SGDKV: Summarization Guided KV Cache Compression

Anonymous ACL submission

Abstract

The deployment of large language models (LLMs) in long-context scenarios faces critical computational bottlenecks due to the quadratic scaling of attention mechanisms and linearly growth of key-value (KV) cache memory requirements. While existing KV cache optimization methods focus on token-level eviction strategies, they heavily rely on retrieval tasks based on the simple one-step or multi-step pattern matching to determine the importance of heads, which overlooks the fundamental insight that some attention heads can extract high-level information. We introduce SGD-KV (Summarization-Guided KV Cache compression), a novel head-aware cache management framework that identifies and prioritizes attention heads specialized in hierarchical information aggregation, which we term "summarization heads", for optimal KV cache allocation. Our approach employs a principled methodology to identify summarization heads through a chunk-summarization diagnostic task that requires semantic aggregation across multiple text segments, revealing that these heads exhibit remarkable consistency across datasets and model architectures. Extensive experiments on Qwen2.5-7B-1M and Qwen3-32B models across diverse long-context benchmarks, including OpenAI MRCR, BABILong, and ETHIC, demonstrate that SGD-KV achieves state-of-the-art performance while reducing KV cache memory usage by up to 75%. Our results validate that understanding and leveraging attention head specialization represents a promising direction for scaling LLMs to even longer contexts, with summarization heads playing a crucial role in complex reasoning and document understanding tasks.

1 Introduction

The remarkable capabilities of Large Language Models (LLMs) have spurred a push towards ever-longer context windows, with models like Qwen2.5-1M (Yang et al., 2025) and Gemini 2.5 (Comanici et al., 2025) now processing millions of tokens. However, a fundamental obstacle remains: the prohibitive memory cost of the Key-Value (KV) cache, which scales linearly with context length and can consume hundreds of gigabytes of GPU memory (Vaswani et al., 2023). This bottleneck severely limits the practical deployment of LLMs in long-context applications.

Recent advances in KV cache optimization have primarily focused on token-level eviction strategies, such as retaining attention sinks and recent tokens (Xiao et al., 2024), or identifying important tokens through attention score analysis (Zhang et al., 2023). However, these approaches treat all attention heads uniformly, overlooking the fact that different heads specialize in distinct cognitive functions (Clark et al., 2019; Zheng et al., 2024b). While some recent work has begun exploring head-level cache allocation (Fu et al., 2024; Feng et al., 2024), existing methods rely on simple retrieval-based metrics that fail to capture the sophisticated reasoning or summarization patterns required for complex long-context tasks. We observe that many real-world long-context applications, including document summarization, multi-turn dialogue, and complex reasoning, require models to perform hierarchical information aggregation rather than simple pattern matching. This process demands specialized attention mechanisms that can identify semantically salient content, maintain global coherence, and synthesize information across multiple levels of abstraction. We hypothesize that certain attention heads, which we term summarization heads, are specifically responsible for these higher-order cognitive capabilities.

In this paper, we introduce SGD-KV (Summarization-Guided KV Cache compression), a novel head-aware cache management framework that identifies and prioritizes summarization heads for optimal KV cache allocation. Our key

contributions are fourfold:

- We propose a principled methodology for identifying summarization scores for each attention head through a novel chunk-summarization task that requires models to perform semantic aggregation across multiple text segments. Further, we validate the generalization and stability of summarization score distributions across various datasets and model architectures.
- We extend existing head-level KV cache budget allocation algorithms with a water-filling inspired redistribution mechanism that prioritizes heads with higher summarization scores while preserving essential structural tokens (attention sinks and recent tokens).
- We demonstrate that SGD-KV achieves state-of-the-art performance across diverse long-context benchmarks, including OpenAI MRCR (OpenAI, 2025b), BABILong (Kuratov et al., 2024), and ETHIC (Lee et al., 2024), while reducing KV cache memory usage by up to 75%. Extensive experiments on Qwen2.5-7B-1M and Qwen3-32B models reveal that summarization heads exhibit remarkable consistency across datasets and tasks, validating our core hypothesis.
- We conduct comprehensive ablation studies examining the impact of different KV cache budgets and question-aware versus question-agnostic settings. These analyses not only validate the effectiveness of our approach but also illuminate promising directions for future research in attention head specialization.

2 Related Work

KV Cache Compression To mitigate the significant memory burden caused by storing an increasing number of KV caches during long-sequence inference, a direct and effective approach is to evict or merge redundant KV caches. Early methods applied uniform cache budgets across all heads and layers: H2O (Zhang et al., 2023), as a pioneering method, leverages the full cumulative attention weight matrix to determine token importance and selectively retains heavy hitters (tokens with high attention scores) along with recent tokens. Building upon this foundation, SnapKV (Li et al., 2024) improves efficiency by using only the last few rows of the attention matrix to identify important prefill tokens, while StreamingLLM (Xiao et al., 2024) (often referred to as attention sinks) maintains a fixed-size cache by preserving initial tokens and a sliding window of recent tokens. These approaches achieve substantial memory reductions but often trade off accuracy in complex reasoning tasks due to their static heuristics. Recent advances have introduced more sophisticated, adaptive allocation strategies to address these limitations. For instance, PyramidKV (Cai et al., 2024) assigns smaller cache budgets to higher layers based on observed attention matrix patterns, exploiting the hierarchical nature of transformers. AdaKV (Feng et al., 2024) dynamically allocates KV cache based on real-time attention scores, while Scissorhands (Liu et al., 2023) exploits system-level optimizations for efficient KV eviction, and Quest (Tang et al., 2024) introduces query-aware token selection for better context relevance. Despite these gains, challenges remain in balancing compression ratios with perplexity degradation in complex real tasks.

Attention Heads Mechanistic interpretability of attention heads in transformers has revealed specialized roles that evolve with model scale and training. Foundational works identified induction heads (Olsson et al., 2022), which copy and complete patterns by attending to prior tokens in smaller models, laying the groundwork for understanding in-context learning. Since Wu et al. (2025) identified the existence of retrieval heads using synthesized needle-in-haystack (NIAH) datasets, various specialized attention head types have been proposed. Tang et al. (2025) introduced echo heads and induction heads, which tend to attend to previous tokens that are identical or directly related to the current token. Similarly, Fu et al. (2024) incorporated induction processes into NIAH datasets to identify retrieval-reasoning (R2) heads that combine retrieval with logical inference. Broader surveys (Zheng et al., 2024b) categorize heads into retrieval, reasoning, and expression-polishing types, highlighting their contributions to capabilities like out-of-distribution generalization (Song et al., 2024) and multi-head synergies in long-context processing (Lu et al., 2024).

