# ABSORBKV: CO-DESIGNING SELECTION AND WRITING FOR KV CACHE MERGING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

KV cache compression reduces the memory cost of long-context inference, but eviction irreversibly discards states that future queries may need. KV cache merging instead transfers information from evicted states into retained ones. However, selection constrains the effectiveness of merging by determining which retained states are available to receive information from evicted ones. Existing methods largely overlook this dependency, so a state may be evicted even when none of the retained states can faithfully preserve its information. We introduce *absorbability* to characterize, relative to a given set of retained states, whether the information of an evicted state can remain readable through a retained key without compromising the information originally stored in that retained state. Guided by this insight, we propose ABSORBKV, which co-designs a Hybrid Selector and Adaptive Absorption. The Hybrid Selector jointly considers importance and absorbability to retain states that are important or difficult to absorb. Adaptive Absorption then merges each evicted state only into a compatible retained state and learns how to adapt its information to the retained key through which future queries will retrieve it, while keeping retained keys unchanged. Extensive experiments demonstrate that AbsorbKV outperforms representative KV cache compression methods across diverse long-context tasks and models. On Qwen3-8B with a 10% KV cache budget, AbsorbKV achieves 42.10 on LongBench and 39.01 on RULER-16K, surpassing representative state-of-the-art KV merging methods such as AsymKV, which obtains 35.44 and 25.59, respectively.

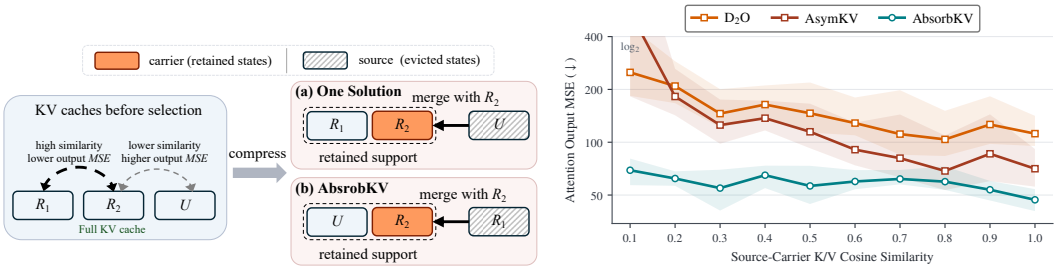
## 1 INTRODUCTION

Large language models are increasingly deployed in agentic workloads, where multi-step interactions with users, tools, and external environments continuously increase pressure on the KV cache, whose memory footprint and bandwidth cost grow with the number of cached tokens (Kang et al., 2025; Wan et al., 2026; Mao et al., 2026). KV cache compression addresses this bottleneck by reducing the size of KV cache states during generation (An et al., 2026; Zhao et al., 2026; Yu et al., 2026; Ramachandran et al., 2026). Most existing approaches achieve this reduction through eviction, retaining a limited subset of important states while discarding the rest. However, eviction is efficient but irreversible. Once a state is removed, both its key, which determines how future queries access it, and its value, which stores the information returned through that access, are lost permanently. KV cache merging offers a way to relax this binary decision by transferring information from evicted states into the retained cache (Wang et al., 2024; Li et al., 2024a; Tian et al., 2026).

Existing work has improved KV cache merging through better merge decisions and more faithful representations of merged information. One line of work uses attention statistics or similarity between states to decide which states to merge and how to weight their contributions (Zhang et al., 2024; Wang et al., 2024; Wan et al., 2025). More recent work also studies how a retained state represents multiple original tokens, accounting for the number of merged tokens and the distinct roles of keys and values (Liu et al., 2025; Cui & Xu, 2025; Liu et al., 2026). Despite these advances, how the choice of retained states constrains subsequent merge remains underexplored. We call an evicted state a *source*, and a retained state that receives source information a *carrier*. **Selecting which KV states to retain determines both the information preserved directly and the carriers available to represent evicted information.** A source may thus be evicted without a compatible carrier, so selection should also consider whether its information can be preserved through subsequent writing.



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




**(a)** Why selection determines what can be written **(b)** Source–carrier compatibility

Figure 1: **Motivation and empirical evidence for absorbability.** **(a)** Retaining the unique state $U$ preserves its retrieval address while allowing the compatible pair $R_1$ and $R_2$ to share a carrier. **(b)** Relationship between source–carrier K/V cosine similarity and attention output MSE. The experiment uses Qwen3-8B on LongBench MultiFieldQA-en with a 10% KV-cache budget, and the experiment details are provided in Appendix A.1.

Figure 1(a) illustrates this dependency under a two slot budget, with a distinct state $U$ and two mutually compatible states $R_1$ and $R_2$. Retaining $R_1$ and $R_2$ leaves no compatible carrier for $U$. Forcing $U$ into $R_2$ can then make the source information difficult to recover while corrupting the carrier's original content. Retaining $U$ and $R_2$, in contrast, preserves $U$'s information and allows $R_1$ to be merged into the compatible carrier $R_2$. The need to retain a state directly therefore depends on whether its information can be preserved through the retained support. We define a source's **absorbability** relative to a given retained support as the ability to keep its information readable through a carrier key without compromising the carrier's original content.

In practice, we find that source–carrier K/V cosine similarity provides a practical proxy for estimating absorbability. Similar keys help future queries access source information through the carrier, while similar values indicate that this information is compatible with the carrier's original content. Figure 1(b) shows that low similarity pairs incur larger attention output MSE than high similarity pairs. This observation implies that source–carrier compatibility should inform both what is retained and how evicted information is merged. Specifically, states that are difficult to absorb should remain in the cache, whereas evicted sources should be routed only to compatible carriers.

Building on this insight, we propose **AbsorbKV**, a KV cache merging method that **co-designs selection and writing** through a *Hybrid Selector* and *Adaptive Absorption*. The Hybrid Selector combines importance with K/V cosine similarity to preserve states that are useful or difficult to absorb. Given the resulting retained support, Adaptive Absorption routes only sources with compatible carriers and learns a carrier conditioned representation for each routed source, allowing its information to be recovered through the carrier key. These carrier conditioned representations are aggregated with the carrier's original value, with each source weighted by its historical attention mass and source–carrier compatibility. The carrier key remains unchanged, preserving the retrieval address used by future attention. As shown in Figure 1(b), AbsorbKV yields lower attention output MSE than $D_2O$ (Wan et al., 2025) and AsymKV (Cui & Xu, 2025) across comparable source–carrier similarity levels. The resulting merged KV cache requires no auxiliary KV slots or modifications to the standard attention interface.

Our contributions are as follows:

- We introduce *absorbability* to characterize whether an evicted state can be represented through a retained carrier while preserving both source readability and the carrier's original information, revealing a fundamental dependency between selection and writing.
- We propose AbsorbKV, which co-designs selection and writing based on absorbability. AbsorbKV retains important states or states that are difficult to absorb, and then routes and writes only compatible sources using carrier conditioned representations.
- Extensive experiments across three models on LongBench and RULER show that AbsorbKV consistently outperforms representative KV cache eviction and merging baselines, with particularly superior performance under aggressive compression.

## 2 Related Work

**KV cache eviction.** KV cache eviction reduces memory by retaining a subset of cached states and discarding the rest (Cai et al., 2025; Jiang et al., 2026). A common approach is to identify persistent or structurally important states from historical attention patterns (Liu et al., 2023; Ge et al., 2024; Adnan et al., 2024). For instance, StreamingLLM (Xiao et al., 2024) preserves initial attention sinks and recent tokens, while $H_2O$ (Zhang et al., 2023) retains recent tokens together with those receiving high accumulated attention. Subsequent work further explores nonuniform allocation across layers and heads (Xiao et al., 2025; Fu et al., 2025). SnapKV (Li et al., 2024b) selects important positions for each head using attention from an observation window at the end of the prompt. PyramidKV (Cai et al., 2024) additionally varies cache budgets across layers to reflect how attention becomes more concentrated in deeper layers. To support reuse across unknown future queries, KVzip (Kim et al., 2025) estimates importance through context reconstruction. Beyond attention weights, the values associated with retained states also determine how eviction affects model outputs. CriticalKV (Feng et al., 2025) incorporates value states and output projection weights into selection by bounding attention output perturbation. CUR (Sengupta et al., 2025) approaches output preservation through approximate key/value leverage scores, which provide the importance measure used by AbsorbKV.

**KV cache merging.** KV cache merging aims to preserve information from evicted states within the retained cache (Wu et al., 2026). CaM (Zhang et al., 2024) transfers evicted values into retained states through attention guided sampling. Similarity based methods further use the relationship between states to decide which ones to merge. Later works exploit relationships among KV cache states and introduce more selective merging strategies. KVMerge (Wang et al., 2024) forms merging sets using key similarity and combines their keys and values with Gaussian kernel weights. $D_2O$ (Wan et al., 2025) couples layer-wise budget allocation with a dynamic similarity threshold for recalling and merging evicted tokens. ZSMerge (Liu et al., 2025) maintains residual merged states with compensated attention scores, explicitly accounting for the fact that a compact state may represent multiple original tokens. Recent work further improves merging by modeling structural properties in the KV cache. AsymKV (Cui & Xu, 2025) distinguishes the roles of keys and values, combining local key merging with cardinality-aware value aggregation that requires an additional logit bias. KVSlimmer (Liu et al., 2026) further derives a closed-form formulation for asymmetric KV merging based on exact Hessian structure. Despite these advances, existing methods primarily focus on how evicted information should be merged or represented after compression. The interaction between the retained support and subsequent writing remains underexplored, even though the selected support determines which carriers are available for absorbing evicted information. AbsorbKV addresses this dependency by coordinating selection and writing through absorbability.

## 3 AbsorbKV

### 3.1 Problem formulation and Overview

**Problem formulation.** Consider a model with $L$ layers, $H$ KV heads per layer, and $N$ cached tokens. Each KV item $(l, h, i)$ consists of a key $k_{l,h,i} \in \mathbb{R}^d$ and a value $v_{l,h,i} \in \mathbb{R}^d$, where $d$ is the head dimension. Given a global budget of $B$ items across all layers and KV heads, KV-cache merging seeks a retained support $S$ with $|S| = B$ and writes information from sources into carriers, with the goal of preserving full cache attention outputs for future queries. We denote the source set by $\mathcal{E}$ and the retained supports in each layer and head by $S_{l,h} = \{i : (l, h, i) \in S\}$. Within a fixed layer, we omit $l$ and write $S_h = S_{l,h}$.

**Overview.** AbsorbKV co-designs selection and writing, as shown in Figure 2. The *Hybrid Selector* ranks all KV items using importance and K/V cosine similarity to retain states that are important or difficult to absorb. Given this retained support, *Adaptive Absorption* routes each source to at most one compatible carrier in the same layer and KV head, discarding sources without a compatible carrier. Carrier conditioned writing learns a representation of each source conditioned on its carrier and uses absorbability to weight that representation. The weighted source representations are then aggregated with the carrier's original value, while the carrier key remains unchanged. The resulting cache preserves the selected KV budget, requiring no auxiliary KV slots or modifications to the standard attention interface.

