Anchoring the Cache: Mitigating Contextual Hallucination in KV-Compressed Long-Context Generation

Anonymous ACL submission

Abstract

Key-Value (KV) cache compression techniques have improved the efficiency of long-context inference in Large Language Models (LLMs), but their impact on model hallucination remains underexplored. In this paper, we present the first systematic study of how KV cache compression affects hallucination in long-context summarization, demonstrating that aggressive compression can increase hallucination scores by up to 335.9%. To mitigate this issue, we propose HalluKV, a decoding-phase strategy that selectively removes generated KV pairs from retrieval heads responsible for retrieving critical information from source context, thereby anchoring their attention on the preserved source information. Our approach maintains computational efficiency while significantly reducing hallucinations across multiple models and datasets, achieving up to 5.48 average point reductions on Llama-3-8B-Instruct, enabling more trustworthy long-context generation.

1 Introduction

Modern LLMs are beginning to support extremely long sequences to meet the diverse needs of users for both input and output lengths. For example, models like Gemini 2.5 Pro (Comanici et al., 2025) have demonstrated the ability to handle inputs of up to 1 million tokens with a maximum output length of 60,000 tokens. This capability unlocks powerful real-world applications, such as answering complex questions that require understanding an entire book (Kryściński et al., 2021; Chang et al., 2023), summarizing lengthy legal or technical documents (Kanapala et al., 2019; Akter et al., 2025).

However, the attention mechanism in LLMs introduces quadratic computational complexity with respect to sequence length. For instance, processing a 128K token context with Llama-3.1-8B (Dubey et al., 2024) requires approximately 67GB of GPU memory for KV cache storage alone, with inference latency growing significantly as attention must be computed over the entire cache for each generated token. To address this challenge, KV cache compression techniques have emerged. For example, StreamingLLM (Xiao et al., 2023) maintains attention sinks and sliding windows for infinite-length generation with bounded memory. SnapKV (Li et al., 2024) selects important KV pairs using attention scores from observation windows.

While these compression methods achieve impressive efficiency improvements, a critical question remains largely unexplored: **how does KV cache compression affect hallucination in generated text?** Most existing work evaluates compression methods using surface-level metrics such as ROUGE score (Lin, 2004), which measure lexical overlap but fail to detect hallucinations where models generate fluent yet factually incorrect content (Maynez et al., 2020). More importantly, with longer input contexts, models tend to struggle to capture the complete context and rely on incomplete representations (Liu et al., 2023). Current compression methods further exacerbate this issue by discarding contextual information during the prefilling phase.

In this work, we present the first systematic study of how KV cache compression affects hallucination in long-context generation. Through extensive experiments on long-context summarization, we reveal two critical findings. (1) We demonstrate that hallucination exhibits a snowballing effect where models progressively drift from source content as generation continues, and this effect becomes dramatically worse under aggressive KV cache compression, with hallucination scores increasing up to 335.9%. (2) By analyzing attention patterns during generation, we find that KV cache compression causes specific retrieval heads to shift attention from source context toward generated content, undermining their crucial role in maintaining factual

grounding.

Building on these insights, we propose HalluKV, a simple yet effective decoding-phase KV cache eviction strategy. We selectively remove generated KV pairs from retrieval heads during decoding, forcing them to remain anchored on the compressed source context. This targeted intervention mitigates the root cause of contextual hallucination while preserving the efficiency gains from prefilling-phase KV cache compression. Experiments across various models and datasets demonstrate that HalluKV consistently outperforms existing KV cache compression methods, maintaining high ROUGE scores while substantially reducing hallucination.

Our contributions can be summarized as follows:

- We present the first systematic study demonstrating that KV cache compression methods consistently cause hallucination issues, with rates increasing up to 335.9% under aggressive compression.
- We identify one of the root causes of contextual hallucination: retrieval heads, which are crucial for maintaining factual grounding, shift attention from source context to generated content when KV cache budgets are constrained.
- We propose HalluKV, a simple yet effective decoding-phase KV cache eviction strategy that selectively removes generated KV pairs from retrieval heads, significantly reducing hallucination while preserving computational efficiency.