However, these approaches rely exclusively on pattern-matching questions to investigate attention head functionality. While effective for simple retrieval tasks, this methodology proves suboptimal for more sophisticated "retrieval" scenarios such as summarization or complex retrieval tasks requir-

ing analysis of very long contextual passages. As the field has progressed from basic retrieval to induction and reasoning capabilities (Olsson et al., 2022), researchers have discovered heads responsible for increasingly complex relationships between tokens. Building on this trajectory and addressing gaps in hierarchical synthesis (Zheng et al., 2024b), we introduce a summarization-based identification task that reveals heads capable of aggregating hidden semantic connections across multiple text segments, thereby uncovering attention mechanisms specialized for hierarchical information synthesis. This advances beyond prior interpretability frameworks by emphasizing real-world tasks over synthetic probes.

3 Method

3.1 The Definition of Summarization Scores & Heads

We define summarization heads as specialized attention mechanisms that orchestrate information concentration and semantic aggregation across token sequences. Unlike retrieval heads that primarily perform lexical pattern matching (Wu et al., 2025), summarization heads exhibit three key cognitive capabilities: (1) content-aware importance estimation for identifying semantically salient tokens, (2) hierarchical information aggregation that transcends local dependencies, and (3) coherent global context representation throughout the summarization process.

The identification of summarization heads presents unique challenges compared to retrieval heads. While retrieval heads can be identified through simple retrieval tasks by recording attention scores when models generate exact answers, summarization requires more sophisticated analysis. In summarization, models cannot rely on direct "passkey" retrieval but must understand semantic content and determine token importance at sentence or paragraph levels.

3.2 Chunk-Summarization Task Design

On the other hand, existing summarization datasets lack adequate length distribution for head identification, and aggressive compression in long-document summarization (e.g., book-to-sentence) loses critical details for realistic applications. We propose a chunk-summarization task that addresses these limitations. We first construct concatenated samples from multiple short summarization datasets:

$$\mathbf{D}_{\text{concat}} = \bigcup_{s \in \mathcal{S}} d_1^{(s)} \oplus d_2^{(s)} \oplus \cdots \oplus d_{k_s}^{(s)} \quad (1)$$

where $\mathcal{S} = \{\text{source dataset 1, source dataset 2, ...}\}$ represents source datasets, and $k_s \sim \mathcal{U}(n_{min}, n_{max})$ uniformly select the number of chunks between the predefined minimal and maximal number, ensuring length diversity. Each concatenated sample contains multiple semantically distinct chunks with recorded boundaries.

Given a concatenated sample, we prompt the model to: (1) identify the number of semantic chunks based on sentence relationships and consistency, and (2) generate keywords for each chunk in a structured format. This design forces models to perform hierarchical understanding rather than simple pattern matching.

3.3 Summarization Score Calculation

Similar to the retrieval heads, which only record the scores when the model answers successfully, we filter effective samples using Average Recall Score (ARS):

$$ARS = \frac{1}{c} \sum_{i=0}^{c} \frac{|\mathcal{U}_i \cap \hat{\mathcal{U}}_i|}{|\mathcal{U}_i|} \quad (2)$$

where c is the number of chunks, $\mathcal{U}_i$ represents uni-grams in the reference summary of chunk i, and $\hat{\mathcal{U}}_i$ represents uni-grams in generated keywords for chunk i. We retain samples with the correct number of chunks and ARS above empirically determined thresholds.

For each attention head h, we compute importance scores by measuring attention weights between generated keywords and their occurrences in corresponding input chunks:

$$I_h = \frac{1}{c} \sum_{m=0}^{c} \frac{1}{k_m} \sum_{j=0}^{k_m} \max_{n=0}^{t_{m,j}} A_h(p^o_{m,j}, p^i_{m,j,n}) \quad (3)$$

where k_m is the number of keywords for chunk m, $t_{m,j}$ is the occurrence count of keyword (m, j) in the original input chunk, $p^o_{m,j}$ is the output position of the keyword (m, j), and $p^i_{m,j,n}$ is the nth input position of the same keyword.

The max operation captures the strongest attention connection, indicating the head's ability to identify and aggregate salient information. This

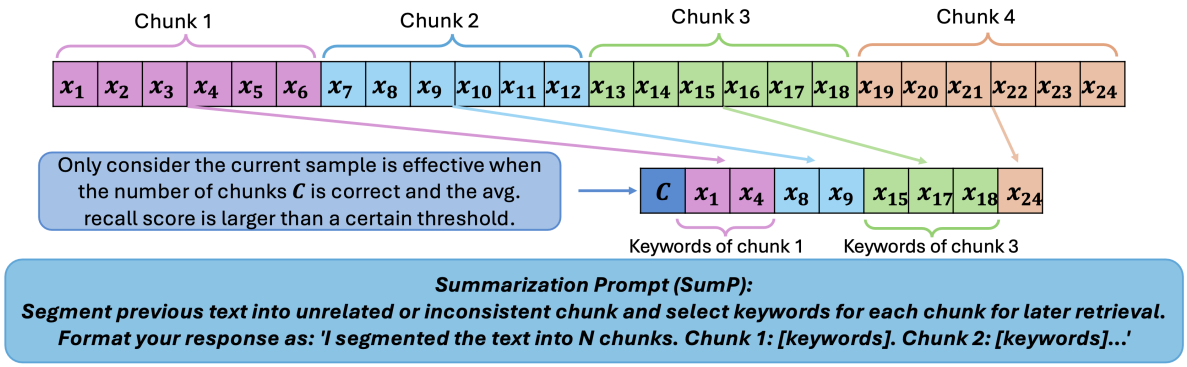

Figure 1: Illustration of calculating summarization scores for one sample and the designed summarization prompt.

formulation ensures that the heads with high importance scores can demonstrate genuine summarization capabilities by connecting output keywords to their most relevant input contexts while respecting chunk boundaries.