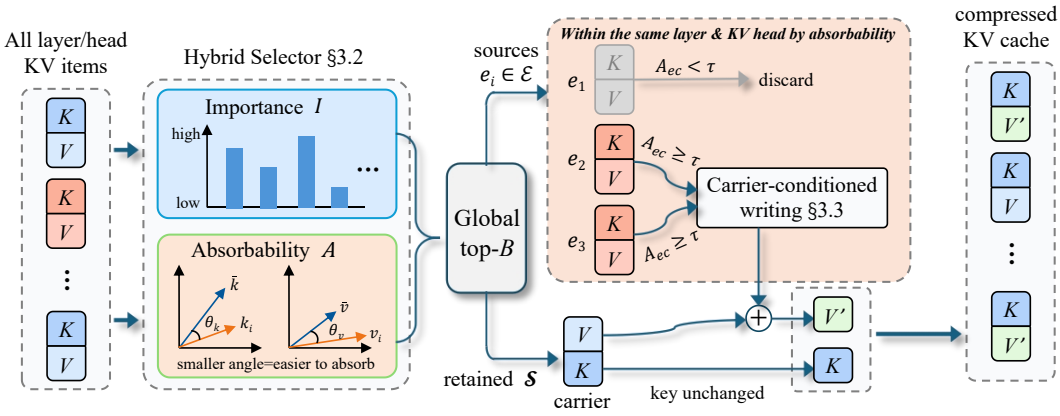



Figure 2: Overview of AbsorbKV. The Hybrid Selector ranks KV items across all layers and heads using importance and K/V cosine similarity. Adaptive Absorption routes compatible sources to retained carriers and writes carrier-conditioned content into their values. Incompatible sources are discarded. The output is a compact KV cache with retained keys unchanged.

### 3.2 Hybrid Selector: Retaining Important and Hard-to-Absorb States

The Hybrid Selector decides which KV states should remain in the compact cache. We therefore assign each KV item $(l, h, i)$ an importance score $I_{l,h,i}$ and an absorbability $A_{l,h,i}$. Since the two signals have different scales, we apply global rank normalization before combining them into a single score,

$$s_{l,h,i} = (1-\lambda)\mathcal{R}(I)_{l,h,i} + \lambda\left[1 - \mathcal{R}(A)_{l,h,i}\right], \qquad \lambda \in [0,1], \tag{1}$$

where $\mathcal{R}$ denotes global rank normalization and $\lambda$ controls the trade off between the two signals. We reserve a small minimum allocation per KV head to avoid empty heads and allocate the remaining budget to the highest scoring items under $s_{l,h,i}$.

**Importance.** For the importance signal of a KV cache, we adopt the approximate key/value leverage score used by CUR (Sengupta et al., 2025). For each layer and KV head, we partition the prompt into non-overlapping blocks of $W$ tokens ($W = 16$ by default). Let $\mathcal{W}(i)$ denote the block containing token $i$. Within each block, $\ell^K_{l,h,i}$ and $\ell^V_{l,h,i}$ measure the relative squared norms of the key and value associated with token $i$, respectively. We multiply the two scores and normalize their products over all $N$ tokens in the same layer and KV head to obtain $I_{l,h,i}$,

$$\ell^K_{l,h,i} = \frac{\|k_{l,h,i}\|_2^2}{\sum_{j\in\mathcal{W}(i)}\|k_{l,h,j}\|_2^2}, \qquad \ell^V_{l,h,i} = \frac{\|v_{l,h,i}\|_2^2}{\sum_{j\in\mathcal{W}(i)}\|v_{l,h,j}\|_2^2},$$
$$I_{l,h,i} = \frac{\ell^K_{l,h,i}\ell^V_{l,h,i}}{\sum_{j=1}^{N}\ell^K_{l,h,j}\ell^V_{l,h,j}}. \tag{2}$$

**Absorbability.** The retained support is unknown at selection time, so source–carrier compatibility cannot yet be evaluated directly. We therefore compute each state's key and value cosine similarities to their respective mean directions within the same layer and KV head. Let $\hat{k}_{l,h,i} = k_{l,h,i}/\|k_{l,h,i}\|_2$ and $\hat{v}_{l,h,i} = v_{l,h,i}/\|v_{l,h,i}\|_2$ denote normalized keys and values. We define the corresponding mean directions as

$$\bar{k}_{l,h} = \frac{\sum_{j=1}^{N}\hat{k}_{l,h,j}}{\left\|\sum_{j=1}^{N}\hat{k}_{l,h,j}\right\|_2}, \qquad \bar{v}_{l,h} = \frac{\sum_{j=1}^{N}\hat{v}_{l,h,j}}{\left\|\sum_{j=1}^{N}\hat{v}_{l,h,j}\right\|_2}.$$

We calculate the absorbability as the average K/V cosine similarity to these mean directions,

$$A_{l,h,i} = \frac{1}{2}\left[\hat{k}^\top_{l,h,i}\bar{k}_{l,h} + \hat{v}^\top_{l,h,i}\bar{v}_{l,h}\right]. \tag{3}$$

By jointly considering importance and absorbability, the Hybrid Selector prioritizes states that are important or difficult to absorb, leaving more compatible sources for subsequent writing. We provide a theoretical analysis of the selection–writing dependency in Appendix B.

### 3.3 Adaptive Absorption: Routing and Carrier-Conditioned Writing

**Absorbability-aware Routing.** Once the retained support is fixed, we assess source–carrier compatibility directly using joint K/V cosine similarity. For each source $e$, we choose the most similar retained state within the same layer and KV head as its carrier $c(e)$. Omitting layer and head indices, the similarity and carrier assignment are

$$A_{ec} = \frac{1}{2}\left(\hat{k}_e^\top \hat{k}_c + \hat{v}_e^\top \hat{v}_c\right), \qquad c(e) = \arg\max_{c \in S_h} A_{ec}. \tag{4}$$

If the best match satisfies $A_{e,c(e)} \geq \tau$, where $\tau \in [0, 1)$ is a fixed similarity threshold, we route $e$ to $c(e)$. Otherwise, the source is dropped permanently.

**Carrier-conditioned writing.** After routing, a source no longer retains its own key, and its information can only be accessed through the attention weight assigned to the carrier. Simply averaging the original source value into the carrier ignores this change in retrieval address. We therefore transform each source into a carrier-conditioned representation that adapts its content to the new retrieval address. Figure 3 illustrates this process when multiple sources share one carrier. Each routed source produces a carrier-conditioned representation together with an absorption weight, and the resulting contributions are aggregated with the carrier's original value.

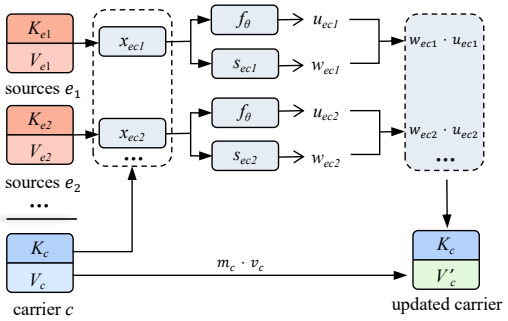



Figure 3: Carrier-conditioned writing. Routed sources are transformed relative to the carrier and aggregated with the carrier's original value, while the carrier key remains unchanged.

For a routed source–carrier pair $(e, c)$, the writer takes their K/V differences, relative position, and the source attention mass as input, and outputs a carrier-conditioned representation of the source value,

$$\begin{aligned} x_{ec} &= \mathrm{concat}(v_e - v_c,\ k_e - k_c,\ \mathrm{PE}(p_e - p_c),\ \log m_e) \in \mathbb{R}^{2d+17}, \\ u_{ec} &= v_e + f_\theta(x_{ec}), \qquad f_\theta : \mathbb{R}^{2d+17} \to \mathbb{R}^d. \end{aligned} \tag{5}$$

The learned transformation $f_\theta$ is a small layer-specific MLP shared across KV heads and trained as described in Section 3.4. On Qwen3-8B, $f_\theta$ contains 6.46M trainable parameters in total, corresponding to 0.079% of the backbone parameters. The key difference $k_e - k_c$ captures the change in retrieval address when the source is accessed through the carrier, while $v_e - v_c$ captures the difference between their stored content. The positional feature $\mathrm{PE}(p_e - p_c)$ encodes the source–carrier offset using a fixed 16-dimensional sinusoidal embedding. The attention mass $m_e$ summarizes how much attention the source receives from recent context queries, accumulated over the final up to 64 context queries on the full cache and averaged across query heads.

We then weight the contribution of $u_{ec}$ using the source's observed attention mass $m_e$ and an absorption coefficient derived from the routing similarity $A_{ec}$,

$$\alpha_{ec} = \left[\frac{A_{ec} - \tau}{1 - \tau}\right]_+, \qquad w_{ec} = m_e \alpha_{ec}, \tag{6}$$

where $[z]_+ = \max(z, 0)$.

Moreover, multiple sources may be routed to the same carrier. We combine their carrier-conditioned representations with the carrier's original value using normalized weights. Let $\mathcal{E}_c$ denote the sources routed to carrier $c$, and let $m_c$ be the carrier's observed attention mass. The updated value is

$$v'_c = \frac{m_c v_c + \sum_{e \in \mathcal{E}_c} w_{ec} u_{ec}}{m_c + \sum_{e \in \mathcal{E}_c} w_{ec}}, \qquad k'_c = k_c. \tag{7}$$

### 3.4 Writer Training

The writer must transform source information before the future queries are known. We therefore use future queries only as training supervision, while $f_\theta$ receives only context-derived features available

Table 1: Performance on LongBench tasks with various models (90% compression ratio).

