distance-KV uses 65.4% less KV cache memory and achieves a $1.66\times$ decoding speedup over Dense.

Our contributions are summarized as follows:

- We show that an individual query head can exhibit markedly different retrieval behavior at different relative distances, exposing a limitation of treating the entire head as a single unit for cache allocation.
- We introduce Distance-KV, a model- and budget-specific static pattern that determines KV retention jointly over layers, query heads, and relative distances, enabling physical KV cache compression without online scoring.
- We evaluate Distance-KV across three backbone models and four complementary long-context benchmarks. Structural ablations confirm the importance of the learned query-head–distance structure, while pattern analyses reveal model-specific distance-selective retention.

2 Related Work

Efficient long-context inference. Long-context inference has been accelerated at the kernel, memory-management, and execution levels. FlashAttention (Dao et al., 2022) reduces memory traffic in exact attention, while PagedAttention (Kwon et al., 2023) manages KV caches through noncontiguous memory blocks. MInference (Jiang et al., 2024) and FlexPrefill (Lai et al., 2025) exploit attention sparsity to reduce the cost of long-context prefilling. Recurrent and segmented execution methods further improve scalability by carrying bounded states across segments (Dai et al., 2019; Shang et al., 2026).

KV cache compression. KV cache compression reduces storage and decoding costs by retaining only a subset of KV states. StreamingLLM (Xiao et al., 2024) preserves fixed sink and recent tokens, while H_2O (Zhang et al., 2023), SnapKV (Li et al., 2024), and DefensiveKV (Feng et al., 2026) use attention-derived scores to retain important tokens. PyramidKV (Cai et al., 2025) varies the cache budget across layers, whereas KVzip (Kim et al., 2025) derives query-agnostic KV importance through context reconstruction.

Attention-head heterogeneity and retrieval behavior. Attention heads exhibit substantial functional heterogeneity. Work on retrieval heads (Wu et al., 2025) identifies a small subset of heads responsible for long-range retrieval, while FastGen (Ge et al., 2024) assigns cache compression policies according to head behavior. DuoAttention (Xiao et al., 2025) separates retrieval heads from streaming heads, and MoA (Fu et al., 2024) searches for sparse attention rules specific to individual layers and heads. Despite their different formulations, these methods assign each head an overall role

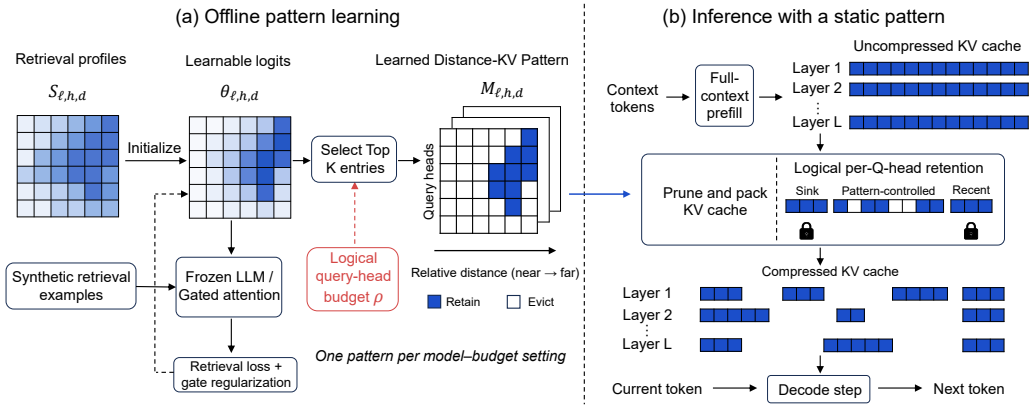


Figure 2: Overview of Distance-KV. (a) Retrieval-guided offline learning produces a model- and budget-specific binary pattern with one query-head-by-distance matrix per layer while keeping the language model frozen. (b) After full-context prefill, the pattern is mapped to the current context and applied once to prune and pack its KV cache; the same pattern is reused across inputs.

or parameterized sparse rule rather than directly modeling its retrieval utility across relative-distance intervals. Distance-KV instead models this missing structure through a model- and budget-specific static retention pattern over layers, query heads, and relative distances.

3 Distance-KV

Existing retrieval-head methods for KV cache compression typically characterize each attention head using a single importance score or binary label and make cache allocation decisions at the granularity of entire heads. However, Figure 1 shows that a head's retrieval capability can vary with relative distance, so head-level allocation can conflate useful and redundant distance ranges. Motivated by this observation, we propose Distance-KV, which learns a fine-grained static KV retention policy over layers, query heads, and relative distances. Figure 2 summarizes how the model- and budget-specific pattern is learned offline and reused to prune and pack the KV cache at inference time. We first quantify the distance-dependent retrieval behavior, then present the pattern representation, its budget-constrained learning procedure, and static-pattern inference.

3.1 Retrieval Capability Varies with Distance

To quantify how the retrieval capability of attention heads varies with distance, we probe every query head in a frozen language model using controlled synthetic retrieval examples (see Appendix A.1 for experimental details). In each example, we place the target value at a different relative distance from the query and record the attention distributions of the query heads as the model predicts the tokens of that value. Adapting the retrieval score of Wu et al. (2025), we define the distance-conditioned retrieval score@5 for query head h in layer ℓ and distance bin d, denoted by $S_{\ell,h,d}$, as the proportion of target tokens whose corresponding source positions appear among the five positions receiving the highest attention from that head.

Figure 1 provides an overall view of how retrieval capability varies with distance across attention heads. In Figure 1(a), we divide each head's scores by its maximum across distance bins and sort the 1,024 query heads in Llama-3.1-8B-Instruct by the bin in which this maximum occurs. The resulting high-score regions span different distance ranges, revealing distinct distance-dependent retrieval behavior across heads. Since this normalization is used only to highlight within-head variation, Figure 1(b) further reports the raw retrieval scores of two example heads with contrasting profiles. L2H22 performs better at shorter distances, whereas L21H8 becomes stronger at longer distances. Figure 1(c) extends this comparison to all query heads by plotting their mean raw retrieval scores over the four farthest distance bins against those over the four nearest bins. Points on both sides of the diagonal show that this contrast extends beyond the two highlighted heads.

The results reveal substantial variation with relative distance across the layer–query-head space. A single head score cannot distinguish the distance ranges in which a head is useful, and the observed profiles are not reducible to a simple near–far dichotomy: some heads peak at intermediate distances or vary nonmonotonically. We therefore formulate KV allocation over individual layer–query-head–distance entries using a Distance-KV Pattern M, formally defined in Section 3.2.

3.2 DISTANCE-KV PATTERN

To capture the distance-dependent retrieval variation identified in Section 3.1, we partition the compressible portion of the context into D relative-distance bins and define a separate retention decision for every layer, query head, and distance bin. For a model with L layers and H query heads per layer, the Distance-KV Pattern is a binary tensor $M \in \{0,1\}^{L\times H\times D}$. Each entry $M_{\ell,h,d} = 1$ indicates that the context positions in distance bin d are logically retained for query head h in layer ℓ, whereas $M_{\ell,h,d} = 0$ indicates that they are not. As illustrated in Figure 2(a), each layer corresponds to a binary query-head-by-distance matrix, and the collection of these matrices forms the complete Distance-KV Pattern. Here, M defines a logical retention policy over query heads. For GQA models, its mapping to shared physical KV heads is described in Section 3.4.