3.4 KV Cache Budget Allocation & Redistribution

The computed summarization scores directly guide our head-aware KV cache allocation strategy. Similar to HeadKV (Fu et al., 2024), we first normalize the scores across all heads across all layers, ensuring that the normalized scores sum to one across the entire model.

Then we always preserve a fixed budget for initial sink tokens (b_{sink}) and a sliding window of recent tokens (b_{window}) as the instructions of users are usually at the beginning or end of the whole input in practice.

The remaining cache budget for the middle, compressible portion of the sequence ($M = C - b_{\text{sink}} - b_{\text{window}}$, where C is total sequence length) is then allocated to each head h in proportion to its normalized summarization score S_h, at a given overall KV cache budget ratio R:

$$b_h = (S_h \cdot L \cdot H \cdot R) \cdot M + b_{\text{sink}} + b_{\text{window}} \quad (4)$$

where L is the number of layers, H is the number of key-value heads (accounting for Group Query Attention (Ainslie et al., 2023)).

Due to the task complexity requirements and the goal of minimal performance degradation, we employ a relatively large KV cache budget, which can result in $S_h \cdot L \cdot H > 1$ for high-importance heads. We address this through a water-filling inspired redistribution algorithm:

This approach prioritizes high-importance summarization heads while maintaining budget constraints, ensuring that critical attention mechanisms

Algorithm 1 Summarization-Aware Budget Redistribution

Input: Normalized head importance scores S_h for all heads h, M is the middle sequence length
Output: Redistributed budget allocation $\tilde{b}_h$ for all heads

1: Initialize b_h for all heads using Equation (6)
2: Compute excess budget: $E = \sum h : b_h > M(b_h - M)$
3: Cap over-allocated heads: $\tilde{b}_h \leftarrow \min(b_h, M)$ for all h
4: Sort remaining heads $\downarrow$ by S_h
5: Distribute E to top-scoring heads with available capacity

receive adequate KV cache allocation for effective long-context summarization. After the head-level cache budgets are allocated, we use the accumulative attention scores to select the important KV cache, similar to SnapKV (Li et al., 2024).

4 Experiments

4.1 Experimental Setup

Models. We evaluate our approach using Qwen2.5-7B-Instruct-1M (Yang et al., 2025) and Qwen3-32B (Team, 2025) as representative models for non-reasoning small LMs and reasoning-capable large LMs, respectively. However, we observed that the original Qwen2.5-7B-Instruct-1M checkpoint from HuggingFace exhibits suboptimal performance on multi-turn conversation tasks, particularly on the OpenAI Multi-Round Co-Reference Resolution (MRCR) benchmark (OpenAI, 2025b). To address this limitation, we perform additional fine-tuning using a diverse dataset comprising: 10K synthetic MRCR samples, 20K synthetic OpenAI GraphWalks samples (OpenAI, 2025a), and 25K

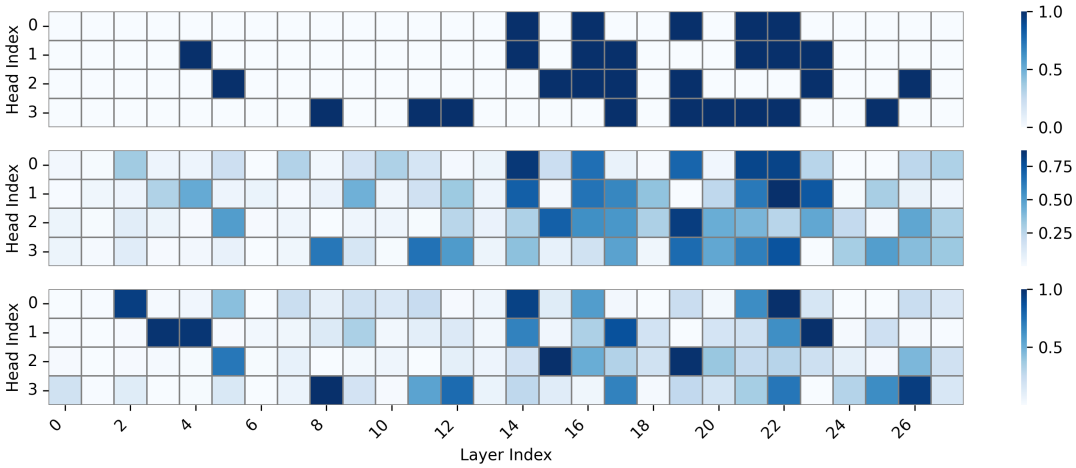

Figure 2: Visualization of the head scores of DuoAttention (top), HeadKV (center), and ours (bottom) in Qwen2.5-7B-Instruct-1M. Y axis is the index of the heads, 4 in total, and the X axis is the index of the layers, 28 in total.

BABILong fine-tuning samples (Kuratov et al., 2024). To prevent overfitting, we incorporate 10K samples from the Gutenberg dataset (Lahiri, 2014) and 10K samples from the Llama-Nemotron-Post-Training-Dataset-v1.1 (Bercovich et al., 2025) for regularization. Detailed fine-tuning configurations are provided in the Appendix A.1.

Baselines. We compare SGD-KV against four strong baselines representing different approaches to KV cache optimization: (i) DuoAttention (Xiao et al., 2025) finetunes a gate function to binary classify heads into retrieval heads (receiving full KV cache) and streaming heads (retaining only recent tokens and attention sink tokens). Since we do not finetune LLMs on retrieval datasets, we use R2 scores to binary partition heads based on available KV cache budget. (ii) AdaKV (Feng et al., 2024) pioneered head-level KV cache budget allocation by analyzing top-K attention values across heads during inference. (iii) HeadKV (Fu et al., 2024) introduced retrieval-reasoning heads, computing offline importance scores for each head and allocating KV cache budget proportionally based on these scores. (iv) MInference (Jiang et al., 2024) determines optimal attention patterns for each head offline and dynamically constructs sparse indices based on assigned patterns during inference. For fair comparison with DuoAttention, we adapt their binary classification approach by using R2 scores to partition heads into retrieval and streaming categories according to our KV cache budget constraints, rather than performing finetuning. For the MInference, since we directly use the VLLM (Kwon et al., 2023) to get the results, it always integrates with the Dual Chunk Attention (DCA) (An et al., 2024)

Configuration. For all baseline methods, we retain the first 1024 tokens as the attention sink, while maintaining a context window of 1024 recent tokens for the Qwen2.5-7B-1M model. For the larger Qwen3-32B model, both the sink size and window size are reduced to 128 tokens. All KV cache allocations occur before repeating the KV values, meaning that the KV cache memory savings are built upon the GQA. We also found that if we rerun the identification process for the models after SFT, the HeadKV method with the new configuration got worse performance. Therefore, we used the configuration of the original checkpoint for HeadKV, DuoAttetnion, and our method.