| | | Single-doc QA | | | Multi-doc QA | | | Summarization | | | Few-shot Learning | | | Synthetic | | Code | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Method** | NrtvQA | Qasper | MF-en | HotpotQA | 2WikiMQA | MuSiQue | GovReport | QMSum | MultiNews | TREC | TriviaQA | SAMSum | PCount | PR-en | LCC | RB-P | **Avg.** |
| Llama-3.1-8B Instruct | Full KV | 28.29 | 47.24 | 55.38 | 59.17 | 50.99 | 32.55 | 35.06 | 25.16 | 26.88 | 29.00 | 84.99 | 38.93 | 10.70 | 100.00 | 51.48 | 44.49 | 45.02 |
| | CaM | 19.02 | 28.90 | 35.02 | 36.28 | 32.03 | 19.12 | 25.69 | 20.19 | 23.19 | 8.25 | 70.84 | 14.41 | 5.56 | 39.00 | 32.47 | 52.74 | 28.92 |
| | $D_2O$ | 18.40 | 23.66 | 31.70 | 45.01 | 35.46 | 16.69 | 23.88 | 21.56 | 21.59 | 6.50 | 90.24 | 32.16 | 4.50 | 61.00 | 40.16 | 53.21 | 32.86 |
| | AsymKV | 26.57 | 22.47 | 31.93 | 45.75 | 23.50 | 19.19 | 27.28 | 22.31 | 23.20 | 47.50 | 87.28 | 16.45 | 5.00 | 55.00 | 27.95 | 42.78 | 32.76 |
| | CUR | 20.17 | 32.61 | 35.60 | 47.09 | 38.26 | 22.57 | 27.27 | 22.26 | 22.78 | 11.00 | 90.32 | 32.98 | 5.00 | 69.00 | 45.81 | 53.83 | 36.03 |
| | **AbsorbKV** | 27.09 | 30.61 | 37.17 | 46.17 | 42.73 | 26.14 | 28.85 | 22.64 | 23.93 | 38.00 | 89.33 | 39.40 | 8.00 | 77.00 | 31.80 | 49.84 | **38.67** |
| Qwen3-8B | Full KV | 26.66 | 44.27 | 55.71 | 62.80 | 48.41 | 35.19 | 33.44 | 24.67 | 24.92 | 41.50 | 90.28 | 40.00 | 9.00 | 90.85 | 64.90 | 60.05 | 47.04 |
| | CaM | 1.32 | 12.49 | 14.15 | 5.43 | 14.59 | 3.79 | 15.20 | 10.95 | 19.72 | 2.12 | 76.56 | 25.91 | 4.24 | 61.60 | 26.87 | 56.51 | 21.97 |
| | $D_2O$ | 16.56 | 24.50 | 33.45 | 33.29 | 30.89 | 15.62 | 26.49 | 21.56 | 20.09 | 18.75 | 47.49 | 26.15 | 3.50 | 25.50 | 38.51 | 62.38 | 27.80 |
| | AsymKV | 21.95 | 21.71 | 31.72 | 40.84 | 26.66 | 15.84 | 28.20 | 20.45 | 20.51 | 58.25 | 89.47 | 37.10 | 6.00 | 49.04 | 40.71 | 58.55 | 35.44 |
| | CUR | 22.49 | 29.45 | 33.56 | 48.19 | 38.97 | 26.33 | 29.75 | 21.68 | 21.31 | 29.00 | 81.59 | 36.04 | 9.08 | 75.68 | 51.75 | 63.07 | 38.62 |
| | **AbsorbKV** | 25.65 | 31.45 | 39.08 | 50.84 | 38.96 | 30.38 | 29.35 | 22.31 | 22.45 | 48.75 | 89.51 | 37.93 | 15.05 | 82.89 | 47.10 | 61.86 | **42.10** |
| Qwen3-4B | Full KV | 24.97 | 40.98 | 56.27 | 61.02 | 46.34 | 35.02 | 31.65 | 24.89 | 25.26 | 48.00 | 87.10 | 40.32 | 6.50 | 99.00 | 59.87 | 53.78 | 46.31 |
| | CaM | 2.71 | 10.54 | 10.95 | 5.51 | 12.63 | 5.04 | 15.76 | 11.99 | 18.77 | 2.00 | 72.83 | 26.07 | 2.10 | 22.83 | 21.86 | 50.25 | 18.24 |
| | $D_2O$ | 15.53 | 19.66 | 27.33 | 26.73 | 26.37 | 14.77 | 24.07 | 21.37 | 20.33 | 30.00 | 76.60 | 33.99 | 4.50 | 22.50 | 36.47 | 54.91 | 28.45 |
| | AsymKV | 20.53 | 18.68 | 30.92 | 41.50 | 24.14 | 19.48 | 26.50 | 21.27 | 20.67 | 54.50 | 86.31 | 38.65 | 2.50 | 42.88 | 39.14 | 55.56 | 33.95 |
| | CUR | 20.58 | 29.81 | 33.49 | 45.43 | 35.30 | 23.93 | 26.92 | 22.10 | 21.27 | 34.25 | 85.78 | 36.31 | 7.00 | 73.25 | 43.32 | 59.38 | 37.38 |
| | **AbsorbKV** | 21.77 | 30.04 | 39.69 | 49.43 | 32.29 | 27.69 | 28.14 | 22.53 | 22.60 | 58.25 | 86.99 | 37.99 | 6.00 | 83.00 | 48.72 | 59.35 | **40.91** |

at compression time. Following LongBench's data construction procedure, we construct 100 training examples for each task whose original source dataset provides sufficient additional data, using examples disjoint from the LongBench evaluation set.

To isolate the effect of writing from that of selection, we construct a Full KV teacher, an AbsorbKV student, and a Direct Eviction student on the same query. The two students use the same retained support, with AbsorbKV differing only in its writes to the retained values. The training objective is

$$\mathcal{L} = \mathcal{L}_{\text{out}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}^{\text{abs}} + \lambda_{\text{rel}} \left[ \mathcal{L}_{\text{KL}}^{\text{abs}} - \mathcal{L}_{\text{KL}}^{\text{dir}} \right]_+ , \tag{8}$$

where $\mathcal{L}_{\text{out}}$ matches the attention outputs of the compressed and Full KV caches under the same queries, and $\mathcal{L}_{\text{KL}}^{x}$ measures the KL divergence between the teacher's next-token distribution and that of student $x \in \{\text{abs}, \text{dir}\}$. The final term penalizes any excess KL divergence of AbsorbKV over Direct Eviction under the same retained support. The hyperparameter $\lambda_{\text{KL}}$ weights next-token distribution matching, while $\lambda_{\text{rel}}$ controls this relative penalty.

At inference time, $f_\theta$ is invoked only when compression is performed. The updated values are then stored in the compact cache, and subsequent decoding proceeds with standard attention until the next compression event. Additional data construction and training details are provided in Appendix A.2.

# 4 Experiments

## 4.1 Experimental Setup

**Models and benchmarks.** We evaluate AbsorbKV and the baselines on Llama-3.1-8B-Instruct (Team, 2024), Qwen3-8B, and Qwen3-4B (Team, 2025). We use the 16 English tasks from LongBench (Bai et al., 2024), covering single- and multi-document question answering, summarization, few-shot learning, synthetic tasks, and code completion. We additionally evaluate on RULER (Hsieh et al., 2024) at multiple context lengths, spanning retrieval, multi-hop tracing, aggregation, and question answering tasks. For both benchmarks, we follow the official evaluation metrics and report macro averages with equal weight across tasks.

**Baselines.** We compare against three representative KV merging methods, CaM (Zhang et al., 2024), $D_2O$ (Wan et al., 2025) and AsymKV (Cui & Xu, 2025). We also include CUR (Sengupta et al., 2025) as a KV eviction baseline, whose key/value leverage scores provide the importance

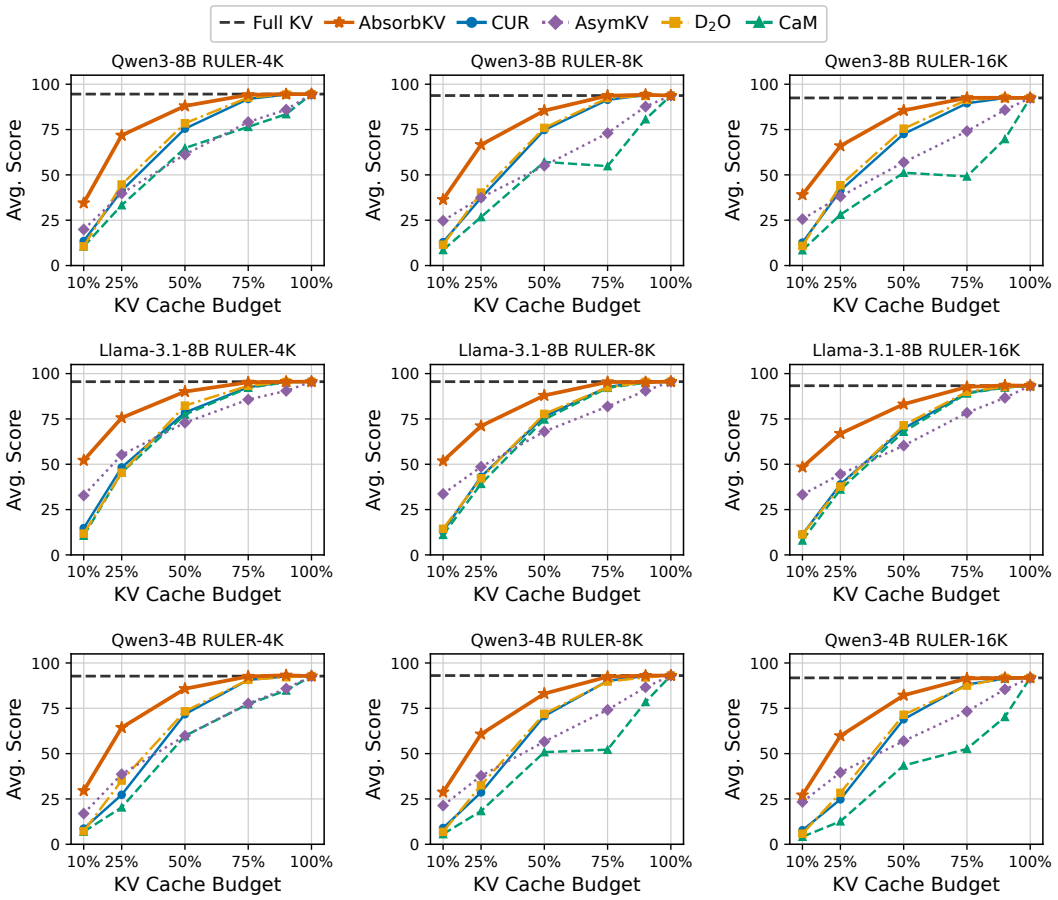



Figure 4: Performance on RULER tasks with varying cache sizes.

signal used by AbsorbKV. Additional comparisons with a broader set of KV cache compression baselines are provided in Appendix C.2.

**Implementation details.** All methods are evaluated in the question-agnostic setting. The context cache is compressed before the task question is observed, after which the question and generated-token states are appended without further compression. For AbsorbKV, we set $\lambda = 0.25$ and $\tau = 0.7$, with sensitivity analysis provided in Appendix C.1. For CaM and $D_2O$, the experiments use CUR as the external selector, while AsymKV retains its own selection policy. All experiments are conducted on a single NVIDIA A100 GPU with 80 GB of memory. Additional evaluation settings, baseline configurations, and writer training details are provided in Appendix A.

### 4.2 Accuracy Comparison on Long-Context Tasks

**LongBench Results.** Table 1 reports LongBench results with only 10% of the context KV cache retained. AbsorbKV achieves the highest average score among the compression methods on all three models, preserving 85.9–89.5% of Full KV performance. Compared with CUR, the strongest baseline, it improves the average score by 2.64, 3.48, and 3.53 on Llama-3.1-8B-Instruct, Qwen3-8B, and Qwen3-4B, respectively. All other merging baselines score below CUR on average, showing that merging does not always improve upon eviction. AbsorbKV instead improves the performance by coordinating which states are retained with how evicted information is written into them. Furthermore, we report LongBench results with various KV cache budgets in Appendix C.5.

**RULER Results.** Figure 4 reports average scores over the 13 RULER tasks on three models with context lengths from 4K to 16K. AbsorbKV substantially reduces the performance loss under ag-

gressive compression and approaches Full KV performance as the cache budget increases. For instance, at 16K with a 25% budget, AbsorbKV exceeds the strongest baseline by 21.42 on average across the three models, showing consistent gains on both the 8B models and the smaller Qwen3-4B. To examine whether these gains extend to longer contexts, we further evaluate Llama-3.1-8B-Instruct at 32K, 64K, and 128K, as shown in Figure 5. AbsorbKV outperforms the evaluated baselines across all KV cache budgets and context lengths. At 128K with a 25% KV cache budget, AbsorbKV reaches 47.27 and exceeds the strongest baseline $D_2O$ by 21.61. With a 75% KV cache budget, AbsorbKV reaches 75.94, approaching the Full KV score of 79.01. These results show that AbsorbKV remains effective at preserving information needed by subsequent queries as the context extends to 128K. Per-task comparisons and detailed scores are provided in Appendix C.6.