2 Background

2.1 KV cache compression

In LLMs, Multi-Head Attention maps input $\mathbf{X} = \{x_1, x_2, \ldots, x_{L_c}\}$ to query, key, value vectors for each head h in layer l:

$$\mathbf{Q}_h^l = \mathbf{X}\mathbf{W}_{Q,h}^l; \mathbf{K}_h^l = \mathbf{X}\mathbf{W}_{K,h}^l; \mathbf{V}_h^l = \mathbf{X}\mathbf{W}_{V,h}^l$$

The attention output is computed as:

$$\mathbf{a}_h^l = softmax(\frac{\mathbf{Q}_h^l(\mathbf{K}_h^l)^T}{\sqrt{d_k}})\mathbf{V}_h^l$$

For auto-regressive generation, as shown in Figure 1, keys and values from the prefilling phase remain constant and can be cached:

$$\mathbf{M} = \{(K_1, V_1), (K_2, V_2), \ldots, (K_{L_c}, V_{L_c})\}$$

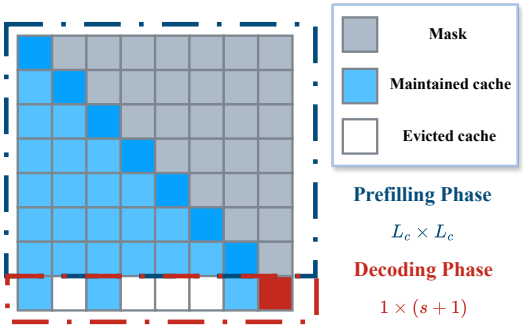

Figure 1: The calculation matrix to get attention. The prefilling phase process the input and only the maintained KV caches after KV cache compression are used during decoding phase.

As context length increases, KV cache grows linearly, requiring higher GPU memory. KV cache compression methods (Xiao et al., 2023; Li et al., 2024; Cai et al., 2024; Fu et al., 2024) evict less important KV pairs while preserving essential information:

$$\mathbf{M} = \{(K_{s_1}, V_{s_1}), (K_{s_2}, V_{s_2}), \ldots, (K_{s_n}, V_{s_n})\}$$

where $n \ll L_c$, enabling a trade-off between computational efficiency and model performance.

2.2 Retrieval Head Identification

Recently, Wu et al. (2024b) proposed retrieval heads by leveraging the Needle-in-a-haystack experiment to identify attention heads responsible for retrieving critical tokens from long context.

In this approach, a query q is paired with a target answer k (the "needle"), which is inserted into a long irrelevant passage c at different positions. During decoding, each head h is assigned an importance score S_h based on its attention to the target token k. The head whose maximum attention weight corresponds to the target token is identified and scored, and these scores are accumulated across the generation process to obtain the final head-level importance score distribution $\mathbf{S}$.

Building upon this definition, Fu et al. (2024) extends the concept of retrieval heads by generalizing the importance score computation to consider attention over the entire inserted needle and constructing a new dataset to mitigate pattern-matching effects. With these modifications, the importance score distribution $\mathbf{S}$ becomes less sparse, providing more reliable guidance for KV cache budget allocation during the prefilling phase. Overall, retrieval heads play a crucial role in preserving core information from context.

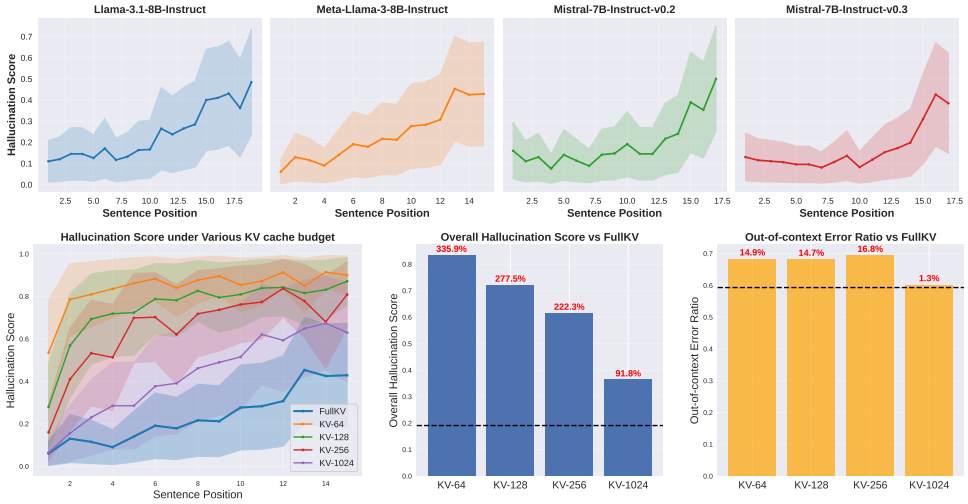

Figure 2: Upper: Snowballing hallucination across different models with FullKV. Lower: For the same Llama-3-8B-Instruct model, the hallucination snowballing effect becomes more severe as the KV cache budget decreases. All results are evaluated on Multi_news dataset.