For each model and retention setting, the Distance-KV Pattern is learned once in relative-distance coordinates and reused across inputs. A distance-agnostic head-level policy is recovered by setting $M_{\ell,h,d} = z_{\ell,h}$ for all d. More generally, each query head can retain arbitrary, potentially disjoint distance regions, rather than being restricted to a single sliding window. Sink and recent tokens are always retained, while $M_{\ell,h,d}$ controls the compressible context between them.

3.3 BUDGET-CONSTRAINED PATTERN LEARNING

As illustrated in Figure 2(a), given a model and a target logical query-head retention ratio ρ, Distance-KV learns the corresponding retention pattern offline from synthetic retrieval examples. During training, all parameters of the pretrained language model remain frozen, and only the gate logits $\theta_{\ell,h,d}$ associated with the layer–query-head–distance entries are optimized. We define $\mathcal{L}_{\mathrm{ret}}(M)$ as the teacher-forced cross-entropy on the target-value tokens of the synthetic retrieval examples when the frozen model uses pattern M. We initialize each gate probability from the corresponding distance-conditioned retrieval score $S_{\ell,h,d}$. Let $K_\rho = \mathrm{round}(\rho LHD)$ denote the number of active entries under the target retention ratio. We formulate the desired binary pattern as

$$\begin{aligned} M^\star_\rho = \arg\min_{M} \quad & \mathcal{L}_{\mathrm{ret}}(M) \\ \text{s.t.} \quad & M \in \{0,1\}^{L\times H\times D}, \quad \|M\|_0 = K_\rho. \end{aligned} \tag{1}$$

To optimize this discrete objective, we first use Hard Concrete gates (Louizos et al., 2018) to learn a continuous relaxation of the retention variables and then refine an exact-budget binary pattern through straight-through Top-K optimization.

In the first stage, we relax the discrete retention decisions using Hard Concrete gates. Each gate logit $\theta_{\ell,h,d}$ parameterizes a logical query-head retention variable $Z_{\ell,h,d}$. During the forward pass, the resulting relaxed gates modulate the attention mass over the context positions associated with each distance bin. Let $p_{\ell,h,d} = \Pr(Z_{\ell,h,d} > 0)$ denote the probability that a pattern entry is active. We optimize

$$\mathcal{L}_{\mathrm{HC}} = \mathcal{L}_{\mathrm{ret}}(Z) + \frac{\lambda}{LHD} \sum_{\ell,h,d} p_{\ell,h,d}. \tag{2}$$

The first term preserves retrieval performance on the synthetic examples, while the second is an expected L_0 regularizer that promotes sparsity. This stage learns differentiable retention probabilities without imposing the exact target ratio. Additional details of the initialization and relaxed gates are provided in Appendix A.2.

In the second stage, we initialize from the relaxed solution. Let $P(\theta) \in [0,1]^{L\times H\times D}$ collect the nonzero probabilities, with $P_{\ell,h,d}(\theta) = p_{\ell,h,d}$. We construct an exact-budget binary pattern using global Top-K selection:

$$M_\rho(\theta) = \mathrm{TopKMask}\left(P(\theta), K_\rho\right), \tag{3}$$

where $\mathrm{TopKMask}(\cdot, K_\rho)$ sets the K_ρ largest entries to one and all remaining entries to zero. Because the Top-K operator is nondifferentiable, we use a straight-through estimator (Bengio et al., 2013): the forward pass follows this exact-budget binary pattern, while gradients are propagated to the gate logits through the relaxed Hard Concrete attention path. After optimization, we apply the same global Top-K selection and export the resulting deterministic Distance-KV Pattern. Details of the straight-through estimator and global Top-K selection are provided in Appendix A.3.

3.4 Static-Pattern Inference

At inference time, given a context of length T_c, we map the relative-distance bins in the learned Distance-KV Pattern to the corresponding context positions, using the end of the context as the reference point. For a context token at position $i \in \{129, \ldots, T_c - 1024\}$, its zero-indexed relative-distance bin is

$$d(i; T_c) = \left\lfloor \frac{T_c - i - 1024}{128} \right\rfloor. \tag{4}$$

The logical retention decision for query head h in layer ℓ at position i is then

$$m_{\ell,h,i} = \begin{cases} 1, & i \in \mathcal{S}(T_c) \cup \mathcal{R}(T_c), \\ M_{\ell,h,d(i;T_c)}, & \text{otherwise.} \end{cases} \tag{5}$$

Here, $\mathcal{S}(T_c)$ and $\mathcal{R}(T_c)$ denote the first 128 sink positions and the most recent 1,024 positions, respectively; both are always retained. The learned pattern controls only the compressible context between these two regions. Because the mapping uses relative rather than absolute positions, only the bins covered by the current context are instantiated, allowing the same pattern to support different context lengths.

As illustrated in Figure 2(b), after dense prefill over the context, we apply the retention decisions defined above to perform a one-time static compression of the context KV cache. The KV states marked for eviction are physically removed, while the remaining states are packed in their original token order. The packing operation changes only the storage layout of the retained KV states and preserves their original position indices and RoPE semantics. The query tokens then attend to the compressed context cache, while the KV states of the query tokens and subsequently generated tokens are appended normally and retained in full. Distance-KV neither recomputes nor reapplies the pattern during generation and therefore requires no query-dependent importance scoring or online pattern optimization.

For MHA, the logical retention decision of each query head applies directly to its corresponding physical KV head. For GQA, the decisions of query heads sharing the same KV head are combined by taking their union, so a position is retained whenever any query head in the group selects it. Consequently, ρ specifies the logical query-head retention ratio, whereas cache usage and attention cost are governed by the physical KV retention ratio after this union.

4 Experiments

4.1 Experimental Setup

Models and benchmarks. We evaluate Distance-KV on four complementary long-context benchmarks using three backbone models: Llama-2-7B-32K-Instruct (Touvron et al., 2023), Llama-3.1-8B-Instruct (Llama Team, 2024), and Qwen2.5-7B-Instruct (Yang et al., 2024). Needle-in-a-Haystack (NIAH) varies both context length and needle position to assess position-wise retrieval robustness. LongBench (Bai et al., 2024) covers diverse downstream tasks, RULER (Hsieh et al., 2024) measures robustness across context lengths through controlled tasks, and SCBench (Li et al., 2025) evaluates KV cache reuse in shared-context, multi-turn settings.

Baselines and evaluation protocol. We compare Distance-KV with the full-cache Dense baseline and four KV cache compression methods: StreamingLLM (Xiao et al., 2024), DuoAttention (Xiao et al., 2025), MoA (Fu et al., 2024), and the context-independent variant of KVzip (Kim et al., 2025), denoted as KVzip-CI. All compression methods determine which states to retain from the initial context KV cache before processing the query, without access to the query content. In SCBench,

the compressed shared-context KV cache is reused across subsequent queries. We follow the official prompt construction, generation, and evaluation protocols of LongBench, RULER, and SCBench.