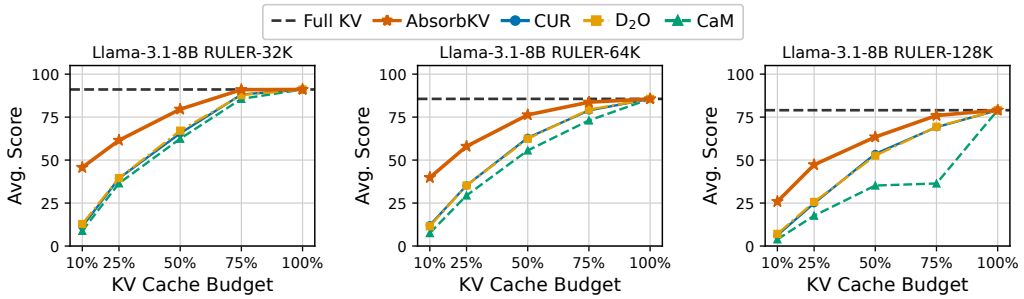
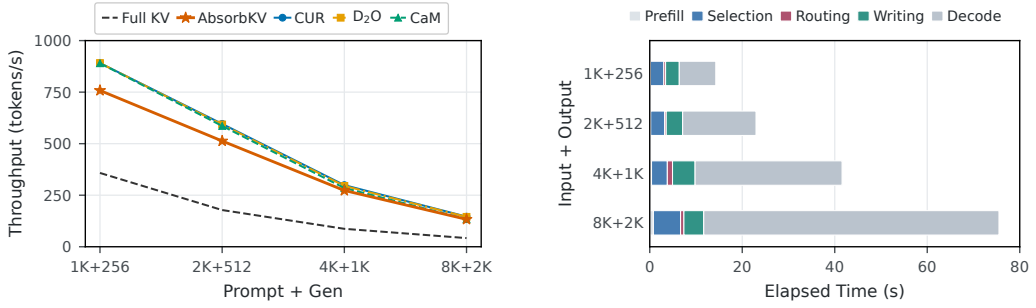



Figure 5: RULER results across KV cache budgets at 32K, 64K, and 128K context lengths. We use Llama-3.1-8B-Instruct for this evaluation to keep all tested lengths within its native context window.

**Writer generalization.** The RULER results also provide evidence of writer generalization. The writer is trained exclusively on examples constructed from the original source datasets of LongBench and is applied to RULER without any additional training or adaptation. Moreover, RULER differs substantially from LongBench in both task construction and query patterns, particularly through its synthetic retrieval, multi-hop tracing, and aggregation tasks. The consistent gains therefore indicate that the writer generalizes beyond the training distribution and remains effective on unseen tasks.

### 4.3 Throughput Analysis




**(a)** Throughput comparison

**(b)** Runtime breakdown

Figure 6: Efficiency on Qwen3-8B. **(a)** Throughput across prompt and generation lengths. **(b)** AbsorbKV runtime breakdown for the same length pairs at batch size 1 and a 10% KV-cache budget.

Figure 6(a) compares maximum end-to-end throughput on Qwen3-8B across different prompt and generation lengths. AsymKV is omitted from the throughput results because it relies on a per-slot logit bias that is not supported by the FlashAttention backend used here. AbsorbKV achieves a 2.12–3.17× speedup over Full KV, while the remaining compression baselines attain slightly higher

throughput. However, this higher throughput comes at the cost of substantially lower task performance under the same cache budget, as shown in Table 1 and Figure 4. More importantly, the throughput gap between AbsorbKV and the baselines steadily narrows as the workload grows. Figure 6(b) explains this trend: selection, routing, and writing are performed only during cache compression rather than at every decoding step, so their share of end-to-end latency decreases as the workload increases. At 8K+2K, for example, the writer accounts for only 5.68% of the total latency.

### 4.4 Ablation Study

In this section, we conduct a series of experiments to investigate the importance of each component in AbsorbKV. Unless otherwise specified, Qwen3-4B is used as the default model under a 10% KV cache budget. For a fair comparison, we evaluate all variants on LongBench datasets spanning six task categories: single-document QA (Qasper), multi-document QA (2WikiMQA), summarization (QMSum), few-shot learning (TREC), synthetic tasks (PR-en), and code completion (RB-P).

Table 2 first reveals a clear interaction between selection and writing. Replacing the Hybrid Selector with importance-only selection reduces the full AbsorbKV score from 47.58 to 43.63, while removing the writer reduces it to 44.59. More importantly, the benefit of writing depends strongly on the retained support. With Hybrid selection, writing improves the score from 44.59 to 47.58, a gain of 2.99. Under importance-only selection, the corresponding improvement is only 1.28, from 42.35 to 43.63. Thus, absorbability aware selection not only improves the retained cache itself, but also creates a support on which subsequent writing is substantially more effective.

The remaining ablations examine writing and source–carrier compatibility. Raw-V Aggregation replaces the learned carrier-conditioned representations with the original source values and aggregates them using their observed attention masses. Selection and routing remain the same as in AbsorbKV. Raw-V Aggregation achieves 44.74, only 0.15 above direct eviction and 2.84 below AbsorbKV. This gap shows the importance of adapting source information to the carrier through which it will be retrieved.

Table 2: Ablation study on Qwen3-4B with a 10% KV-cache budget. Scores are averaged over the LongBench tasks. $\Delta$ denotes the score change relative to AbsorbKV.

| Variant | Avg. | $\Delta$ |
|---|---|---|
| Full KV | 52.17 | – |
| **AbsorbKV** | **47.58** | 0.00 |
| w/o Absorbability in Selection | 43.63 | -3.95 |
| w/o Writer | 44.59 | -2.99 |
| w/o Absorbability in Selection & Writer | 42.35 | -5.23 |
| w/ Raw-V Aggregation | 44.74 | -2.84 |
| w/ Key-only Similarity | 46.51 | -1.07 |
| w/ Pre-RoPE Keys | 47.40 | -0.18 |

Using key-only similarity for both selection and routing reduces the score to 46.51, indicating that value similarity provides complementary information for AbsorbKV. Using Pre-RoPE rather than the default Post-RoPE keys yields a similar score of 47.40. However, Pre-RoPE similarity requires storing an additional copy of the key representations, thereby roughly doubling key-storage overhead and increasing peak memory. AbsorbKV therefore operates directly on the cached Post-RoPE keys.

## 5 Conclusion

In this paper, we show that selection and writing in KV cache merging are strongly interdependent because the retained support determines how evicted information can be retrieved by future queries. We introduce absorbability to characterize whether source information can remain readable through a carrier while preserving the carrier's original content. Guided by this insight, AbsorbKV co-designs a Hybrid Selector and Adaptive Absorption to retain important states or states that are difficult to absorb and learn how to write compatible source information into retained carriers. Experiments on LongBench and RULER across three language models demonstrate improved performance over representative eviction and merging baselines, particularly under aggressive compression. AbsorbKV introduces no additional attention bias and preserves the standard attention formulation, thereby maintaining compatibility with FlashAttention. Future work will extend AbsorbKV to streaming inference, allowing the retained support and writing strategy to adapt as long horizon interactions evolve.

### AI USE STATEMENT

We used generative AI tools for language polishing, including improvements to grammar, clarity, and readability of this paper. Generative AI was not used to generate research ideas, formulate the proposed method or determine the claims or conclusions of this work. All AI-assisted edits were reviewed by the authors for technical accuracy and consistency with the intended meaning. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

All datasets used in our experiments are publicly available, and we follow the evaluation protocols provided by their official repositories. The benchmark settings and evaluation procedures are described in Appendix A.1. Appendix A.2 describes the construction of AbsorbKV's training data, together with the implementation details and hyperparameter settings used for training and inference. Additional comparisons, hyperparameter sensitivity analyses, and task-level results are provided in Appendix C. We will release the complete implementation upon acceptance.

## REFERENCES

Muhammad Adnan, Akhil Arunkumar, Gaurav Jain, Prashant J. Nair, Ilya Soloveychik, and Purushotham Kamath. Keyformer: KV cache reduction through key tokens selection for efficient generative inference. In Phillip B. Gibbons, Gennady Pekhimenko, and Christopher De Sa (eds.), *Proceedings of the Seventh Annual Conference on Machine Learning and Systems, MLSys 2024, Santa Clara, CA, USA, May 13-16, 2024*. mlsys.org, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/48fecef47b19fe501d27d338b6d52582-Abstract-Conference.html.

Yongqi An, Chang Lu, Kuan Zhu, Tao Yu, Chaoyang Zhao, Hong Wu, Ming Tang, and Jinqiao Wang. Rest-kv: Robust KV cache eviction with layer-wise output reconstruction and spatial-temporal smoothing. *CoRR*, abs/2605.08840, 2026. doi: 10.48550/ARXIV.2605.08840. URL https://doi.org/10.48550/arXiv.2605.08840.

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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 3119–3137. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.172. URL https://doi.org/10.18653/v1/2024.acl-long.172.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Baobao Chang, Junjie Hu, and Wen Xiao. Pyramidkv: Dynamic KV cache compression based on pyramidal information funneling. *CoRR*, abs/2406.02069, 2024. doi: 10.48550/ARXIV.2406.02069. URL https://doi.org/10.48550/arXiv.2406.02069.

Zefan Cai, Wen Xiao, Hanshi Sun, Cheng Luo, Yikai Zhang, Ke Wan, Yucheng Li, Yeyang Zhou, Li-Wen Chang, Jiuxiang Gu, Zhen Dong, Animashree Anandkumar, Abedelkadir Asi, and Junjie Hu. R-KV: redundancy-aware KV cache compression for reasoning models. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/57e0358aed314784cad14a26e1eba642-Abstract-Conference.html.

Wanyun Cui and Mingwei Xu. Homogeneous keys, heterogeneous values: Exploiting local KV cache asymmetry for long-context llms. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván

Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/750b0f9fccafad88e0da366315e03d1a-Abstract-Conference.html.

Alessio Devoto, Maximilian Jeblick, and Simon Jégou. Expected attention: KV cache compression by estimating attention from future queries distribution. *CoRR*, abs/2510.00636, 2025. doi: 10.48550/ARXIV.2510.00636. URL https://doi.org/10.48550/arXiv.2510.00636.

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

Yu Fu, Zefan Cai, Abedelkadir Asi, Wayne Xiong, Yue Dong, and Wen Xiao. Not all heads matter: A head-level KV cache compression method with integrated retrieval and reasoning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=FJFVmeXusW.

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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=uNrFpDPMyo.

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? *CoRR*, abs/2404.06654, 2024. doi: 10.48550/ARXIV.2404.06654. URL https://doi.org/10.48550/arXiv.2404.06654.

Jiantong Jiang, Peiyu Yang, Rui Zhang, and Feng Liu. Towards efficient large language model serving: A survey on system-aware KV cache optimization. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 38450–38476. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.FINDINGS-ACL.1916. URL https://doi.org/10.18653/v1/2026.findings-acl.1916.

Minki Kang, Wei-Ning Chen, Dongge Han, Huseyin A. Inan, Lukas Wutschitz, Yanzhi Chen, Robert Sim, and Saravan Rajmohan. ACON: optimizing context compression for long-horizon LLM agents. *CoRR*, abs/2510.00615, 2025. doi: 10.48550/ARXIV.2510.00615. URL https://doi.org/10.48550/arXiv.2510.00615.

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 Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/f4eaa4b8f2d08edb3f0af990d56134ea-Abstract-Conference.html.

Yingxin Li, Ye Li, Yuan Meng, Xinzhu Ma, Zihan Geng, Shutao Xia, and Zhi Wang. EMS: adaptive evict-then-merge strategy for head-wise KV cache compression based on global-local importance. *CoRR*, abs/2412.08521, 2024a. doi: 10.48550/ARXIV.2412.08521. URL https://doi.org/10.48550/arXiv.2412.08521.

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 Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information*

*Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024b. URL http://papers.nips.cc/paper_files/paper/2024/hash/28ab418242603e0f7323e54185d19bde-Abstract-Conference.html.

Lianjun Liu, Hongli An, Weiqi Yan, Xin Du, Shengchuan Zhang, Huazhong Liu, and Yunshan Zhong. Kvslimmer: Theoretical insights and practical optimizations for asymmetric KV merging. *CoRR*, abs/2603.00907, 2026. doi: 10.48550/ARXIV.2603.00907. URL https://doi.org/10.48550/arXiv.2603.00907.