3 Motivation Study

3.1 KV Cache Compression Amplifies Hallucination Snowballing

LLMs often generate outputs that drift away from the source information as generation progresses, a phenomenon known as contextual hallucination (Yang et al., 2025). While this issue can significantly reduce model reliability, existing research on model efficiency, particularly KV cache compression methods, has not thoroughly examined their impact on hallucination. To investigate this gap, we conduct a systematic study across different models and varying KV cache budgets, revealing two key findings.

Baseline Hallucination Patterns. We first establish baseline hallucination patterns using full KV cache (FullKV) across various models, as shown in the upper panel of Figure 2. All evaluated models exhibit a hallucination snowballing effect, where the hallucination score increases progressively as generation continues. These findings align with existing observations (Yang et al., 2025).

KV Cache Compression Amplifies Hallucination. We further examine the Llama-3-8B-Instruct model under various KV cache budgets using SnapKV compression. Results demonstrate that the hallucination snowballing effect becomes significantly more pronounced as the cache budget decreases (lower panel of Figure 2). Specifically, under aggressive compression with a 64 KV cache budget, hallucination scores increase by up to 335.9% compared to FullKV, indicating that KV cache budget constraints substantially exacerbate the underlying hallucination problem. Furthermore, our analysis reveals that the proportion of out-of-context errors increases significantly under constrained KV cache budgets (lower panel of Figure 2), confirming that models lose connection to the source context during generation.

These findings highlight the need for hallucination-aware KV cache compression methods that can maintain computational efficiency while preserving model reliability.

3.2 Inference-time Attention Deviation

To understand why KV cache compression amplifies hallucination, we investigate how budget constraints influence attention patterns, with particular focus on the critical role of retrieval heads in maintaining contextual information.

To quantify attention behavior, we introduce the *lookback ratio*, defined as the proportion of attention weight assigned to source context tokens versus generated tokens. Our analysis across different KV cache budgets reveals two critical insights.

Global Attention Shift Under Compression. We first examine overall attention patterns across the entire model. As shown in the leftmost panel of Figure 3, the lookback ratio generally exhibits a gradual decline as generation progresses. More importantly, this decline becomes significantly more pronounced under constrained KV cache budgets. Specifically, models with smaller cache budgets show a steeper and faster transition from context-dependent to self-referential attention patterns, in-

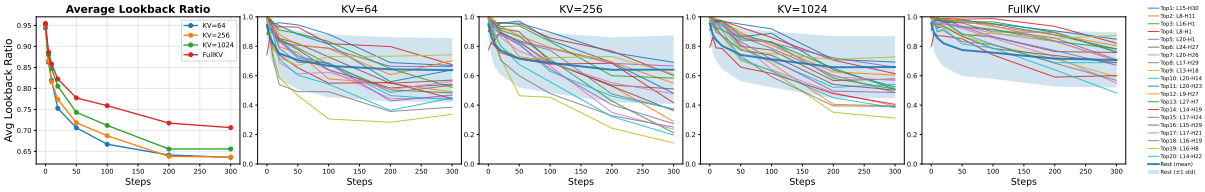

Figure 3: Lookback ratio across generation steps under different KV cache budgets using Gov_report dataset. Left: Overall lookback ratio shows that models shift attention from source context to generated content as generation progresses, with smaller KV cache budgets accelerating this decline. Right: Retrieval heads maintain high lookback ratios but experience significant degradation under constrained KV cache budgets.

dicating that KV cache compression accelerates the model's drift away from source information.