4.2 Visualization of Different Types of Heads

As shown in Fig. 2, we visualize the KV cache budget allocation across different attention heads. Since we evict KV cache entries before they repeat, the visualization displays 4 rows representing head indices and 28 columns representing layer indices. Both R2 heads and summarization heads exhibit some similar patterns in their KV cache distributions, indicating that retrieval capability serves as a foundational ability underlying our summarization heads, which encompass both simple retrieval mechanisms and complex information aggregation processes. There are also a few distinct patterns between R2 heads and summarization heads. Our method allocates relatively more budget to earlier layers, which are essential for complex reasoning and summarization tasks. This allocation strategy proves crucial because when faced with complex

Method	KV Budget (%)	8k	16k	32k	64k	128k	256k	512k	1M
FullKV	100	99.16	98.16	96.02	95.01	96.38	88.6	63.84	43.29
Minference	-	99.18	99.04	95.23	97.11	95.43	82.21	57.39	43.87
AdaKV	25	60.84	33.16	33.24	28.34	29.56	21.99	15.28	10.98
DuoAttention	25	**97.09**	**93.65**	**88.78**	**85.80**	**89.82**	73.78	39.10	24.73
HeadKV	25	82.27	70.31	67.18	68.52	74.81	66.98	44.36	28.90
Ours	25	91.30	83.73	81.06	85.09	87.19	**83.29**	**48.86**	**34.16**

Table 1: Accuracy comparison on OpenAI MRCR dataset with fine-tuned Qwen2.5-7B-Instruct-1M.

queries, models may be unable to identify all necessary tokens before generation. In such scenarios, methods like HeadKV may inadvertently evict key information that lacks direct surface-level connections to the query but remains semantically relevant for comprehensive understanding.

4.3 Performance Evaluation

MRCR We first evaluate the methods on the MRCR benchmark (OpenAI, 2025b), which assesses long-context retrieval capabilities through multi-turn, synthetically generated conversations. In this task, the model engages in an extended dialogue where the user requests various pieces of writing, and the model is ultimately prompted to retrieve the i-th instance of a specific request. As shown in Table 1, our method consistently outperforms existing SOTA KV cache eviction approaches, including AdaKV and HeadKV, across all context lengths. But our method demonstrates a distinct performance profile compared to DuoAttention across varying context lengths. In the range of 8k to 128k tokens, DuoAttention exhibits a performance advantage. We attribute this to the nature of the MRCR task at these intermediate context lengths, which predominantly rewards precise, token-level information retrieval. DuoAttention's architecture, which dedicates a full KV cache to specialized retrieval heads, is exceptionally well-suited for this particular task regime. These heads are optimized to pinpoint and extract specific pieces of information, a crucial capability when the primary challenge is locating a small number of discrete needles. However, as the context length extends beyond 128k tokens, we observe a significant performance inversion, with SGD-KV consistently outperforming DuoAttention. We posit that at these extensive scales, the fundamental challenge of the MRCR task evolves. The sheer volume of information introduces a more complex problem where simple token retrieval becomes less effective. The model must not only locate information but also comprehend and navigate a vast, hierarchically structured context to distinguish between multiple, similar "needles". It is in these ultra-long-context scenarios that the architectural strengths of SGD-KV become manifest. By prioritizing "summarization heads," our method excels at hierarchical information aggregation. This allows the model to form a compressed, semantic understanding of the entire context, which is critical for effective reasoning and disambiguation when the informational landscape is exceptionally large and dense. The consistency of these summarization heads, as identified by our chunk-summarization diagnostic task, suggests they play a fundamental role in maintaining contextual coherence at extreme scales.

Method	8k	16k	32k	64k
FullKV	79.22	70.40	68.18	46.42
AdaKV	26.45	14.7	15.88	12.47
DuoAttention	76.19	59.17	60.36	28.37
HeadKV	**78.16**	65.90	**65.72**	34.90
Ours	77.05	**68.22**	62.02	**40.13**

Table 2: Accuracy comparison on OpenAI MRCR dataset with Qwen3-32B.

These results suggest that our fine-grained head-level KV cache allocation strategy becomes increasingly advantageous for maintaining performance as context length extends into the very long range.

In addition, we evaluate the methods on the Qwen3-32B model to assess the effectiveness of different methods when combined with chain-of-thought (CoT) reasoning. As shown in Table 2, the results differ slightly from those obtained with the Qwen2.5 model. Both HeadKV and our method consistently outperform DuoAttention across context lengths from 8K to 64K, indicating that head-level KV cache allocation is superior to binary classification with full KV heads and streaming

Method	KV Budget (%)	32k	64k	128k	256k	512k	1M
FullKV	100	95.4	96.6	97.2	95.6	96.2	94.6
Minference	-	95.8	96.2	97.6	96.0	87.2	76.6
DuoAttention	25	89.2	92.8	92.0	95.6	88.4	88.2
HeadKV	25	**94.8**	**96.0**	**96.6**	**95.8**	**96.0**	**94.2**
Ours	25	94.4	95.8	**96.6**	95.6	95.8	**94.2**

Table 3: Accuracy comparison on BABILong benchmark with fine-tuned Qwen2.5-7B-Instruct-1M.

heads for LLM with CoT mode. While HeadKV outperforms our method at 8K and 32K context lengths, our method demonstrates better overall performance compared to HeadKV across all evaluated context lengths.