Xin Liu, Xudong Wang, Pei Liu, and Guoming Tang. Zsmerge: Zero-shot kv cache compression for memory-efficient long-context llms. *arXiv preprint arXiv:2503.10714*, 2025.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a452a7c6c463e4ae8fbdc614c6e983e6-Abstract-Conference.html.

Yuzhen Mao, Qitong Wang, Martin Ester, and Ke Li. Icecache: Memory-efficient kv-cache management for long-sequence llms. *CoRR*, abs/2604.10539, 2026. doi: 10.48550/ARXIV.2604.10539. URL https://doi.org/10.48550/arXiv.2604.10539.

Matanel Oren, Michael Hassid, Yarden Nir, Yossi Adi, and Roy Schwartz. Transformers are multi-state rnns. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 18724–18741. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.1043. URL https://doi.org/10.18653/v1/2024.emnlp-main.1043.

Akshat Ramachandran, Marina Neseem, Charbel Sakr, Rangharajan Venkatesan, Brucek Khailany, and Tushar Krishna. Thinkv: Thought-adaptive kv cache compression for efficient reasoning models. In *International Conference on Learning Representations*, volume 2026, pp. 110072–110106, 2026.

Ayan Sengupta, Siddhant Chaudhary, and Tanmoy Chakraborty. Value-guided KV compression for llms via approximated CUR decomposition. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/721b0b930cae88656b0f71a8f6097ee4-Abstract-Conference.html.

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.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Yuxuan Tian, Zihan Wang, Yebo Peng, Aomufei Yuan, Zhiming Wang, Bairen Yi, Xin Liu, Yong Cui, and Tong Yang. Keepkv: Achieving periodic lossless KV cache compression for efficient LLM inference. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 33259–33267. AAAI Press, 2026. doi: 10.1609/AAAI.V40I39.40611. URL https://doi.org/10.1609/aaai.v40i39.40611.

Guangya Wan, Mingyang Ling, Xiaoqi Ren, Rujun Han, Sheng Li, and Zizhao Zhang. COMPASS: enhancing agent long-horizon reasoning with evolving context. 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), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 3360–3380. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.152. URL https://doi.org/10.18653/v1/2026.acl-long.152.

Zhongwei Wan, Xinjian Wu, Yu Zhang, Yi Xin, Chaofan Tao, Zhihong Zhu, Xin Wang, Siqi Luo, Jing Xiong, Longyue Wang, and Mi Zhang. D2O: dynamic discriminative operations for efficient long-context inference of large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=HzBfoUdjHt.

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. *CoRR*, abs/2407.08454, 2024. doi: 10.48550/ARXIV.2407.08454. URL https://doi.org/10.48550/arXiv.2407.08454.

Shunlong Wu, Hai Lin, Shaoshen Chen, Tingwei Lu, Yongqin Zeng, Shaoxiong Zhan, Hai-Tao Zheng, and Hong-Gee Kim. Semanticache: Efficient kv cache compression via semantic chunking and clustered merging. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 19562–19566. IEEE, 2026.

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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 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, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=cFu7ze7xUm.

Zhiyuan Yu, Shijian Xiao, Zhangyue Yin, Xiaoran Liu, Lekai Xing, Wenzhong Li, Cam-Tu Nguyen, and Sanglu Lu. Protokv: Long-context knowledges are already well-organized before your query. In *International Conference on Learning Representations*, volume 2026, pp. 142982–143009, 2026.

Yuxin Zhang, Yuxuan Du, Gen Luo, Yunshan Zhong, Zhenyu Zhang, Shiwei Liu, and Rongrong Ji. Cam: Cache merging for memory-efficient llms inference. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 58840–58850. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/zhang24n.html.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark W. Barrett, Zhangyang Wang, and Beidi Chen. H2O: heavy-hitter oracle for efficient generative inference of large language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/6ceefa7b15572587b78ecfcebb2827f8-Abstract-Conference.html.

Liang Zhao, Xiaocheng Feng, Weihong Zhong, Lei Huang, Kun Zhu, Baoxin Wang, Dayong Wu, Guoping Hu, Ting Liu, and Bing Qin. Question tells you where the answer is: Intention-aware long-context KV cache compression. 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), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 27153–27169. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.1250. URL https://doi.org/10.18653/v1/2026.acl-long.1250.

# A Implementation and Reproducibility Details

This appendix describes the evaluation protocol, baseline adaptations, and implementation details of AbsorbKV. It also specifies the construction of training examples and supervision to support reproducibility of AbsorbKV.

## A.1 Benchmarks and Scoring

**LongBench.** We evaluate on the 16 English tasks of LongBench (Bai et al., 2024), with their categories and evaluation metrics summarized in Table 3. We follow the official prediction preprocessing and scoring rules. Scores are averaged over examples.

Table 3: The 16 English LongBench tasks used in our evaluation and the scoring metrics.

| Category | Tasks | Metrics |
|---|---|---|
| Single-document QA | NarrativeQA, Qasper, MultiFieldQA-en | F1 |
| Multi-document QA | HotpotQA, 2WikiMultihopQA, MuSiQue | F1 |
| Summarization | GovReport, QMSum, MultiNews | ROUGE-L |
| Few-shot learning | TREC, TriviaQA, SAMSum | Classification accuracy; F1 for TriviaQA; ROUGE-L for SAMSum |
| Synthetic tasks | PassageCount, PassageRetrieval-en | Count and retrieval scores |
| Code completion | LCC, RepoBench-P | Edit similarity |

**RULER.** We evaluate all 13 RULER (Hsieh et al., 2024) tasks at context lengths ranging from 4K to 128K. The tasks cover retrieval with varying numbers of keys, values, and queries, as well as variable tracking, common- and frequent-word extraction, and question answering. Scoring uses case-insensitive string matching. Question answering accepts any reference alias, while tasks requiring multiple answers are scored by the fraction of reference answers matched.

**Experiment setting in Figure 1(b).** We study source–carrier compatibility on MultiFieldQA-en with Qwen3-8B and a 10% KV-cache budget. CUR determines a fixed retained support, and each merge modifies one carrier using one evicted source. All writers receive the same source–carrier pairs chosen by AbsorbKV, so differences in reconstruction error arise from their updates. We group pairs by their joint K/V cosine similarity and measure attention output MSE relative to Full KV. Shaded intervals reflect variation across input examples.

## A.2 AbsorbKV Implementation

**Implementation settings.** Table 4 summarizes the implementation settings of AbsorbKV. For selection, $W$ specifies the block size used to compute CUR importance, while $\lambda$ controls the balance between importance and absorbability. We reserve a minimum allocation of $0.1rT$ states for each KV head to avoid empty heads, where $r$ is the KV cache budget ratio and $T$ is the prompt length. Both selection and routing operate directly on cached Post-RoPE keys, avoiding the need to store an additional copy of Pre-RoPE representations. $f_\theta$ is a three-layer MLP with SiLU activations, shared across KV heads. The source–carrier relative position is encoded using a fixed 16-dimensional sinusoidal embedding. The source attention mass is accumulated over up to the final 64 context queries. The learned transformation $f_\theta$ is lightweight. On Qwen3-8B, for example, $f_\theta$ contains only 6.46M trainable parameters in total, corresponding to approximately 0.079% of the backbone model.

**Training Data Construction.** We construct the training dataset from the original source datasets used by LongBench rather than from the LongBench evaluation set. For each task, we sample 100 examples for training and process them using the same task specific context and query construction procedure as LongBench. Tasks without sufficient additional examples are excluded from writer training. This construction keeps the training data disjoint from the LongBench evaluation set while preserving the input format of each task. RULER is not used for training.

Table 4: AbsorbKV implementation settings.

| Setting | Value |
|---|---|
| CUR block size $W$ | 16 tokens |
| Hybrid weight $\lambda$ | 0.25 |
| Routing threshold $\tau$ | 0.7 |
| Per-head minimum | $0.1rT$, where $r$ is the KV cache budget ratio and $T$ is the prompt length |
| Key representation for cosine comparisons | Cached Post-RoPE keys |
| Attention-mass observation window | Final up to 64 context queries |
| Positional embedding | Fixed 16-dimensional sinusoidal embedding |
| MLP depth and activation | three layers with SiLU |
| Trainable parameters | 6.46M for Qwen3-8B |

**Training details.** We optimize $f_\theta$ using the objective in Equation 8, while keeping the backbone and the remaining AbsorbKV components fixed. The optimization settings are summarized in Table 5. Training uses 250 steps and takes approximately 8.5 seconds per step on our setup. The learned transformation is then used unchanged for both LongBench and RULER evaluation.

Table 5: Writer training configuration.

| Setting | Value |
|---|---|
| Training examples per eligible task | 100 |
| Optimizer and learning rate | AdamW, $10^{-3}$; weight decay 0 |
| Learning-rate schedule and warmup | Cosine decay with 10-step linear warmup |
| Training steps | 250 |
| Loss weights $\lambda_{\mathrm{KL}}$, $\lambda_{\mathrm{rel}}$ | 1.0, 5.0 |
| Query definition for attention output matching | Fixed Full-KV teacher queries |
| Random seeds | 151 |

## B Theoretical Motivation for AbsorbKV

Selection determines the retained keys through which queries access the compressed cache, whereas writing updates the values associated with those keys. We show how the retained support constrains attention output reconstruction and use this dependency to motivate selection–writing co-design.

**Setup and notation.** Fix a context, one layer and KV head, and $M \geq 1$ evaluation queries. Let $S$ contain $b \geq 1$ retained states, with at least one unmasked state for each query. The matrix $A_S \in \mathbb{R}^{M\times b}$ contains the attention weights over the retained keys, including positional encoding and masking. These weights are renormalized over $S$ after eviction. Let $V_S \in \mathbb{R}^{b\times d}$ contain the original retained values and $O_{\mathrm{full}} \in \mathbb{R}^{M\times d}$ the Full KV outputs for the same queries. A value update $\Delta V \in \mathbb{R}^{b\times d}$ produces reconstruction error

$$\mathcal{L}(S, \Delta V) = \|O_{\mathrm{full}} - A_S(V_S + \Delta V)\|_F^2. \tag{9}$$

Direct eviction corresponds to $\Delta V = 0$. The update is shared across all queries and leaves the retained keys unchanged. We use the same evaluation queries for every support as offline probes, rather than inputs to compression.

**Retained keys constrain reconstruction.** Each column of $A_S$ describes how the evaluation queries attend to one retained state. Value updates combine these attention patterns, so every column of the reconstructed output must lie in the column space of $A_S$.

**Proposition 1** (Best reconstruction under fixed retained keys)**.** *Let $P_S = A_S A_S^\dagger$ be the orthogonal projector onto the column space of $A_S$, where $A_S^\dagger$ is its Moore–Penrose pseudoinverse. Then*

$$\mathcal{L}^*(S) = \min_{\Delta V\in\mathbb{R}^{b\times d}} \mathcal{L}(S, \Delta V) = \|(I_M - P_S)O_{\mathrm{full}}\|_F^2, \tag{10}$$

*where $I_M$ is the $M \times M$ identity matrix. A minimizer is $\Delta V^\star = A_S^\dagger O_{\text{full}} - V_S$. Exact reconstruction is possible if and only if every column of $O_{\text{full}}$ lies in the column space of $A_S$.*

*Proof.* The columns of $(I_M - P_S)O_{\text{full}}$ are orthogonal to those of $P_S O_{\text{full}} - A_S(V_S + \Delta V)$. Therefore,

$$\mathcal{L}(S, \Delta V) = \|(I_M - P_S)O_{\text{full}}\|_F^2 + \|P_S O_{\text{full}} - A_S(V_S + \Delta V)\|_F^2. \tag{11}$$

The first term is independent of $\Delta V$, and the second is nonnegative. Choosing $\Delta V^\star = A_S^\dagger O_{\text{full}} - V_S$ makes the second term zero because $A_S A_S^\dagger O_{\text{full}} = P_S O_{\text{full}}$. The minimum is zero exactly when every column of $O_{\text{full}}$ belongs to the column space of $A_S$. These projection identities hold for rank deficient $A_S$ as well. □

**Implications for co-design.** Proposition 1 minimizes reconstruction error over unrestricted value updates, yielding a lower bound for any writer that preserves the retained keys. This bound is determined by the component of the Full KV output outside the column space of $A_S$. Selection should therefore consider whether evicted information can be recovered through the retained support, alongside the importance of the states preserved directly. AbsorbKV follows this principle through a Hybrid Selector that retains states that are important or difficult to absorb. Given this support, Adaptive Absorption routes compatible sources to retained carriers and transforms their values for retrieval through the carrier keys.