Implementation details. Distance-KV uses a logical Q-head retention ratio of 20% in all main experiments. For GQA models, the resulting physical KV usage is measured after unioning the decisions of query heads that share a KV head, as described in Section 3.4, and is reported in Section 4.3 and Appendix D. We learn one model-specific pattern for each backbone and reuse it unchanged across all benchmarks evaluated with that model. The packed KV-cache update kernel used during inference is adapted from DefensiveKV (Feng et al., 2026). All baselines are evaluated using their publicly released implementations with fixed model-specific configurations across benchmarks. Additional pattern-learning details are provided in Appendix A, and baseline configurations are provided in Appendix B. Our code is available at link.

4.2 Long-Context Performance

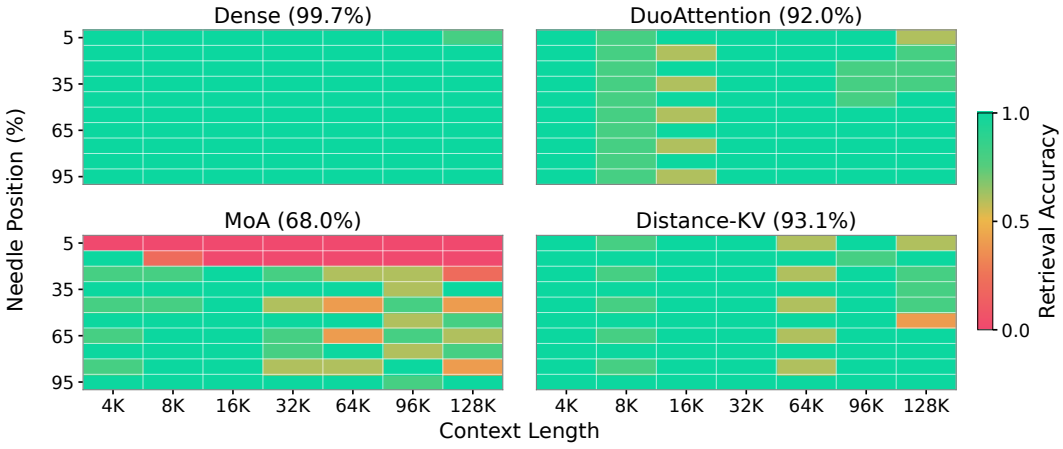


Figure 3: Retrieval accuracy of Llama-3.1-8B-Instruct on multi-needle NIAH across context lengths and needle positions. Values in parentheses report the mean accuracy over the full evaluation grid.

Position-wise retrieval robustness. Figure 3 shows that Distance-KV preserves high retrieval accuracy over nearly the entire length–position grid, including 128K contexts. MoA exhibits broad low-accuracy regions, whereas DuoAttention remains competitive but shows more localized degradation. A single static pattern therefore supports retrieval across widely varying relative distances.

Downstream long-context performance. As shown in Table 1, Distance-KV achieves the highest average LongBench score among all KV cache compression methods on each of the three backbones. It is also the best-performing compression method on multi-document question answering for all three models, a category that requires evidence integration across documents. The patterns learned from synthetic retrieval therefore transfer to natural long-document tasks and remain effective on both MHA and GQA backbones. Complete task-level results are reported in Appendix C.1.

Robustness across context lengths. Figure 4 shows that the separation among methods is clearest at long context lengths. Averaged over the eight task groups at 128K, Distance-KV outperforms the strongest compression baseline by 5.7 points on Llama-3.1-8B-Instruct and 9.3 points on Qwen2.5-7B-Instruct. The gains extend beyond the four NIAH variants to variable tracking, aggregation, and question answering, indicating that the benefit of distance-conditioned retention is not limited to exact retrieval. Complete task-level results at 128K are reported in Appendix C.2.

Shared-context multi-turn performance. Distance-KV achieves the highest average among compression methods on both backbones (Table 2), ranking first on 7 of 12 Llama-3.1 tasks and 10 of 12 Qwen2.5 tasks, with wins in every capability category. A single fixed pattern therefore remains effective across successive queries without query-dependent rescoring.

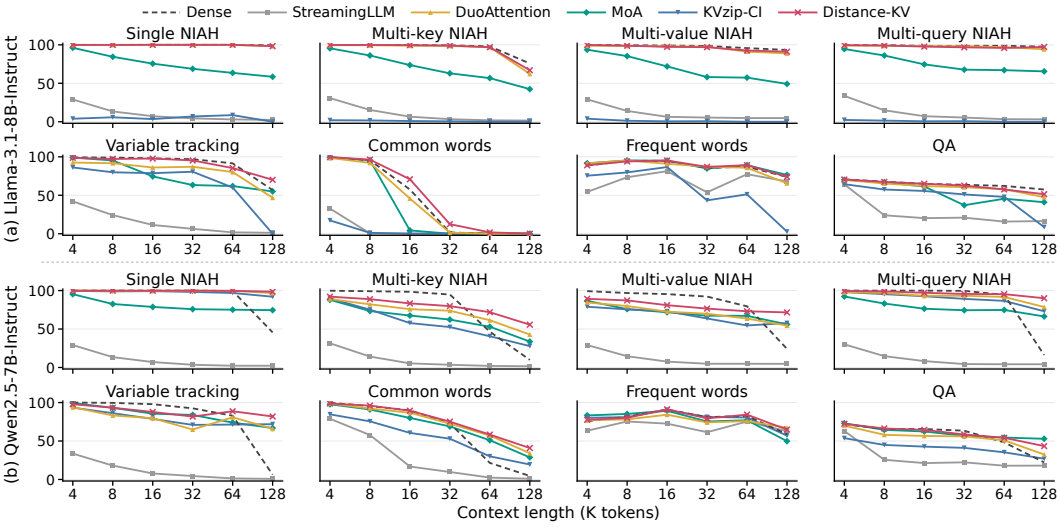


Figure 4: **RULER task-group scores across context lengths.** Results are shown for (a) Llama-3.1-8B-Instruct and (b) Qwen2.5-7B-Instruct from 4K to 128K tokens. For task groups containing multiple tasks, we report the mean score within each group.

Table 1: LongBench performance across three backbone models. Scores are averaged within each of the six official task categories, and Avg. denotes the overall average across all 21 datasets. Bold indicates the best KV cache compression method in each column.