BABILong We also evaluate the methods on the BABILong benchmark (Kuratov et al., 2024), which is specifically designed to test the retrieval and reasoning capabilities of LLMs. As shown in Table 3, despite HeadKV's use of similar question formats to identify R2 heads, it shows negligible differences compared to our summarization heads and achieves similar accuracy to the full KV cache model. This further validates the generalization ability of our summarization heads. Furthermore, both HeadKV and our method surpass DuoAttention, suggesting that when the KV cache budget is sufficient, head-level KV eviction methods typically outperform DuoAttention methods under the same KV cache constraints.

ETHIC Besides the retrieval-based benchmark, we also evaluate our method on the ETHIC benchmark (Lee et al., 2024), which comprises tasks with high Information Coverage (IC) scores (91%), representing the proportion of input context necessary for answering queries. As shown in Table 4, under the constrain of 25% KV cache, our SGDKV achieves SOTA overall performance which are very closed to the full KV cache baseline when using Qwen3-32B. Compared to the results on the MRCR dataset, the performance gains of our method are slightly smaller. We hypothesize that this is because the base models' inherent limitations create a performance ceiling, making it difficult to observe the full impact of different methods.

Model	AT	OG	RC	Avg.
Qwen2.5-7B-Instruct-1M				
Full KV	26.19	17.31	21.45	21.65
DuoAttention	25.58	**19.55**	13.40	19.51
HeadKV	26.46	15.48	20.67	20.87
Ours	**27.53**	15.38	**21.12**	**21.34**
Qwen3-32B				
Full KV	31.08	26.11	28.39	28.53
DuoAttention	28.01	20.1	25.35	24.49
HeadKV	30.90	**25.61**	28.05	28.19
Ours	**30.94**	25.60	**28.60**	**28.38**

Table 4: Performance comparison on ETHIC benchmark. "AT", "OG", and "RC" represent the attributing, organizing, and recalling sub-tasks.

5 Ablation Studies

5.1 Different KV Cache Budget

In this section, we compare the accuracy on the MRCR dataset under different KV cache budgets: 25%, 50%, and 75%, as shown in Table 5. Additional results for KV cache budgets of 10%, 20%, 30%, and 35% are provided in Appendix A.3. The results demonstrate that SGDKV consistently outperforms both AdaKV and HeadKV across all KV cache budgets, while the performance gap between HeadKV and SGDKV becomes minimal once the KV cache budget reaches 50%. And when the cache budget exceeds 50%, both SGDKV and HeadKV begin to outperform MInference, particularly for context lengths longer than 256K.

5.2 Question unaware

Our KV cache compression involves two stages: head-level budget allocation and token-level selection. The token selection step, similar to SnapKV (Li et al., 2024), uses the cumulative attention scores generated by a final observation window to identify important tokens. In this study, we ablate the choice of this observation window to understand its impact on performance. We define three conditions: **Query-Aware** (Default): The standard approach used in our main results, where the last 128 tokens (including the real query) guide token

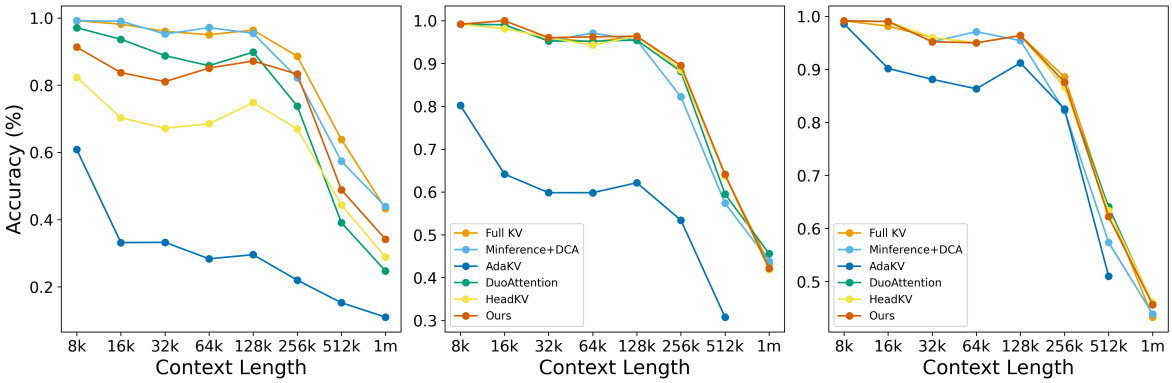

Figure 3: The comparisons of accuracy on the MRCR dataset under Different KV Cache Budget (25% (left), 50% (center), and 75% (right)). We omit the legend for the left figure due to space constraints.

Mode	Method	8k	16k	32k	64k	128k	256k	512k	1M
	FullKV	99.16	98.16	96.02	95.01	96.38	88.6	63.84	43.29
Query Aware	HeadKV	82.27	70.31	67.18	68.52	74.81	66.98	44.36	28.90
	Ours	91.30	83.73	81.06	**85.09**	**87.19**	**83.29**	48.86	**34.16**
Query Unaware	HeadKV	87.91	64.4	54.08	48.73	52.62	41.13	26.21	20.38
	Ours	93.5	85.55	73.02	70.94	77.56	71.06	44.38	33.20
Proxy Query	HeadKV	90.48	68.71	60.90	57.52	57.13	50.62	32.41	23.1
	Ours	**97.19**	**89.67**	**82.90**	82.62	83.96	80.65	**50.00**	32.24

Table 5: Accuracy comparison on OpenAI MRCR dataset with fine-tuned Qwen2.5-7B-Instruct-1M.

selection. **Query-Unaware**: The final question is excluded, and the last 128 tokens of the preceding context are used to guide selection. This simulates a scenario where the specific query is unknown during compression. **Proxy-Query**: A fixed, task-agnostic prompt—*"Segment previous text into unrelated or inconsistent chunks and select keywords for each chunk for later retrieval."* —is used as a universal proxy for the final query. This prompt contains only 32 tokens when tokenized by the Qwen2.5 model.

The results presented in Table 5 illustrates that removing the final question's guidance causes a significant performance degradation for both SGDKV and HeadKV compared to their Query-Aware performance. However, using the summarization prompt as a Proxy-Query substantially mitigates this performance loss, which demonstrates that the summarization task induces a focus on semantically salient information, making our diagnostic prompt an effective and efficient proxy for a real user query. This finding is particularly valuable for applications where the final query is not available beforehand. Finally, we note that across all three conditions, SGDKV consistently outperforms HeadKV, underscoring the robustness and superior design of our summarization-guided, head-level allocation strategy.