# C Additional Experimental Results

This appendix supplements the main results with hyperparameter analyses, comparisons with additional eviction baselines, long sequence modeling perplexity, and multi-key NIAH retrieval. We also report LongBench results at 25% and 50% KV cache budgets and detailed per-task RULER scores. Unless otherwise specified, the LongBench evaluations use the same settings as in the ablation study.

## C.1 Hyperparameter Sensitivity

Table 6: Hyperparameter sensitivity on LongBench tasks with Qwen3-4B and a 10% KV cache budget. Admitted sources are measured as a percentage of evicted KV items. † marks the default configuration.

(a) Selector weight, fixed $\tau = 0.7$

| $\lambda$ | Avg. score | Admitted sources |
|---|---|---|
| 0 | 43.63 | 45.18% |
| $0.25^\dagger$ | **47.58** | 46.95% |
| 0.5 | 47.23 | 43.56% |
| 0.75 | 45.29 | 40.47% |
| 1 | 41.86 | 38.65% |

(b) Routing threshold, fixed $\lambda = 0.25$

| $\tau$ | Avg. score | Admitted sources |
|---|---|---|
| 0 | 47.71 | 100.00% |
| 0.3 | **48.04** | 99.99% |
| 0.5 | 47.57 | 95.47% |
| $0.7^\dagger$ | 47.58 | 46.95% |
| 0.9 | 44.51 | 10.77% |

We evaluate sensitivity to the Hybrid Selector weight $\lambda$ and routing threshold $\tau$ using Qwen3-4B with a 10% KV cache budget. Table 6(a) varies $\lambda$ with $\tau = 0.7$, while Table 6(b) varies $\tau$ with $\lambda = 0.25$. The writer and remaining settings are fixed. Alongside the average task score, we report the fraction of evicted sources admitted for subsequent writing.

The selector benefits from balancing importance and absorbability rather than relying on either signal alone. The best selection weight is $\lambda = 0.25$, yielding 47.58 compared with 43.63 for importance alone and 41.86 for absorbability alone. In the routing sweep, scores remain between 47.57 and 48.04 as $\tau$ increases from 0 to 0.7, while the admitted fraction falls from 100% to 46.95%. Although $\tau = 0.3$ achieves the highest score, we use $\tau = 0.7$ by default because it retains nearly the same performance while admitting less than half as many sources, substantially reducing the merge overhead. Increasing $\tau$ further to 0.9 admits only 10.77% of sources and reduces the score to 44.51, indicating that overly restrictive routing discards useful source information.

### C.2 Comparisons with Additional Baselines

We compare AbsorbKV with four additional KV cache eviction baselines, including SnapKV (Li et al., 2024b), PyramidKV (Cai et al., 2024), TOVA (Oren et al., 2024), and Expected Attention (Devoto et al., 2025). We use Qwen3-8B with 10% and 25% KV cache budgets. In addition to LongBench, we also evaluate all 13 RULER tasks at a 4K context length. Dataset descriptions and scoring details are provided in Appendix A.1. We use the KVPress (Devoto et al., 2025) implementations of the additional baselines, and all methods compress the context cache before observing the question.

Table 7 shows that AbsorbKV achieves the highest average score on both benchmarks at both budgets. CUR remains the strongest KV cache eviction baseline on LongBench, while TOVA performs best among the eviction methods on RULER-4K. AbsorbKV improves over CUR by 4.73 and 2.08 on LongBench at 10% and 25% budgets, respectively, and exceeds TOVA by 10.34 and 11.18 on RULER-4K. The task level results reveal different strengths among the additional baselines. At a 10% budget, TOVA performs well on passage retrieval but poorly on TREC, whereas Expected Attention exhibits the opposite pattern. AbsorbKV achieves the best performance across these tasks, demonstrating its advantage beyond comparisons with CUR and existing merging methods.

Table 7: Comparison with additional KV cache eviction baselines on Qwen3-8B.

| Budget | Method | LongBench | | | | | | | RULER-4K |
|---|---|---|---|---|---|---|---|---|---|
| | | Qasper | 2WikiMQA | QMSum | TREC | PR-en | RB-P | Avg. | Avg. |
| — | Full KV | 44.27 | 48.41 | 24.67 | 41.50 | 90.85 | 60.05 | 51.63 | 94.56 |
| 10% | SnapKV | 20.95 | 29.85 | 19.05 | 31.25 | 47.29 | 62.40 | 35.13 | 15.07 |
| | PyramidKV | 20.79 | 29.60 | 19.00 | 32.00 | 46.13 | 62.06 | 34.93 | 15.11 |
| | TOVA | 14.42 | 34.75 | 19.30 | 9.50 | 88.18 | 60.62 | 37.80 | 24.15 |
| | Expected Attention | 27.71 | 32.26 | 20.96 | 62.00 | 16.00 | 61.09 | 36.67 | 10.94 |
| | CUR | 29.45 | 38.97 | 21.68 | 29.00 | 75.68 | 63.07 | 42.97 | 13.53 |
| | AbsorbKV | 31.45 | 38.96 | 22.31 | 48.75 | 82.89 | 61.86 | **47.70** | **34.49** |
| 25% | SnapKV | 29.76 | 41.17 | 21.26 | 37.25 | 88.50 | 61.34 | 46.55 | 27.54 |
| | PyramidKV | 26.64 | 35.61 | 20.71 | 31.25 | 82.29 | 60.85 | 42.89 | 27.41 |
| | TOVA | 27.21 | 41.94 | 21.04 | 20.00 | 90.77 | 60.57 | 43.59 | 60.64 |
| | Expected Attention | 34.73 | 38.48 | 22.79 | 69.00 | 47.63 | 61.30 | 45.66 | 21.38 |
| | CUR | 39.59 | 44.57 | 22.53 | 62.00 | 91.29 | 64.27 | 54.04 | 41.30 |
| | AbsorbKV | 39.55 | 46.66 | 23.10 | 70.25 | 94.04 | 63.13 | **56.12** | **71.82** |

### C.3 Reference Perplexity during Decoding

To evaluate long sequence language modeling under compression, we measure reference perplexity during decoding on Qwen3-8B with a 10% KV cache budget. We use the three LongBench summarization tasks, where generation requires information from the input to remain useful throughout decoding. For each task, we select 50 examples and generate up to 400 tokens, with prompt lengths ranging from 2K to 52K. At each decoding step, all methods are conditioned on the same reference prefix, isolating the effect of the compressed cache on next-token prediction. Figure 7 reports the resulting perplexity curves for each method, and AbsorbKV achieves lower reference perplexity than the baselines throughout decoding. These results show that AbsorbKV better preserves the long range information dependencies even under a limited KV cache budget.

### C.4 Multi-Key Needle-in-a-Haystack Retrieval

We evaluate retrieval across input lengths and needle positions using the more challenging multi-key Needle-in-a-Haystack (NIAH) setting with Qwen3-4B. Each context contains multiple key needles, and the model must retrieve the value associated with the queried key. As shown in Figure 8, AbsorbKV preserves retrieval across a much broader range of input lengths and needle depths than CUR and AsymKV. At 50%, the accuracy of AbsorbKV is nearly perfect throughout the settings, including intermediate depths where AsymKV frequently fails. At 10%, retrieval remains difficult

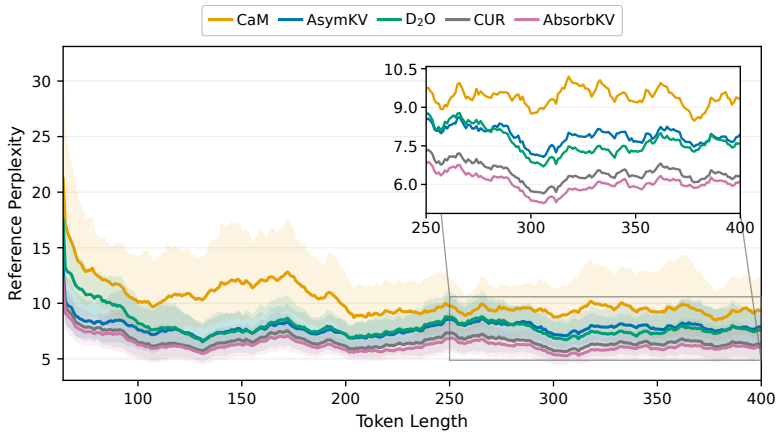



Figure 7: Reference perplexity during summarization on Qwen3-8B with a 10% KV cache budget. We evaluate 50 examples from each of the three LongBench summarization tasks and score up to 400 output tokens. Lower perplexity indicates more faithful next-token prediction under the compressed cache.

for all three methods, but AbsorbKV still recovers more target needles. These results show that AbsorbKV's retrieval gains extend across needle positions rather than being confined to the beginning or end of the context.

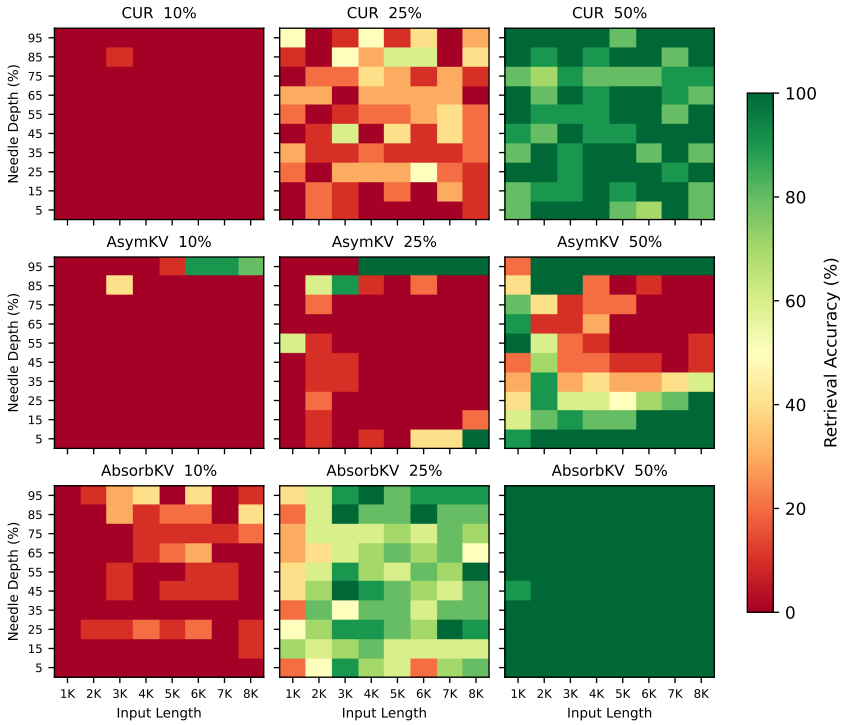



Figure 8: Multi-key NIAH retrieval accuracy on Qwen3-4B.