Model	Method	LongBench						
		S-Doc	M-Doc	Summ.	Few.	Synth.	Code	Avg.
Llama-2-7B-32K-Instruct	Dense	12.32	12.59	15.18	56.90	7.19	62.64	25.47
	StreamingLLM	8.42	9.12	13.51	36.13	3.11	57.71	18.74
	DuoAttention	**13.82**	11.57	14.74	48.24	5.61	54.48	22.82
	MoA	11.77	10.90	13.75	**55.86**	**9.16**	61.21	24.72
	KVzip-CI	8.85	9.15	11.73	30.47	2.25	50.36	16.59
	Distance-KV	12.58	**12.63**	**14.76**	54.34	8.70	**61.56**	**25.07**
Llama-3.1-8B-Instruct	Dense	48.58	41.92	17.75	61.53	66.34	60.01	47.53
	StreamingLLM	23.96	24.26	15.88	41.60	14.00	60.42	27.89
	DuoAttention	45.77	39.48	17.51	**60.53**	**67.25**	**65.25**	46.92
	MoA	43.50	35.51	17.21	59.64	57.39	62.76	43.86
	KVzip-CI	43.49	31.70	14.60	55.03	53.98	19.66	37.17
	Distance-KV	**48.39**	**41.08**	**17.76**	60.44	66.72	61.20	**47.30**
Qwen2.5-7B-Instruct	Dense	45.97	39.09	16.42	56.86	62.17	60.36	44.79
	StreamingLLM	23.07	22.70	15.48	44.64	13.00	55.24	27.29
	DuoAttention	43.36	36.11	**16.85**	**56.20**	58.83	61.19	43.28
	MoA	40.41	34.27	16.48	56.06	56.50	**62.25**	42.04
	KVzip-CI	33.44	29.11	14.36	50.16	29.72	44.40	32.68
	Distance-KV	**43.91**	**37.90**	16.46	55.68	**62.17**	60.85	**44.00**

4.3 Efficiency

Inference efficiency. Figure 5 compares the physical KV cache size and per-token decoding latency as the context length increases. We omit MoA because its optimized kernel operates on an MHA-converted checkpoint rather than the native GQA backbone. At 128K, Distance-KV uses 5.41 GiB for the context KV cache, 65.4% less than Dense and also less than DuoAttention and KVzip-CI. Its decoding latency stays close to 18 ms/token from 4K through 64K and rises to only 19.88 ms/token at 128K, the lowest at this length among the compared methods and a 1.66× speedup over

Table 2: SCBench multi-turn performance on two backbone models across 12 tasks grouped into four capability categories. All scores are reported on a 0–100 scale, with higher values indicating better performance. Avg. denotes the unweighted mean of the four category averages, and bold indicates the best KV cache compression method in each column.

Method	String retrieval			Semantic retrieval				Global			Multi-task		Avg.
	KV	P–S	VT	Repo	En.QA	Zh.QA	Choice	ICL	Summ.	Math	S+N	R+KV	
Llama-3.1-8B-Instruct													
Dense	80.4	41.6	47.6	50.5	29.6	20.5	69.4	43.3	40.4	21.8	61.2	68.9	49.8
StreamingLLM	0.2	0.2	0.0	0.2	5.7	6.4	26.2	**72.2**	25.3	5.4	30.3	0.2	14.8
DuoAttention	**58.4**	19.6	**43.0**	**42.1**	25.9	16.9	65.1	58.9	35.8	9.4	49.7	**45.5**	40.0
MoA	26.2	13.0	11.4	20.5	16.2	15.6	66.8	61.9	31.4	9.8	42.5	14.8	27.4
KVzip-CI	0.0	0.0	1.9	7.7	5.1	1.8	18.8	57.8	19.3	0.2	20.4	4.6	11.8
Distance-KV	56.8	**20.8**	28.8	38.6	**27.1**	**20.2**	**70.3**	59.6	**36.6**	**20.4**	**58.2**	43.2	**41.0**
Qwen2.5-7B-Instruct													
Dense	12.4	26.8	36.8	44.5	23.1	4.1	63.3	57.8	33.7	23.2	65.6	44.1	38.0
StreamingLLM	0.2	0.0	0.0	0.2	6.3	1.8	24.5	70.7	27.5	5.2	37.7	0.0	15.4
DuoAttention	2.8	4.2	18.5	24.5	19.9	3.1	55.0	**71.9**	29.4	16.2	63.2	18.9	28.6
MoA	6.6	4.8	8.9	25.7	18.8	2.8	55.9	71.5	29.3	10.6	51.9	14.1	25.7
KVzip-CI	1.2	0.0	24.6	12.3	16.5	**3.3**	52.0	58.5	25.5	5.8	45.6	9.1	21.7
Distance-KV	**15.0**	**13.2**	**30.0**	**37.5**	**24.5**	2.6	**62.0**	64.8	**32.3**	**22.6**	**64.7**	**27.3**	**34.2**

Task abbreviations: KV, key–value retrieval; P–S, prefix–suffix matching; VT, variable tracking; Repo, repository QA; S+N, summarization with NIAH; R+KV, repository QA with KV retrieval.

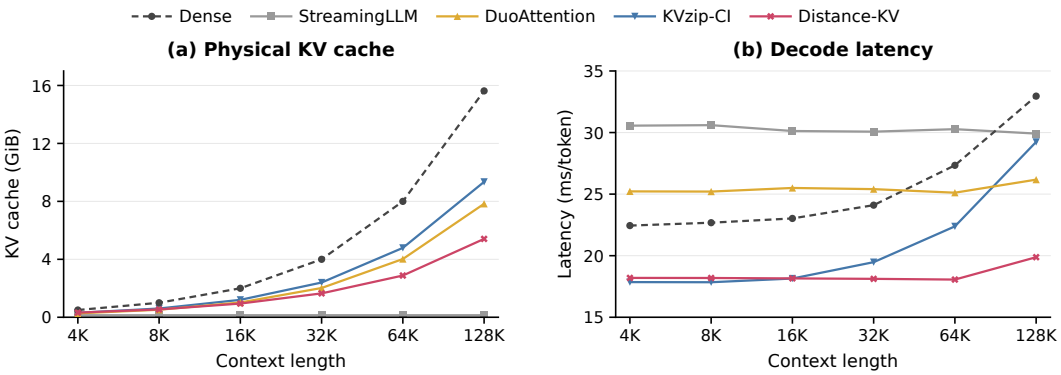


Figure 5: **Inference efficiency across context lengths.** (a) Physical KV cache size and (b) per-token decoding latency for Llama-3.1-8B-Instruct on one NVIDIA H100 GPU with batch size 1.

Dense. The compacted cache therefore reduces both storage and decoding latency as the context grows. Appendix D reports results for Qwen2.5-7B-Instruct and peak GPU memory.

4.4 Ablation Studies

We conduct structural ablations on RULER at 128K with matched physical KV retention across variants. As shown in Table 3, Distance-KV performs best across all eight task groups, averaging 63.7. Head-only applies one retention decision to all distance bins within each head and drops by 19.2 points, while Distance-only shares distance-conditioned decisions across query heads within each layer and drops by 51.0 points. Neither whole-head retention nor head-independent distance allocation is therefore sufficient. Distance-shuffled preserves per-head retention counts and physical KV cost but permutes distance-bin assignments, causing a 9.0-point drop. This isolates the learned

Table 3: Ablation results on RULER at 128K for Llama-3.1-8B-Instruct. Single, M-Key, and QA average three, three, and two constituent tasks, respectively; Avg. is the mean over all 13 tasks. Bold indicates the best result in each column. All variants are evaluated at the same training step.