6 Conclusion

Based on the widely recognized view that attention heads in large language models have different types and specialized roles, we rethink the process to identify these heads and introduce the chunk-summarization task to calculate the summarization scores from each attention head. With extensive experiments results, we demonstrate that our SGDKV significantly outperforms existing approaches across complex reasoning and summarization tasks while maintaining substantial efficiency gains. We also point out a potential direction that use a proxy query to mitigate the performance degradation when performing KV eviction without access to the question. We believe our results provide insights into the identification and impact of the summarization heads. We hope it will advance the research related to head specialty and lossless KV cache compression in long-context scenarios.

Limitations

Chunk-summarization task is a more difficult task compared with the simple passkey retrieval task. Therefore, for some small and weak LLMs, they may only succeed on the samples that contain a limited number of chunks, and each chunk is easy to summarize. This may impact the quality of the summarization score distribution and result in performance degradation when applying it in allocating KV cache budget.

Acknowledgments

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.

Chenxin An, Fei Huang, Jun Zhang, Shansan Gong, Xipeng Qiu, Chang Zhou, and Lingpeng Kong. 2024. Training-free long-context scaling of large language models. *Preprint*, arXiv:2402.17463.

Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, Ido Shahaf, Oren Tropp, Ehud Karpas, Ran Zilberstein, Jiaqi Zeng, Soumye Singhal, Alexander Bukharin, Yian Zhang, Tugrul Konuk, and 114 others. 2025. Llama-nemotron: Efficient reasoning models. *Preprint*, arXiv:2505.00949.

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

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does bert look at? an analysis of bert's attention. *Preprint*, arXiv:1906.04341.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3290 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*.

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

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

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Pin-Lun Hsu, Yun Dai, Vignesh Kothapalli, Qingquan Song, Shao Tang, Siyu Zhu, Steven Shimizu, Shivam Sahni, Haowen Ning, Yanning Chen, and Zhipeng Wang. 2025. Liger-kernel: Efficient triton kernels for LLM training. In *Championing Open-source DEvelopment in ML Workshop @ ICML25*.

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. 2024. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. 2024. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. In *Advances in Neural Information Processing Systems*, volume 37, pages 106519–106554. Curran Associates, Inc.

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

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online. Association for Computational Linguistics.

Shibamouli Lahiri. 2014. Complexity of Word Collocation Networks: A Preliminary Structural Analysis. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 96–105, Gothenburg, Sweden. Association for Computational Linguistics.

Taewhoo Lee, Chanwoong Yoon, Kyochul Jang, Donghyeon Lee, Minju Song, Hyunjae Kim, and Jaewoo Kang. 2024. Ethic: Evaluating large language models on long-context tasks with high information coverage. *arXiv preprint arXiv:2410.16848*.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. Snapkv: Llm knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. 2023. Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yi Lu, Xin Zhou, Wei He, Jun Zhao, Tao Ji, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Longheads: Multi-head attention is secretly a long context processor. *Preprint*, arXiv:2402.10685.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. In-context learning and induction heads. *Preprint*, arXiv:2209.11895.

OpenAI. 2025a. Graphwalks. `https://huggingface.co/datasets/openai/graphwalks`. Accessed: [2025-05].

OpenAI. 2025b. Mrcr. `https://huggingface.co/datasets/openai/mrcr`. Accessed: [2025-05].

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Jiajun Song, Zhuoyan Xu, and Yiqiao Zhong. 2024. Out-of-distribution generalization via composition: a lens through induction heads in transformers. *Preprint*, arXiv:2408.09503.

Hanlin Tang, Yang Lin, Jing Lin, Qingsen Han, Danning Ke, Shikuan Hong, Yiwu Yao, and Gongyi Wang. 2025. Razorattention: Efficient KV cache compression through retrieval heads. In *The Thirteenth International Conference on Learning Representations*.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. 2024. Quest: Query-aware sparsity for efficient long-context llm inference. *Preprint*, arXiv:2406.10774.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2025. Retrieval head mechanistically explains long-context factuality. In *The Thirteenth International Conference on Learning Representations*.

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

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

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, and 9 others. 2025. Qwen2.5-1m technical report. *arXiv preprint arXiv:2501.15383*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Re, Clark Barrett, Zhangyang

Wang, and Beidi Chen. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024a. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Mingchuan Yang, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2024b. Attention heads of large language models: A survey. *Preprint*, arXiv:2409.03752.

A Appendix

A.1 Experiments details

We use the Llama Factory library (Zheng et al., 2024a) as the framework to supervise fine-tuning the Qwen2.5-7B-Instruct-1M (Yang et al., 2025) model. We use full-parameter fine-tuning, cosine learning schedule with initial learning rate of 1.0×10^{-5}, warm-up ratio is 0.1, total batch size is 128, and total training epochs is 2. To reduce GPU memory, we use flash attention (Dao, 2024), DeepSpeed with stage 0 (Rasley et al., 2020), and Liger kernel (Hsu et al., 2025). We conduct all experiments using eight A100 GPUs. For all inference benchmarks, we employ greedy decoding.

A.2 Quantitatively Analysis of the Stability and Generalization of the Summarization Heads

At first, we investigate whether summarization heads exhibit consistent patterns across different datasets and tasks. Since we do not impose hard thresholds to classify heads as "summarization heads," we maintain continuous importance scores for all heads, following the approach of HeadKV (Fu et al., 2024).

To quantitatively assess distributional consistency, we rank all attention heads by their importance scores in descending order and compute Intersection over Union (IoU) between ranked lists from different experimental conditions:

$$\text{IoU@}k = \frac{|\mathcal{H}_1^{(k)} \cap \mathcal{H}_2^{(k)}|}{|\mathcal{H}_1^{(k)} \cup \mathcal{H}_2^{(k)}|} \tag{5}$$

where $\mathcal{H}_i^{(k)}$ represents the top-k heads from ranking i.