### C.5 LongBench Results at Additional Cache Budgets

Table 8 reports results on the 16 English LongBench tasks using Qwen3-8B with 25% and 50% KV cache budgets. AbsorbKV achieves the highest average score at both budgets, outperforming

CUR by 1.10 and 0.35, respectively. Together with the 10% result in Table 1, these results show that AbsorbKV provides its largest gains under aggressive compression, with the advantage gradually narrowing as more KV states are retained. This trend is consistent with our motivation, since the choice of retained support becomes more consequential when fewer states are available to absorb evicted information.

Table 8: LongBench results on Qwen3-8B with 25% and 50% KV cache budgets. Avg. is the mean score over the 16 English tasks.

| | | Single-doc QA | | | Multi-doc QA | | | Summarization | | | Few-shot Learning | | | Synthetic | | Code | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Budget** | **Method** | NrtvQA | Qasper | MF-en | HotpotQA | 2WikiMQA | MuSiQue | GovReport | QMSum | MultiNews | TREC | TriviaQA | SAMSum | PCount | PR-en | LCC | RB-P | **Avg.** |
| — | Full KV | 26.66 | 44.27 | 55.71 | 62.80 | 48.41 | 35.19 | 33.44 | 24.67 | 24.92 | 41.50 | 90.28 | 40.00 | 9.00 | 90.85 | 64.90 | 60.05 | 47.04 |
| 25% | CaM | 3.06 | 14.19 | 24.25 | 8.09 | 15.52 | 7.82 | 16.74 | 12.98 | 23.18 | 12.38 | 82.75 | 31.46 | 5.72 | 89.03 | 33.21 | 58.60 | 27.44 |
| | $D_2O$ | 23.97 | 33.41 | 42.51 | 47.74 | 42.70 | 29.06 | 31.60 | 22.16 | 23.40 | 58.00 | 80.76 | 37.87 | 11.00 | 76.00 | 50.78 | 61.84 | 42.05 |
| | AsymKV | 23.62 | 28.43 | 43.18 | 52.44 | 37.66 | 28.00 | 30.97 | 21.82 | 22.70 | 68.50 | 89.49 | 38.72 | 6.59 | 79.47 | 56.19 | 58.36 | 42.88 |
| | CUR | 28.34 | **39.59** | 44.13 | 56.22 | 44.57 | 34.10 | 32.20 | 22.53 | 23.50 | 62.00 | 88.28 | 37.78 | **13.88** | 91.29 | **62.09** | **64.27** | 46.55 |
| | **AbsorbKV** | **30.17** | 39.55 | **45.81** | **57.94** | **46.66** | **35.63** | **32.74** | **23.10** | **24.04** | **70.25** | **89.54** | **39.14** | 10.00 | **94.04** | 60.62 | 63.13 | **47.65** |
| 50% | CaM | 2.39 | 16.33 | 26.25 | 9.33 | 20.94 | 8.91 | 16.37 | 17.99 | 24.63 | 30.25 | 80.28 | 33.15 | 5.34 | 90.45 | 44.22 | 57.42 | 30.26 |
| | $D_2O$ | 26.49 | 40.05 | 48.60 | 59.38 | 44.75 | 29.98 | **34.12** | 23.31 | **25.08** | 65.00 | 85.40 | 40.54 | 6.50 | **97.75** | 59.94 | 61.01 | 46.75 |
| | AsymKV | 26.20 | 37.55 | 49.09 | 58.02 | 44.14 | 32.52 | 32.29 | 22.97 | 24.37 | 67.00 | 88.51 | 39.53 | 7.00 | 90.02 | 63.76 | 58.70 | 46.35 |
| | CUR | 28.02 | 42.61 | 49.45 | **63.76** | 47.78 | 34.21 | 33.39 | 23.48 | 24.81 | 70.50 | **88.87** | 39.73 | **11.50** | 95.96 | **64.67** | 61.58 | 48.77 |
| | **AbsorbKV** | **29.07** | **44.16** | **51.81** | 62.13 | **50.29** | **36.73** | 33.29 | **23.70** | 24.56 | **72.00** | 88.84 | **41.11** | 9.00 | 94.25 | 63.22 | **61.75** | **49.12** |

### C.6 Per-Task RULER Results

Figure 9 complements the average scores in Figure 4 with a comparison across all 13 RULER tasks on Qwen3-8B at a 16K context length. The performance gains of AbsorbKV are most evident in retrieval, variable tracking, and word extraction under small cache budgets. At a 25% budget, AbsorbKV scores 86.0 on multi-value retrieval and 82.0 on multi-query retrieval, compared with 54.5 and 37.5 for $D_2O$, the strongest baseline on these two tasks. AbsorbKV also approaches Full KV performance on variable tracking and word extraction at this budget. The single-key and multi-key retrieval variants require more retained states, but AbsorbKV recovers more retrieval accuracy than the baselines at 50% and matches Full KV at 75% KV cache budget. In addition, we report AbsorbKV's per-task scores across all settings in Tables 9–11.

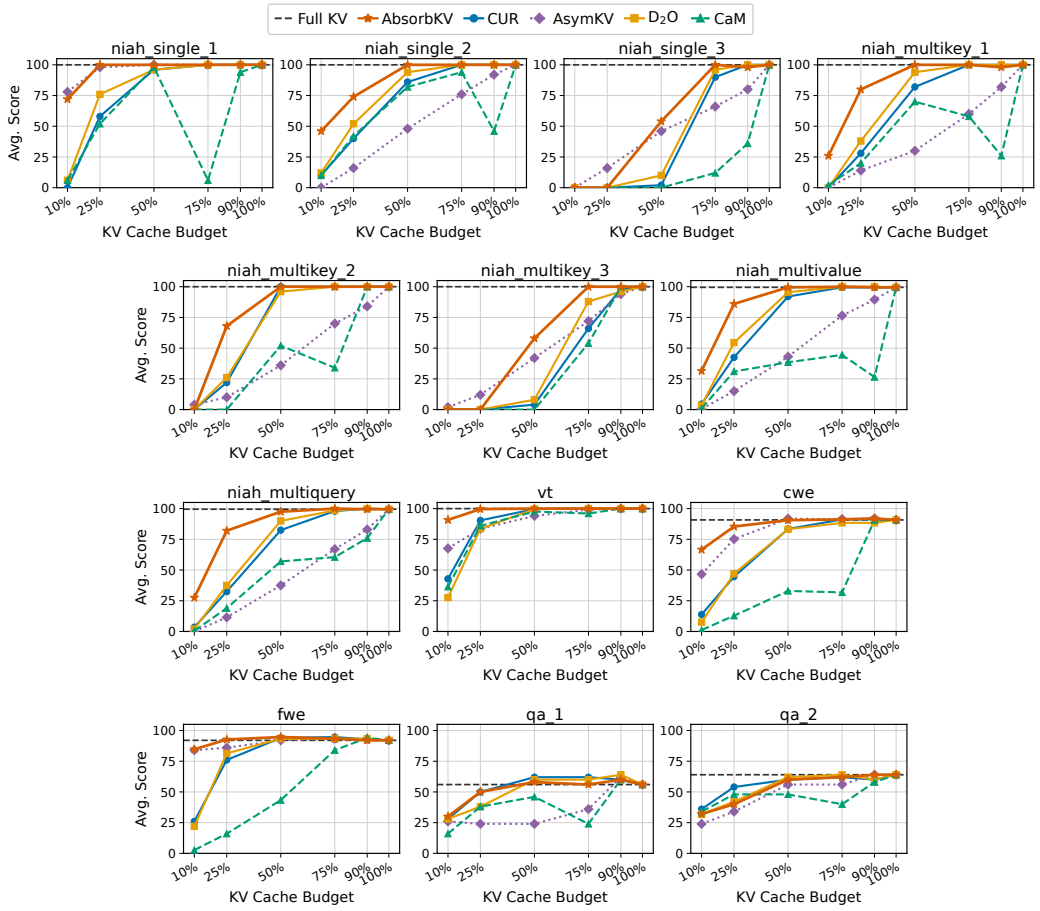



Figure 9: Per-task RULER results on Qwen3-8B with 16K contexts across KV-cache budgets. Each panel corresponds to one task, and the dashed horizontal line marks its Full-KV score.

Table 9: Per-task RULER scores for Qwen3-8B. Full KV is the uncompressed reference; the five remaining columns report AbsorbKV at the indicated context-prefix budgets. Each score averages 50 examples.

| | | AbsorbKV | | | | |
|---|---|---|---|---|---|---|
| Task | Full KV | 10% | 25% | 50% | 75% | 90% |
| Context length 4K | | | | | | |
| Single-key 1 | 100.00 | 42.00 | 96.00 | 100.00 | 100.00 | 100.00 |
| Single-key 2 | 100.00 | 22.00 | 90.00 | 100.00 | 100.00 | 100.00 |
| Single-key 3 | 100.00 | 0.00 | 0.00 | 68.00 | 100.00 | 100.00 |
| Multi-key 1 | 100.00 | 18.00 | 96.00 | 100.00 | 100.00 | 100.00 |
| Multi-key 2 | 100.00 | 2.00 | 80.00 | 100.00 | 100.00 | 100.00 |
| Multi-key 3 | 100.00 | 0.00 | 0.00 | 56.00 | 100.00 | 100.00 |
| MV | 100.00 | 24.50 | 94.50 | 100.00 | 100.00 | 100.00 |
| MQ | 100.00 | 13.50 | 84.50 | 100.00 | 100.00 | 100.00 |
| VT | 100.00 | 88.40 | 100.00 | 100.00 | 100.00 | 100.00 |
| CWE | 100.00 | 84.60 | 96.00 | 99.60 | 100.00 | 100.00 |
| FWE | 85.33 | 73.33 | 86.67 | 86.67 | 84.00 | 86.67 |
| QA-1 | 74.00 | 44.00 | 58.00 | 68.00 | 74.00 | 76.00 |
| QA-2 | 70.00 | 36.00 | 52.00 | 66.00 | 66.00 | 66.00 |
| Avg. | 94.56 | 34.49 | 71.82 | 88.02 | 94.15 | 94.51 |
| Context length 8K | | | | | | |
| Single-key 1 | 100.00 | 68.00 | 98.00 | 100.00 | 100.00 | 100.00 |
| Single-key 2 | 100.00 | 26.00 | 80.00 | 98.00 | 100.00 | 100.00 |
| Single-key 3 | 100.00 | 0.00 | 0.00 | 58.00 | 100.00 | 100.00 |
| Multi-key 1 | 100.00 | 32.00 | 82.00 | 100.00 | 100.00 | 100.00 |
| Multi-key 2 | 100.00 | 0.00 | 68.00 | 100.00 | 100.00 | 100.00 |
| Multi-key 3 | 98.00 | 0.00 | 0.00 | 50.00 | 98.00 | 98.00 |
| MV | 100.00 | 26.50 | 87.00 | 100.00 | 100.00 | 100.00 |
| MQ | 100.00 | 21.00 | 83.50 | 98.50 | 100.00 | 100.00 |
| VT | 100.00 | 87.20 | 99.20 | 100.00 | 100.00 | 100.00 |
| CWE | 98.80 | 72.20 | 86.80 | 96.80 | 98.40 | 98.60 |
| FWE | 86.00 | 80.00 | 83.33 | 83.33 | 85.33 | 86.00 |
| QA-1 | 68.00 | 28.00 | 50.00 | 62.00 | 68.00 | 72.00 |
| QA-2 | 68.00 | 32.00 | 48.00 | 64.00 | 68.00 | 70.00 |
| Avg. | 93.75 | 36.38 | 66.60 | 85.43 | 93.67 | 94.20 |
| Context length 16K | | | | | | |
| Single-key 1 | 100.00 | 72.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| Single-key 2 | 100.00 | 46.00 | 74.00 | 100.00 | 100.00 | 100.00 |
| Single-key 3 | 100.00 | 0.00 | 0.00 | 54.00 | 100.00 | 98.00 |
| Multi-key 1 | 100.00 | 26.00 | 80.00 | 100.00 | 100.00 | 98.00 |
| Multi-key 2 | 100.00 | 0.00 | 68.00 | 100.00 | 100.00 | 100.00 |
| Multi-key 3 | 100.00 | 0.00 | 0.00 | 58.00 | 100.00 | 100.00 |
| MV | 99.50 | 31.50 | 86.00 | 99.50 | 100.00 | 99.50 |
| MQ | 99.50 | 27.50 | 82.00 | 97.50 | 100.00 | 99.50 |
| VT | 100.00 | 90.80 | 99.60 | 100.00 | 100.00 | 100.00 |
| CWE | 90.80 | 66.60 | 85.40 | 90.60 | 91.20 | 92.00 |
| FWE | 92.00 | 84.67 | 92.67 | 94.67 | 93.33 | 92.00 |
| QA-1 | 56.00 | 30.00 | 50.00 | 58.00 | 56.00 | 60.00 |
| QA-2 | 64.00 | 32.00 | 40.00 | 60.00 | 62.00 | 64.00 |
| Avg. | 92.45 | 39.01 | 65.97 | 85.56 | 92.50 | 92.54 |