Variant	NIAH				VT	CWE	FWE	QA	Avg.
	Single	M-Key	M-Value	M-Query					
Head-only	73.8	25.5	84.8	87.2	2.6	0.0	59.9	22.9	44.5
Distance-only	10.0	4.4	16.3	13.8	3.6	0.1	55.8	16.0	12.7
Distance-shuffled	84.7	36.9	90.1	92.3	33.6	0.1	64.5	32.8	54.7
Distance-KV	**95.9**	**45.5**	**93.2**	**95.5**	**61.4**	**0.2**	**70.3**	**41.8**	**63.7**

layer–query-head–distance correspondence as key to Distance-KV's performance. Appendix E provides complete task-level results, including the fully shuffled control.

4.5 Analysis of Learned Patterns

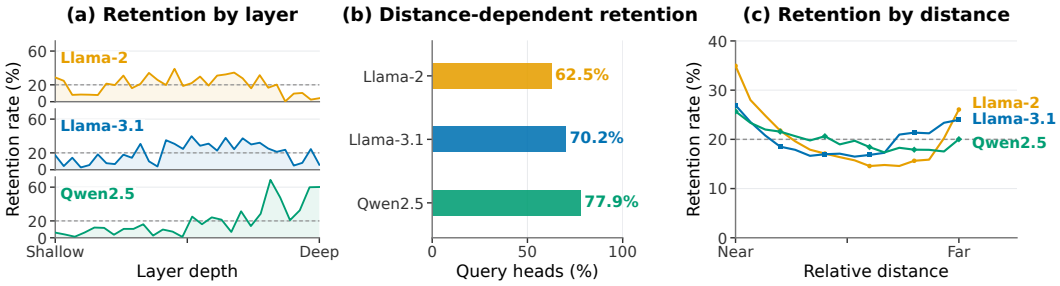


Figure 6: **Learned Distance-KV patterns across three backbone models.** All patterns use a 20% logical query-head retention ratio. (a) Mean retention rate by layer. (b) Fraction of query heads that retain only a subset of relative-distance bins. (c) Mean retention rate by relative distance. Gray dashed lines in (a) and (c) indicate the 20% retention ratio.

Despite sharing the same overall retention ratio of 20%, the three models learn distinct, nonuniform layer-wise allocations (Figure 6(a)). Specifically, across all layers, 62.5%, 70.2%, and 77.9% of query heads in Llama-2, Llama-3.1, and Qwen2.5, respectively, retain only a subset of distance bins, showing that most heads use distance-dependent retention rather than a single decision across all distances (Figure 6(b)). Retention is also distance-dependent: all three models favor nearby positions; Llama-2 and Llama-3.1 increase retention again at long distances, whereas Qwen2.5 has a flatter profile (Figure 6(c)). Together with Figure 1, these patterns establish relative distance as a distinct dimension of retrieval behavior and KV allocation.

5 Conclusion

We show that retrieval capability varies substantially with relative distance, even within the same query head. Building on this observation, we introduce Distance-KV, which uses a reusable static retention pattern defined jointly over layers, query heads, and relative distances. Experiments across multiple backbone models show that Distance-KV consistently achieves the best overall performance among the evaluated KV cache compression methods while reducing physical KV cache size and decoding latency. Ablations and learned-pattern analyses further establish the importance of the correspondence between query heads and distance ranges. More broadly, our findings identify relative distance as an important structural dimension for understanding long-context retrieval and motivate further study.

AI USE STATEMENT

Generative AI tools were used to assist with implementing portions of the research code, drafting parts of the manuscript, and polishing the language for clarity, concision, and readability. All AI-assisted code was reviewed and tested by the authors, and all AI-assisted text was reviewed and revised by the authors. The authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, Section 3 describes Distance-KV, Section 4.1 specifies the experimental protocols and implementation settings, and Appendices A–E provide additional implementation details and experimental results. The anonymized implementation is linked in Section 4.1.

REFERENCES

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013. URL https://arxiv.org/abs/1308.3432.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=ayi7qezU87.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285. URL https://aclanthology.org/P19-1285/.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=H4DqfPSibmx.

Yuan Feng, Haoyu Guo, Junlin Lv, S. Kevin Zhou, and Xike Xie. DefensiveKV: Taming the fragility of KV cache eviction in LLM inference. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=nJgS06sX3O.

Tianyu Fu, Haofeng Huang, Xuefei Ning, Genghan Zhang, Boju Chen, Tianqi Wu, Hongyi Wang, Zixiao Huang, Shiyao Li, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. MoA: Mixture of sparse attention for automatic large language model compression. *CoRR*, abs/2406.14909, 2024. doi: 10.48550/ARXIV.2406.14909. URL https://doi.org/10.48550/arXiv.2406.14909.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uNrFpDPMyo.

Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

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 D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 167563–167591. Curran Associates, Inc., 2025. doi: 10.52202/085713-5585. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f4eaa4b8f2d08edb3f0af990d56134ea-Paper-Conference.pdf.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pp. 611–626, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702297. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. FlexPrefill: A context-aware sparse attention mechanism for efficient long-sequence inference. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=OfjIlbelrT.

Yucheng Li, Huiqiang Jiang, Qianhui Wu, Xufang Luo, Surin Ahn, Chengruidong Zhang, Amir H. Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, and Lili Qiu. SCBench: A KV cache-centric analysis of long-context methods. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=gkUyYcY1W9.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Llama Team. The Llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through l_0 regularization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1Y8hhg0b.

Xianpeng Shang, Jiang Li, Zehua Duo, Qianyi Cai, and Xiangdong Su. Training–inference consistent segmented execution for long-context LLMs. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=PoRigyDOcC.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan

Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and Fine-Tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/ARXIV.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=EytBpUGB1Z.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. DuoAttention: Efficient long-context LLM inference with retrieval and streaming heads. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=cFu7ze7xUm.

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, 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. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, Weijia Xu, Wenbiao Yin, Wenyuan Yu, Xiafei Qiu, Xingzhang Ren, Xinlong Yang, Yong Li, Zhiying Xu, and Zipeng Zhang. Qwen2.5-1M technical report, 2025. URL https://arxiv.org/abs/2501.15383.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2O: Heavy-Hitter oracle for efficient generative inference of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=RkRrPp7GKO.

A Pattern-Learning Details

For pattern learning, we reserve 256 tokens of each model's configured sequence-length budget for the query and answer, leaving 32,512 context tokens for Llama-2 and 130,816 for Llama-3.1 and Qwen2.5. We reserve the first 128 context tokens as sink tokens and the most recent 1,024 context tokens as recent tokens, and partition the remaining compressible context into 128-token relative-distance bins. This yields $D = 245$ for Llama-2 and $D = 1{,}013$ for Llama-3.1 and Qwen2.5.

A.1 Retrieval-Profile Measurement

Figure 1 uses retrieval scores collected for retrieval-guided pattern initialization with Llama-3.1-8B-Instruct. We insert four key–value records at controlled relative distances into a context of unrelated filler text. Each context uses either access codes consisting of three three-digit groups or verification phrases consisting of three words. Each record is queried separately. Representative records from the two value-type conditions are:

> *Numerical:* The access code associated with record key "harbor" is "470 668 610".
> *Word:* The verification phrase associated with record key "harbor" is "antique amber armchair".