Retrieval Head Vulnerability to Compression. To further investigate whether models can consistently focus on the source information after KV cache compression, we examine the attention patterns of retrieval heads, which are crucial for maintaining source information. As shown in the right panel of Figure 3, these heads show high lookback ratios under FullKV, consistently maintaining above-average lookback ratios (thick blue line) and keeping the model connected to source information throughout generation. However, those retrieval heads show a significant decline in their lookback ratios when KV cache budgets are reduced.

This finding reveals that KV compression methods affect all attention heads, especially those most important for maintaining core information. When KV cache budgets are constrained, retrieval heads lose their ability to consistently focus on source information, which directly contributes to contextual hallucination. This suggests that hallucination can be mitigated by ensuring these retrieval heads maintain their original role in attending to the source context after KV cache compression.

4 Method

This section presents our proposed HalluKV method. Based on our motivation study showing that KV cache compression causes retrieval heads to increasingly attend to generated content, we propose a decoding-phase eviction strategy that anchors these critical heads to the source context throughout generation to mitigate contextual hallucination.

4.1 Overview

Given an input context $\mathbf{X} = \{x_1, x_2, \ldots, x_{L_c}\}$ during the prefilling phase, existing KV cache compression methods reduce the cache size by retaining only the most important KV pairs. For a head h in layer l, the compressed KV cache after prefilling is:

$$\mathbf{M}_h^l = \text{Compress}(\{(K_{h,i}^l, V_{h,i}^l)\}_{i=1}^{L_c}, \text{budget}) \quad (1)$$

where the compression function selects a subset of KV pairs based on their importance. During the decoding phase, as the model generates tokens $\{y_1, y_2, \ldots, y_t\}$, the KV cache is extended with newly generated KV pairs. However, as shown in our motivation study, this causes retrieval heads to gradually shift attention away from the compressed source context toward the generated content, leading to the hallucination snowballing effect.

Our method addresses this issue by selectively managing the KV cache for retrieval heads during the decoding phase. Specifically, we prevent retrieval heads from attending to the generated content, forcing them to remain anchored to the compressed source context. This allows us to leverage the efficiency gains from prefilling-phase compression while mitigating contextual hallucination during generation.

4.2 Retrieval Head Selection

We leverage the retrieval head importance score distribution $\mathbf{S}$ introduced by Wu et al. (2024b) to identify which heads play critical roles in maintaining contextual information. Given the importance scores, we select the top-k retrieval heads based on a mask ratio α:

$$\mathcal{H}_{ret} = \{h \mid S_h \geq \text{TopK}(\mathbf{S}, k)\}, \; k = \lfloor \alpha \cdot H \rfloor \quad (2)$$

where H is the total number of heads in the LLM, and $\alpha \in [0, 1]$ controls the proportion of heads to be masked. These selected heads undergo special KV cache management during the decoding phase to prevent attention drift.

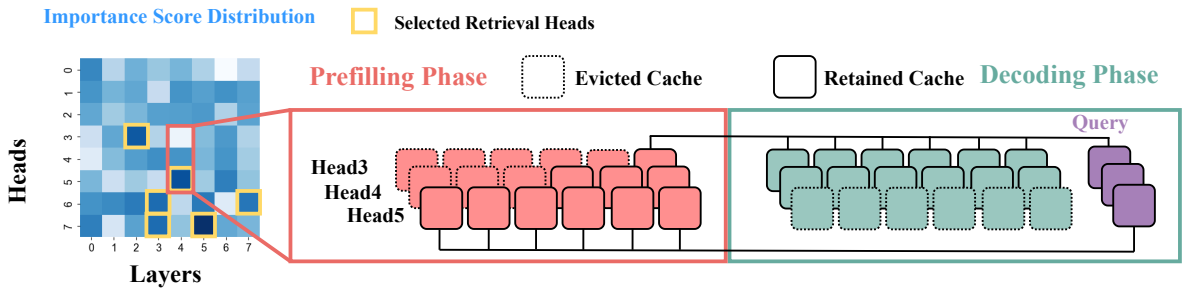

Figure 4: HalluKV: Hallucination-Aware KV Cache compression Strategy. Left: Importance score distribution across attention heads (Y-axis) and layers (X-axis), with selected retrieval heads highlighted in yellow. Prefilling Phase: KV cache from important heads is retained while less important cache is evicted to satisfy budget constraints. Decoding Phase: For selected retrieval heads, generated KV cache is evicted to force them to focus on source context, while other heads use both retained and generated cache for generation.