Table 10: Per-task RULER scores for Llama-3.1-8B-Instruct. Full KV is the uncompressed reference; the five remaining columns report AbsorbKV at the indicated context-prefix budgets. Each score averages 50 examples.

| | | AbsorbKV | | | | |
|---|---|---|---|---|---|---|
| Task | Full KV | 10% | 25% | 50% | 75% | 90% |
| Context length 4K | | | | | | |
| Single-key 1 | 100.00 | 94.00 | 98.00 | 98.00 | 98.00 | 100.00 |
| Single-key 2 | 100.00 | 76.00 | 98.00 | 100.00 | 100.00 | 100.00 |
| Single-key 3 | 100.00 | 0.00 | 8.00 | 68.00 | 100.00 | 100.00 |
| Multi-key 1 | 100.00 | 58.00 | 98.00 | 98.00 | 100.00 | 100.00 |
| Multi-key 2 | 100.00 | 38.00 | 90.00 | 100.00 | 100.00 | 100.00 |
| Multi-key 3 | 100.00 | 0.00 | 10.00 | 76.00 | 98.00 | 100.00 |
| MV | 100.00 | 71.00 | 95.50 | 99.50 | 100.00 | 99.50 |
| MQ | 100.00 | 62.00 | 96.00 | 99.00 | 100.00 | 100.00 |
| VT | 100.00 | 65.60 | 92.00 | 97.20 | 99.60 | 100.00 |
| CWE | 100.00 | 60.80 | 91.20 | 99.80 | 100.00 | 100.00 |
| FWE | 90.00 | 78.67 | 82.00 | 87.33 | 90.00 | 90.00 |
| QA-1 | 80.00 | 38.00 | 62.00 | 78.00 | 80.00 | 80.00 |
| QA-2 | 72.00 | 36.00 | 62.00 | 70.00 | 72.00 | 72.00 |
| Avg. | 95.54 | 52.16 | 75.59 | 90.06 | 95.20 | 95.50 |
| Context length 8K | | | | | | |
| Single-key 1 | 100.00 | 88.00 | 98.00 | 100.00 | 100.00 | 100.00 |
| Single-key 2 | 100.00 | 92.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| Single-key 3 | 100.00 | 0.00 | 2.00 | 44.00 | 100.00 | 100.00 |
| Multi-key 1 | 100.00 | 62.00 | 98.00 | 100.00 | 100.00 | 100.00 |
| Multi-key 2 | 100.00 | 24.00 | 80.00 | 98.00 | 100.00 | 100.00 |
| Multi-key 3 | 100.00 | 0.00 | 12.00 | 84.00 | 100.00 | 100.00 |
| MV | 99.00 | 74.00 | 89.00 | 97.00 | 99.00 | 99.00 |
| MQ | 99.50 | 75.50 | 99.00 | 99.00 | 99.00 | 99.00 |
| VT | 100.00 | 67.20 | 83.20 | 96.40 | 98.80 | 100.00 |
| CWE | 99.00 | 28.80 | 61.20 | 91.60 | 98.40 | 98.80 |
| FWE | 94.67 | 80.00 | 90.00 | 91.33 | 94.67 | 94.00 |
| QA-1 | 72.00 | 36.00 | 58.00 | 70.00 | 78.00 | 76.00 |
| QA-2 | 78.00 | 46.00 | 54.00 | 72.00 | 72.00 | 72.00 |
| Avg. | 95.55 | 51.81 | 71.11 | 87.95 | 95.37 | 95.29 |
| Context length 16K | | | | | | |
| Single-key 1 | 100.00 | 98.00 | 98.00 | 100.00 | 100.00 | 100.00 |
| Single-key 2 | 100.00 | 78.00 | 98.00 | 100.00 | 100.00 | 100.00 |
| Single-key 3 | 100.00 | 0.00 | 12.00 | 40.00 | 92.00 | 100.00 |
| Multi-key 1 | 100.00 | 60.00 | 96.00 | 98.00 | 100.00 | 100.00 |
| Multi-key 2 | 100.00 | 30.00 | 66.00 | 94.00 | 100.00 | 100.00 |
| Multi-key 3 | 100.00 | 0.00 | 8.00 | 72.00 | 100.00 | 100.00 |
| MV | 99.50 | 73.00 | 88.50 | 95.50 | 97.50 | 99.50 |
| MQ | 98.50 | 69.00 | 97.00 | 97.00 | 99.00 | 98.50 |
| VT | 98.80 | 48.00 | 72.80 | 90.40 | 98.40 | 98.80 |
| CWE | 91.60 | 20.60 | 40.00 | 75.20 | 93.00 | 93.80 |
| FWE | 96.67 | 86.67 | 91.33 | 94.67 | 96.67 | 96.67 |
| QA-1 | 74.00 | 30.00 | 54.00 | 70.00 | 72.00 | 76.00 |
| QA-2 | 54.00 | 36.00 | 48.00 | 54.00 | 56.00 | 52.00 |
| Avg. | 93.31 | 48.41 | 66.89 | 83.14 | 92.66 | 93.48 |

Table 11: Per-task RULER scores for Qwen3-4B. Full KV is the uncompressed reference; the five remaining columns report AbsorbKV at the indicated context-prefix budgets. Each score averages 50 examples.

| | | AbsorbKV | | | | |
|---|---|---|---|---|---|---|
| Task | Full KV | 10% | 25% | 50% | 75% | 90% |
| Context length 4K | | | | | | |
| Single-key 1 | 100.00 | 66.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| Single-key 2 | 100.00 | 6.00 | 80.00 | 100.00 | 100.00 | 100.00 |
| Single-key 3 | 100.00 | 0.00 | 0.00 | 66.00 | 98.00 | 100.00 |
| Multi-key 1 | 100.00 | 6.00 | 74.00 | 100.00 | 100.00 | 100.00 |
| Multi-key 2 | 100.00 | 0.00 | 62.00 | 98.00 | 100.00 | 100.00 |
| Multi-key 3 | 100.00 | 0.00 | 0.00 | 62.00 | 100.00 | 100.00 |
| MV | 100.00 | 9.00 | 86.50 | 97.50 | 98.00 | 100.00 |
| MQ | 100.00 | 7.50 | 76.00 | 100.00 | 100.00 | 100.00 |
| VT | 100.00 | 78.40 | 100.00 | 100.00 | 100.00 | 100.00 |
| CWE | 97.00 | 82.80 | 90.60 | 94.80 | 97.40 | 97.00 |
| FWE | 70.67 | 60.00 | 67.33 | 68.67 | 70.00 | 70.67 |
| QA-1 | 72.00 | 32.00 | 48.00 | 68.00 | 74.00 | 76.00 |
| QA-2 | 66.00 | 36.00 | 52.00 | 60.00 | 66.00 | 68.00 |
| Avg. | 92.74 | 29.52 | 64.34 | 85.77 | 92.57 | 93.21 |
| Context length 8K | | | | | | |
| Single-key 1 | 100.00 | 38.00 | 90.00 | 100.00 | 100.00 | 100.00 |
| Single-key 2 | 100.00 | 8.00 | 66.00 | 100.00 | 100.00 | 100.00 |
| Single-key 3 | 100.00 | 0.00 | 0.00 | 52.00 | 100.00 | 100.00 |
| Multi-key 1 | 100.00 | 16.00 | 66.00 | 98.00 | 100.00 | 100.00 |
| Multi-key 2 | 100.00 | 0.00 | 58.00 | 100.00 | 100.00 | 100.00 |
| Multi-key 3 | 100.00 | 0.00 | 0.00 | 52.00 | 98.00 | 100.00 |
| MV | 100.00 | 11.00 | 84.00 | 99.00 | 99.50 | 100.00 |
| MQ | 100.00 | 13.00 | 73.50 | 99.50 | 100.00 | 100.00 |
| VT | 100.00 | 80.80 | 97.20 | 100.00 | 100.00 | 100.00 |
| CWE | 92.40 | 65.40 | 75.00 | 82.20 | 91.60 | 92.40 |
| FWE | 89.33 | 83.33 | 90.00 | 87.33 | 88.00 | 89.33 |
| QA-1 | 64.00 | 24.00 | 46.00 | 58.00 | 64.00 | 66.00 |
| QA-2 | 64.00 | 34.00 | 44.00 | 52.00 | 60.00 | 62.00 |
| Avg. | 93.06 | 28.73 | 60.75 | 83.08 | 92.39 | 93.06 |
| Context length 16K | | | | | | |
| Single-key 1 | 100.00 | 52.00 | 98.00 | 100.00 | 100.00 | 100.00 |
| Single-key 2 | 100.00 | 30.00 | 80.00 | 100.00 | 100.00 | 100.00 |
| Single-key 3 | 100.00 | 0.00 | 2.00 | 62.00 | 100.00 | 100.00 |
| Multi-key 1 | 100.00 | 8.00 | 68.00 | 96.00 | 100.00 | 100.00 |
| Multi-key 2 | 100.00 | 0.00 | 60.00 | 100.00 | 100.00 | 100.00 |
| Multi-key 3 | 98.00 | 0.00 | 0.00 | 42.00 | 98.00 | 98.00 |
| MV | 100.00 | 15.00 | 77.00 | 98.00 | 100.00 | 100.00 |
| MQ | 100.00 | 13.50 | 65.50 | 99.00 | 100.00 | 100.00 |
| VT | 100.00 | 42.00 | 91.20 | 99.60 | 100.00 | 100.00 |
| CWE | 83.20 | 44.00 | 66.00 | 70.40 | 84.00 | 83.00 |
| FWE | 94.00 | 86.00 | 93.33 | 93.33 | 92.67 | 94.00 |
| QA-1 | 58.00 | 32.00 | 36.00 | 52.00 | 54.00 | 58.00 |
| QA-2 | 60.00 | 30.00 | 40.00 | 56.00 | 60.00 | 60.00 |
| Avg. | 91.78 | 27.12 | 59.77 | 82.18 | 91.44 | 91.77 |