We use two single-key query templates:

> *Q1:* The document contains record-key-to-{record_kind} entries. Retrieve the {item} associated with record key "{key}".
> *Q3:* According to the {record_kind} records in the document, what {item} corresponds to record key "{key}"?

Here, {record_kind} is *access-code* or *verification-phrase*, and {item} is the corresponding value name. Both templates additionally specify the value format and instruct the model to output only that value.

The probe uses four distance-coverage rounds. For each distance bin, we average repeated examples within each value-type–template–round condition, then average the condition means equally. For visualization, we aggregate the 1,013 original 128-token distance bins into 16 coarser bins. The near and far scores in Figure 1(c) average the first and last four of these bins, respectively.

A.2 Retrieval-Guided Initialization and Relaxed Gates

Distance-KV associates each layer–query-head–distance entry with a trainable gate logit $\theta_{\ell,h,d}$. Let $\bar{S}$ denote the mean of the distance-conditioned retrieval scores over all entries. We map each score to an initial nonzero probability as

$$\pi^{(0)}_{\ell,h,d} = \mathrm{clip}_{[p_{\min},p_{\max}]}\left[\sigma\left(\mathrm{logit}(\mu) + a\left(S_{\ell,h,d} - \bar{S}\right)\right)\right], \tag{A.1}$$

where μ specifies the central retention probability, a controls the dependence on the retrieval score, and $p_{\min}$ and $p_{\max}$ bound the resulting probabilities.

We parameterize each relaxed gate $Z_{\ell,h,d}$ using the Hard Concrete distribution (Louizos et al., 2018), with temperature β and stretch bounds γ and ζ. Its nonzero probability is

$$p_{\ell,h,d} = \Pr\left(Z_{\ell,h,d} > 0\right) = \sigma\left(\theta_{\ell,h,d} - \beta \log \frac{-\gamma}{\zeta}\right). \tag{A.2}$$

We initialize the gate logits by inverting this relation:

$$\theta^{(0)}_{\ell,h,d} = \mathrm{logit}\left(\pi^{(0)}_{\ell,h,d}\right) + \beta \log \frac{-\gamma}{\zeta}. \tag{A.3}$$

For GQA models, the relaxed gates remain defined separately for query heads that share the same KV head and independently modulate their corresponding attention distributions during pattern learning. The expected L_0 regularizer is likewise defined over these logical query-head gates. The binary retention decisions are combined by union only when the learned logical pattern is mapped to shared physical KV heads at inference, as described in Section 3.4.

A.3 Exact-Budget Refinement

Let $P(\theta) \in [0,1]^{L\times H\times D}$ denote the tensor of nonzero probabilities, with $P_{\ell,h,d}(\theta) = p_{\ell,h,d}$. For a target logical retention ratio ρ, we construct an exact-budget binary pattern as

$$K_\rho = \mathrm{round}\left(\rho LHD\right), \qquad M_\rho(\theta) = \mathrm{TopKMask}\left(P(\theta), K_\rho\right), \tag{A.4}$$

where the Top-K selection is performed globally over all layer–query-head–distance entries.

Let A^{rel} and A^{bin} denote the normalized attention probabilities produced by the relaxed gates and the binary pattern, respectively. We define the straight-through attention probabilities as

$$A^{\mathrm{ST}} = A^{\mathrm{rel}} + \mathrm{stopgrad}\left(A^{\mathrm{bin}} - A^{\mathrm{rel}}\right). \tag{A.5}$$

Consequently, the forward pass follows the binary pattern, while the gradients with respect to the gate logits follow the relaxed attention path:

$$A^{\mathrm{ST}} \overset{\text{forward}}{=} A^{\mathrm{bin}}, \qquad \frac{\partial A^{\mathrm{ST}}}{\partial \theta_{\ell,h,d}} = \frac{\partial A^{\mathrm{rel}}}{\partial \theta_{\ell,h,d}}. \tag{A.6}$$

The straight-through estimator is applied to the normalized attention probabilities rather than directly to the gate tensor. The refinement objective is

$$\mathcal{L}_{\mathrm{ST}} = \mathcal{L}_{\mathrm{ret}}\left(A^{\mathrm{ST}}\right) + \frac{\lambda}{LHD}\sum_{\ell,h,d} p_{\ell,h,d}. \tag{A.7}$$

After optimization, we apply the same global Top-K selection to the learned nonzero probabilities and export the resulting binary pattern for inference.

Table 4: Shared hyperparameters for offline pattern learning.

Hyperparameter	Value
Hard Concrete temperature β	$2/3$
Stretch bounds (γ, ζ)	$(-0.1, 1.1)$
Initialization center μ	0.7
Initialization slope a	6
Probability bounds $[p_{\min}, p_{\max}]$	$[0.001, 0.999]$
Optimizer	AdamW
Learning rate	0.01
AdamW weight decay	0
Gradient clipping norm	1.0
Final L_0 coefficient λ	0.5
L_0 ramp steps	4

Hyperparameters. Table 4 lists the shared hyperparameters for pattern learning.

B Baseline Configurations

All baselines use their publicly released implementations with fixed model-specific configurations across benchmarks. When an author-provided retention pattern or sparse attention configuration is available for a backbone, we use it directly; otherwise, we derive a model-specific configuration using the official implementation and search procedure. Dense retains the full KV cache. StreamingLLM retains 128 sink tokens and 1,024 recent tokens. DuoAttention retains 64 sink tokens and 256 recent tokens, with a sparsity ratio of 0.75 for Llama-2-7B-32K-Instruct and 0.50 for the other two backbones. MoA uses a fixed model-specific sparse attention configuration with a density of 0.50. For KVzip, we use its context-independent variant, denoted as KVzip-CI, with a head-level pruning ratio of 0.60.

C Additional Benchmark Results

C.1 Task-Level LongBench Results

Tables 5–7 report the complete results on all 21 LongBench tasks. These task-level results are aggregated into the six category scores reported in Table 1.

C.2 Task-Level RULER Results at 128K

Tables 8 and 9 report the complete results on all 13 RULER tasks at a context length of 128K. The group averages correspond to the eight task groups reported in Figure 4.

D Additional Efficiency Results

Figure 7(a)–(b) shows that, on Qwen2.5-7B-Instruct, the physical KV cache of Distance-KV grows substantially more slowly with context length than that of Dense, while maintaining the lowest decoding latency among the evaluated methods. At 128K, Distance-KV uses 3.28 GiB of physical KV cache, a 52.0% reduction from Dense, and achieves a decoding latency of 16.96 ms/token, corresponding to a $5.80\times$ speedup. Figure 7(c)–(d) further shows that physical cache compression reduces overall memory usage during decoding. At 128K, Distance-KV uses 20.80 GiB and 17.80 GiB of peak allocated memory on Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct, reducing memory usage relative to Dense by 32.6% and 22.0%, respectively. These results demonstrate consistent reductions in KV cache size, decoding latency, and runtime memory across backbone models.