Fig. 4 demonstrates high consistency in summarization head identification. The blue curve shows IoU between two disjoint 100-sample subsets from Databricks Dolly dataset (Conover et al., 2023), achieving near-perfect overlap (IoU > 0.9), indicating robust head identification within datasets. Cross-dataset analysis reveals substantial generalization: Dolly vs. DialogSum (Chen et al., 2021) (orange), Dolly vs. SAMSum (green), and DialogSum vs. SAMSum (red) maintain high IoU scores, particularly for top-ranked heads. Notably, dialogue-based datasets (DialogSum and SAMSum) exhibit stronger similarity, suggesting task-specific head specialization.

For the final configuration of the summarization heads, we use the average scores from the 6 datasets, DialogSum dataset (Chen et al., 2021), SAMSum dataset (Gliwa et al., 2019), CNN/Dailymail dataset (See et al., 2017), Extreme Summarization (XSum) dataset (Narayan et al., 2018), and the summarization part of the Databricks Dolly dataset (Conover et al., 2023) and WikiLingua (Ladhak et al., 2020) dataset.

We also compare summarization heads with retrieval heads and recently proposed R2 heads. The right panel of Fig. 4 shows that while the top 20% of heads overlap significantly across head types, the 20-60% percentile range reveals distinct preferences, confirming that summarization heads capture unique attention patterns beyond simple retrieval mechanisms.

A.3 Performance under Different KV Cache Budget

The results in Fig.5 demonstrate the robust performance of SGDKV, which consistently surpasses both AdaKV and HeadKV across almost all KV cache budgets. The performance margin narrows only in extreme cases (KV cache budgets <15% or >50%). Moreover, when the cache budget exceeds 50%, both SGDKV and HeadKV begin to outperform MInference, particularly for context lengths longer than 256K. This finding aligns with our results on the BABILong benchmark, confirming that given an adequate cache, head-level eviction strategies are more effective than token-level approaches like DuoAttention under identical budget constraints. Due to time and computational constraints, we omitted some results for DuoAttention (at 10%, 15%, 35% budgets) and AdaKV (at 10%, 15%, 20%, 30%, 35% budgets). Nevertheless, the trends established by the available data strongly suggest that these omissions do not alter our overall conclusions.

A.4 Efficiency Analysis

In terms of system-level efficiency metrics, such as peak GPU memory usage and Time to First Token (TTFT), our approach exhibits performance comparable to that of HeadKV, owing to the same implementation of head-level KV cache eviction. The primary distinction lies in our more effective distribution of the KV cache budget across individual attention heads. Consequently, under an identical overall KV cache budget, our method surpasses

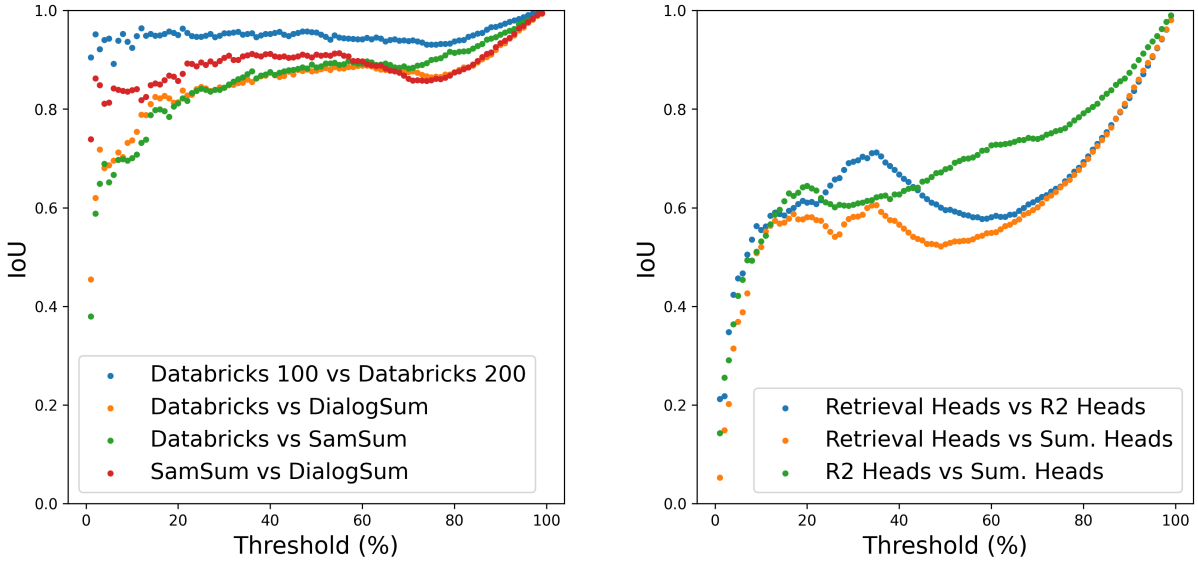

Figure 4: The comparisons of IoU scores between summarization heads with different datasets (left) and different types of heads (right).

HeadKV in accuracy while maintaining similar efficiency.

A.5 Head Configurations Variants

In this section, we conduct a series of experiments with different configurations of the summarization heads and the R2 heads to evaluate the effectiveness of both types. Due to time and computational resource constraints, we omit some results on the 1M context length. However, since the observed trends are generally consistent across context lengths, the absence of these results does not affect our overall conclusions. We introduce an additional column in Table 6, Full KV Heads (%), which denotes the proportion of attention heads granted access to the full KV cache.

In the first (top) part of Table 6, we report results for the summarization-head variant of DuoAttention (denoted DuoAttention (Sum.)). Within the DuoAttention framework, using summarization heads consistently underperforms compared to R2 heads. This suggests that R2 heads prioritize ranking heads by retrieval capability, whereas summarization heads primarily focus on fine-grained allocation of KV cache across heads. We further evaluate DuoAttention (Sum., Reverse), where the bottom 75% of heads (with low summarization scores) are assigned full KV cache access, while the top 25% only retain access to initial and recent tokens. The substantial performance drop demonstrates that constraining the KV cache of high-summarization-score heads is far more detrimental than constraining low-score heads, thereby validating the informativeness of the summarization score.

In the second (middle) part of Table 6, we present results for SGDKV (Reverse), where the KV cache allocation is inverted. For instance, a head originally assigned 10% of the KV cache is instead allocated 90%. The third row shows that even with 3× KV cache (and it got out of memory (OOM) when context length is 1M), SGDKV (Reverse) performs significantly worse than SGDKV, further underscoring the importance of score-guided distribution. We also report results for SGDKV + HeadKV, obtained by averaging the normalized R2 scores and summarization scores. As expected, its performance lies between HeadKV and SGDKV.