Table 5: Task-level LongBench results on Llama-2-7B-32K-Instruct. Avg. is the unweighted mean over all 21 tasks. Bold indicates the best KV cache compression method in each row.

Task	Dense	Streaming LLM	Duo Attention	MoA	KVzip-CI	Distance KV
Single-document QA						
NarrativeQA	4.89	4.14	5.00	4.33	**6.79**	3.64
Qasper	12.30	7.22	**15.04**	12.32	6.91	11.59
MultiFieldQA-en	20.97	15.93	**23.07**	20.83	17.00	21.93
MultiFieldQA-zh	11.14	6.38	12.19	9.60	4.69	**13.18**
Multi-document QA						
HotpotQA	13.58	8.21	13.01	**13.30**	10.54	12.74
2WikiMQA	11.40	9.09	10.15	**12.20**	8.05	11.93
MuSiQue	7.08	3.91	5.67	6.84	5.25	**7.50**
DuReader	18.29	15.27	17.47	11.27	12.76	**18.36**
Summarization						
GovReport	19.92	15.98	19.27	19.02	17.29	**19.42**
QMSum	11.20	**11.45**	11.01	10.05	7.97	10.17
MultiNews	16.04	15.08	15.56	15.35	11.49	**16.08**
VCSUM	13.57	11.53	13.12	10.60	10.18	**13.38**
Few-shot learning						
TREC	72.50	46.00	**72.50**	72.00	41.00	70.50
TriviaQA	85.12	50.91	76.63	84.23	49.27	**84.91**
SAMSum	35.97	34.13	32.32	**36.21**	22.20	35.96
LSHT	34.00	13.50	11.50	**31.00**	9.42	26.00
Synthetic tasks						
PassageCount	0.67	0.17	**0.67**	**0.67**	0.10	0.50
PassageRetrieval-en	13.62	5.04	12.42	**22.20**	4.25	16.12
PassageRetrieval-zh	7.29	4.12	3.75	4.62	2.40	**9.47**
Code completion						
LCC	63.62	59.28	**65.44**	61.07	42.87	62.06
RepoBench-P	61.67	56.14	43.52	**61.35**	57.86	61.07
Avg.	25.47	18.74	22.82	24.72	16.59	**25.07**

E Additional Ablation Results

Table 10 reports the complete task-level results for the structural ablations and additionally includes the fully shuffled variant. This variant disrupts all correspondences among layers, query heads, and relative-distance bins while preserving the logical query-head and physical KV retention ratios. Fully shuffled reduces the overall average from 63.72 for Distance-KV to 17.67 and causes substantial degradation on all eight retrieval tasks. Although it achieves a higher score on Frequent words, this isolated improvement does not extend to other tasks or overall performance. Together with the Distance-shuffled results, these findings show that matching the cache size alone is insufficient and that the learned layer–query-head–distance structure is critical to preserving long-context capabilities.

Table 6: Task-level LongBench results on Llama-3.1-8B-Instruct. Avg. is the unweighted mean over all 21 tasks. Bold indicates the best KV cache compression method in each row.

Task	Dense	Streaming LLM	Duo Attention	MoA	KVzip-CI	Distance KV
Single-document QA						
NarrativeQA	29.74	17.81	28.88	28.38	28.87	**31.40**
Qasper	45.01	21.15	41.08	**44.50**	42.62	44.09
MultiFieldQA-en	56.18	30.05	53.64	44.92	48.22	**55.29**
MultiFieldQA-zh	63.39	26.83	59.47	56.21	54.27	**62.77**
Multi-document QA						
HotpotQA	58.01	34.62	57.12	54.27	40.38	**58.67**
2WikiMQA	48.75	26.06	44.16	45.09	38.30	**47.35**
MuSiQue	32.93	11.57	28.20	24.21	22.91	**30.17**
DuReader	28.00	24.78	**28.43**	18.45	25.22	28.12
Summarization						
GovReport	19.85	16.32	19.66	18.97	**20.16**	19.89
QMSum	19.26	15.51	**19.30**	18.26	7.43	19.06
MultiNews	15.46	14.92	14.95	15.24	14.85	**15.54**
VCSUM	16.41	**16.75**	16.14	16.38	15.95	16.55
Few-shot learning						
TREC	73.00	48.50	**72.50**	**72.50**	61.00	72.00
TriviaQA	92.28	68.23	89.92	89.11	**91.33**	91.08
SAMSum	35.36	31.16	34.71	33.45	23.28	**35.16**
LSHT	45.50	18.50	**45.00**	43.50	44.50	43.50
Synthetic tasks						
PassageCount	9.08	7.50	4.50	7.67	4.18	**8.25**
PassageRetrieval-en	99.50	15.00	**99.50**	68.00	97.00	**99.50**
PassageRetrieval-zh	90.45	19.50	**97.75**	96.50	60.77	92.42
Code completion						
LCC	65.17	64.68	**69.15**	65.62	9.99	65.19
RepoBench-P	54.84	56.16	**61.35**	59.90	29.33	57.21
Avg.	47.53	27.89	46.92	43.86	37.17	**47.30**

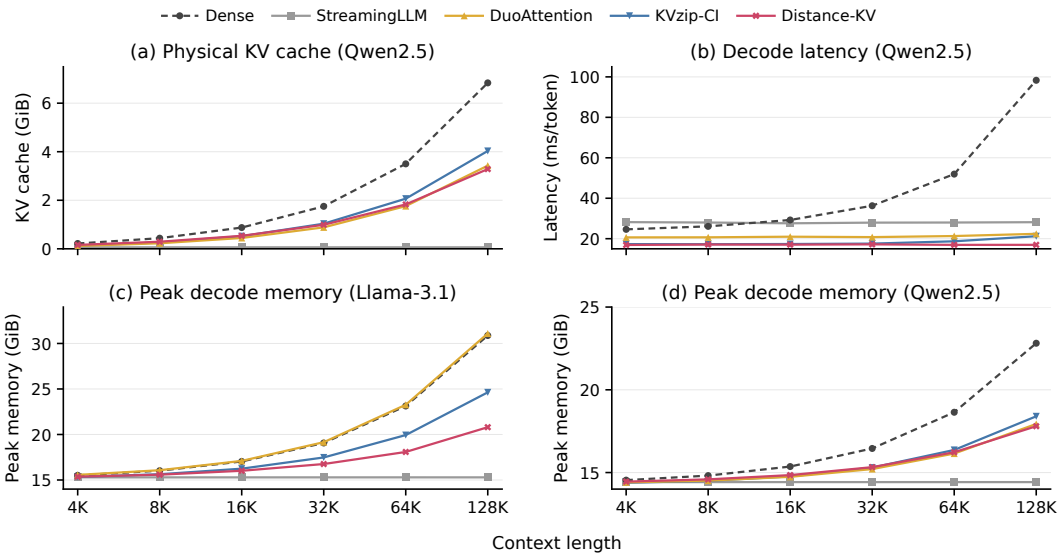


Figure 7: Additional efficiency results on Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct. (a)–(b) Physical KV cache size and per-token decoding latency on Qwen2.5-7B-Instruct. (c)–(d) Peak allocated GPU memory during decoding on both models. All measurements use a single H100 GPU with a batch size of 1.

Table 7: Task-level LongBench results on Qwen2.5-7B-Instruct. Avg. is the unweighted mean over all 21 tasks. Bold indicates the best KV cache compression method in each row.

Task	Dense	Streaming LLM	Duo Attention	MoA	KVzip-CI	Distance KV
Single-document QA						
NarrativeQA	26.85	11.29	**22.92**	22.83	15.45	22.00
Qasper	43.30	18.34	40.38	**42.87**	30.47	41.27
MultiFieldQA-en	52.09	32.85	49.53	40.34	36.96	**50.82**
MultiFieldQA-zh	61.62	29.80	60.62	55.62	50.86	**61.54**
Multi-document QA						
HotpotQA	57.25	32.60	52.70	52.13	41.37	**56.23**
2WikiMQA	46.47	27.35	38.59	39.75	29.20	**43.46**
MuSiQue	30.59	10.03	**29.31**	28.83	21.77	28.85
DuReader	22.06	20.82	23.83	16.37	**24.12**	23.06
Summarization						
GovReport	18.26	16.24	18.12	**18.17**	11.36	**18.17**
QMSum	18.10	14.47	17.94	**17.98**	16.06	17.88
MultiNews	14.73	14.59	14.68	**14.92**	14.13	14.59
VCSUM	14.60	16.63	**16.66**	14.87	15.90	15.21
Few-shot learning						
TREC	65.50	43.50	**66.00**	**66.00**	57.00	62.00
TriviaQA	87.05	84.48	87.50	87.03	82.82	**87.52**
SAMSum	34.91	31.07	35.76	34.35	29.74	**36.20**
LSHT	39.98	19.50	35.56	36.85	31.10	**37.00**
Synthetic tasks						
PassageCount	5.00	3.00	5.00	**8.00**	2.17	6.00
PassageRetrieval-en	99.00	14.00	94.50	86.00	61.00	**97.50**
PassageRetrieval-zh	82.50	22.00	77.00	75.50	26.00	**83.00**
Code completion						
LCC	56.79	54.84	59.50	**60.11**	29.37	58.32
RepoBench-P	63.92	55.64	62.87	**64.39**	59.42	63.38
Avg.	44.79	27.29	43.28	42.04	32.68	**44.00**

Table 8: Task-level RULER results at 128K on Llama-3.1-8B-Instruct. Task Avg. averages all 13 tasks, while Group Avg. averages the eight task groups used in the main-text evaluation. Bold indicates the best KV cache compression method in each row.

Task	Dense	Streaming LLM	Duo Attention	MoA	KVzip-CI	Distance KV
Single NIAH						
Single 1	100.00	0.80	99.60	51.80	0.00	**100.00**
Single 2	98.60	2.20	**97.60**	72.20	0.00	95.60
Single 3	99.60	4.40	97.80	51.40	0.00	**99.40**
Multi-key NIAH						
Multi-key 1	95.20	4.00	95.00	68.80	0.00	**95.60**
Multi-key 2	73.40	0.40	54.20	39.60	0.00	**64.60**
Multi-key 3	58.80	0.20	37.60	18.80	0.00	**40.80**
Other retrieval tasks						
Multi-value	93.35	4.80	88.80	49.20	0.00	**91.05**
Multi-query	97.65	3.20	94.25	65.50	0.00	**97.25**
Tracking and aggregation						
Variable tracking	56.76	1.24	46.80	55.00	0.00	**70.12**
Common words	0.04	**0.28**	0.08	0.10	0.00	0.14
Frequent words	73.87	68.07	65.60	**76.47**	2.73	74.00
Question answering						
QA 1	71.40	14.40	57.60	51.40	9.80	**63.80**
QA 2	43.60	18.60	37.60	30.60	7.40	**39.00**
Task Avg.	74.02	9.43	67.12	48.53	1.53	**71.64**
Group Avg.	69.30	12.26	62.97	48.52	1.42	**68.66**

Table 9: Task-level RULER results at 128K on Qwen2.5-7B-Instruct. Task Avg. averages all 13 tasks, while Group Avg. averages the eight task groups used in the main-text evaluation. Bold indicates the best KV cache compression method in each row.

Task	Dense	Streaming LLM	Duo Attention	MoA	KVzip-CI	Distance KV
Single NIAH						
Single 1	89.80	0.80	**100.00**	75.80	**100.00**	**100.00**
Single 2	27.00	3.00	98.40	73.60	90.40	**99.20**
Single 3	20.00	3.00	91.80	74.20	85.60	**96.60**
Multi-key NIAH						
Multi-key 1	27.40	4.20	92.20	66.40	66.80	**94.20**
Multi-key 2	2.00	0.00	30.60	25.60	13.60	**52.40**
Multi-key 3	0.20	0.60	5.80	9.00	3.00	**20.40**
Other retrieval tasks						
Multi-value	24.25	4.75	54.45	56.05	57.75	**71.60**
Multi-query	16.25	4.20	78.65	66.35	73.10	**90.05**
Tracking and aggregation						
Variable tracking	6.20	0.80	65.96	66.88	71.88	**81.92**
Common words	4.60	0.90	34.16	28.74	19.48	**40.90**
Frequent words	58.80	62.80	**67.33**	49.80	57.00	64.87
Question answering						
QA 1	22.80	18.00	38.60	**71.20**	29.80	49.20
QA 2	21.60	18.00	26.40	34.60	24.20	**37.60**
Task Avg.	24.68	9.31	60.33	53.71	53.28	**69.15**
Group Avg.	23.47	11.91	59.08	53.61	53.25	**68.38**

Table 10: Task-level structural ablation results on RULER at a context length of 128K using Llama-3.1-8B-Instruct. All variants are matched in physical KV retention, and Avg. denotes the mean over all 13 tasks. Bold indicates the best result in each row.

Task	Head-only	Distance-only	Distance-shuffled	Fully shuffled	Distance-KV
Single NIAH					
Single 1	81.00	14.80	74.00	19.00	**93.60**
Single 2	73.20	10.00	87.20	13.00	**95.60**
Single 3	67.20	5.20	93.00	7.00	**98.60**
Multi-key NIAH					
Multi-key 1	69.60	12.80	87.40	14.80	**92.60**
Multi-key 2	6.60	0.20	22.20	4.80	**36.80**
Multi-key 3	0.20	0.20	1.20	1.20	**7.00**
Other retrieval tasks					
Multi-value	84.85	16.30	90.10	9.45	**93.25**
Multi-query	87.25	13.80	92.35	11.70	**95.50**
Tracking and aggregation					
Variable tracking	2.56	3.56	33.60	9.28	**61.36**
Common words	0.02	0.08	0.14	0.04	**0.18**
Frequent words	59.93	55.80	64.53	**76.60**	70.33
Question answering					
QA 1	27.20	15.60	41.60	35.40	**53.00**
QA 2	18.60	16.40	24.00	27.40	**30.60**
Avg.	44.48	12.67	54.72	17.67	**63.72**