In the last row of the second part, we evaluate SGDKV (thr), where heads with summarization scores below the mean are set to zero before redistribution. This strategy biases cache allocation toward higher-scoring heads, yielding a larger proportion of full KV heads. Results show improved accuracy for context lengths up to 64k, but degraded performance beyond 128k. This indicates that heads with relatively low summarization scores remain important for ultra-long context modeling.

In the last (bottom) part of Table 6, we investigate the effect of leveraging summarization scores to guide attention score aggregation in models with GQA. Take the Qwen2.5-7B-Instruct-1M model as an example: it has 28 attention heads but only 4

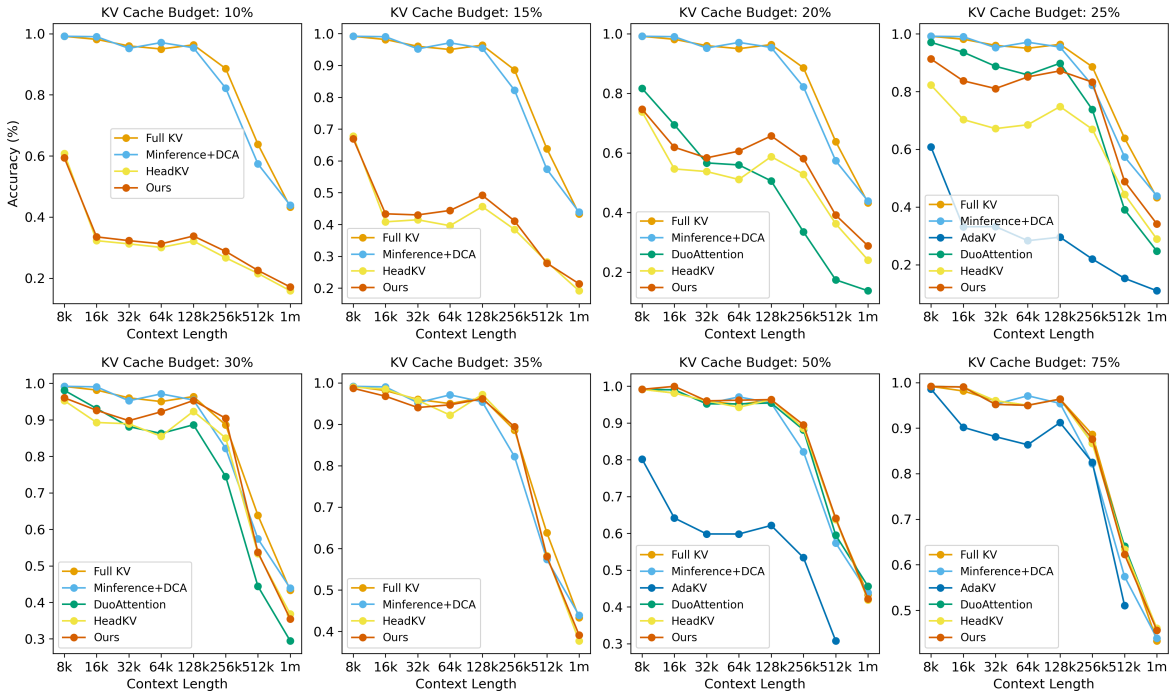

Figure 5: The comparisons of accuracy on the MRCR dataset under Different KV Cache Budget.

key-value (KV) heads, corresponding to a group size of 7. When computing summarization scores, we obtain a score for each individual attention head, but these must be aggregated into a single score per KV head. The most straightforward approaches are averaging or taking the maximum. In our experiments, we adopt the maximum, as it consistently outperforms the mean operation by a small margin.

Based on this, we evaluate three new configurations:

- **Ipt., Max**: multiply the attention scores by the summarization scores for each head, followed by max-pooling within each group.
- **Ipt., Mean**: multiply the attention scores by the summarization scores for each head, followed by mean-pooling within each group.
- **Ipt., Only**: use only the summarization scores. For instance, if head 3 has the highest summarization score within a group of 7, then all 7 heads inherit the attention scores of head 3.

As shown in the results, all three methods improve performance to varying degrees. Among them, Ipt., Max achieves the best overall accuracy from 8k to 1M context lengths, indicating that incorporating summarization scores into attention score aggregation is beneficial for GQA models.

Method	Full KV Heads(%)	8k	16k	32k	64k	128k	256k	512k	1M
FullKV	100	99.16	98.16	96.02	95.01	96.38	88.6	63.84	43.29
DuoAttention (Retrieval)	25	64.05	37.26	32.70	29.41	17.43	15.32	13.38	9.76
DuoAttention (R2)	25	**97.09**	**93.65**	88.78	85.80	89.82	73.78	39.10	24.73
DuoAttention (Sum.)	25	92.11	82.25	72.86	70.56	79.55	65.24	28.45	16.74
DuoAttention (Sum., Reverse)	75	45.62	14.13	12.75	8.5	5.69	5.96	6.26	4.87
HeadKV	0.0	82.27	70.31	67.18	68.52	74.81	66.98	44.36	28.90
SGDKV	3.57	91.30	83.73	81.06	85.09	87.19	83.29	48.86	34.16
SGDKV (Sum., Reverse)	0.0	47.21	17.31	15.85	11.31	8.21	8.62	8.61	OOM
SGDKV + HeadKV	1.79	88.12	77.55	75.34	75.52	82.53	77.32	45.28	30.70
SGDKV (thr)	16.96	95.51	89.62	82.49	85.43	86.24	70.03	32.22	OOM
SGDKV (Ipt., max)	3.57	95.5	91.49	**89.00**	**87.74**	**91.16**	**87.75**	**57.23**	**39.95**
SGDKV (Ipt., mean)	3.57	94.37	89.55	87.34	85.37	90.55	85.37	55.57	36.40
SGDKV (Ipt., only)	3.57	95.71	86.35	86.38	87.30	90.63	82.09	49.91	34.60

Table 6: Accuracy comparison between different methods on OpenAI MRCR dataset from 8K to 1M sequence length using fine-tuned Qwen2.5-7B-Instruct-1M.