4.3 Decoding-Phase KV Cache Anchoring

During the decoding phase at generation step t, for each selected retrieval head $h \in \mathcal{H}_{ret}$, we implement the following KV cache management strategy:

Step 1: Maintain Compressed Source Context. We preserve all compressed KV pairs from the prefilling phase that correspond to the input context:

$$\mathbf{M}^l_{h,ctx} = \mathbf{M}^l_h \tag{3}$$

where $\mathbf{M}^l_h$ is the compressed KV cache obtained from the prefilling phase as shown in Eq. 1.

Step 2: Evict Generated Content. For the selected retrieval heads, we remove all KV pairs corresponding to previously generated tokens:

$$\mathbf{M}^l_{h,gen} = \{(K^l_{h,j}, V^l_{h,j}) \mid L_c < j < t\} \rightarrow \emptyset \quad \forall h \in \mathcal{H}_{ret} \tag{4}$$

Step 3: Compute Attention. For the current generation step t, the attention computation for retrieval heads only considers the compressed source context:

$$\mathbf{a}^l_{h,t} = \text{softmax}\left(\frac{\mathbf{Q}^l_{h,t}(\mathbf{K}^l_{h,ctx})^T}{\sqrt{d_k}}\right)\mathbf{V}^l_{h,ctx} \quad \forall h \in \mathcal{H}_{ret} \tag{5}$$

For non-retrieval heads $h \notin \mathcal{H}_{ret}$, the standard attention mechanism operates over the full KV cache including both compressed source context and generated content:

$$\mathbf{a}^l_{h,t} = \text{softmax}\left(\frac{\mathbf{Q}^l_{h,t}(\mathbf{K}^l_{h,full})^T}{\sqrt{d_k}}\right)\mathbf{V}^l_{h,full} \quad \forall h \notin \mathcal{H}_{ret} \tag{6}$$

where $\mathbf{K}^l_{h,full}$ includes both the compressed source context and generated KV pairs. This ensures that the model maintains awareness of its generation history for coherence and fluency.

5 Experiment

5.1 Experiment Details

Models and Datasets. We evaluate our HalluKV method on six open-source LLMs: Llama-3-8B-Instruct, Llama-3.1-8B-Instruct (Dubey et al., 2024), Mistral-7B-Instruct-v0.2, Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), Qwen2-7B-Instruct (Team et al., 2024), and Qwen2.5-7B-Instruct (Yang et al., 2024b). The evaluation is conducted on two long-context summarization datasets from LongBench (Bai et al., 2023): Multi_news for multi-document summarization and Gov_report for long-document summarization.

Evaluation Metrics. We assess model performance across two key dimensions. For generation quality, we employ ROUGE scores (Lin, 2004) to measure the lexical overlap between generated and reference summaries. For hallucination detection, we use FineSurE (Song et al., 2024), a fine-grained automated framework powered by Claude-3.7 (Cla). The hallucination score represents the percentage of unfaithful statements, where lower values indicate better faithfulness.

Implementation Details. We compare against three strong KV cache compression methods:

Model	Method	KV=64				KV=128				KV=256				KV=1024				Avg.	
		Gov_report		Multi_news		Gov_report		Multi_news		Gov_report		Multi_news		Gov_report		Multi_news			
		ROUGE↑	Hal.↓	ROUGE↑	Hal.↓	ROUGE↑	Hal.↓	ROUGE↑	Hal.↓	ROUGE↑	Hal.↓	ROUGE↑	Hal.↓	ROUGE↑	Hal.↓	ROUGE↑	Hal.↓	Avg.ROUGE↑	Avg.Hal.↓
Llama-3-8B-Instruct	SnapKV	19.50	73.01	19.56	83.10	20.83	65.34	21.83	71.96	22.15	56.55	23.53	61.45	25.92	38.88	26.68	36.56	22.50	60.86
	PyramidKV	19.79	70.83	20.20	79.71	21.02	63.73	22.10	70.56	22.47	54.26	23.84	59.86	25.95	36.02	26.87	35.32	22.78	58.79
	HeadKV	21.69	59.39	23.32	60.33	23.14	50.73	24.80	46.89	24.43	45.78	25.84	37.32	27.52	27.70	27.27	22.50	24.75	43.83
	HalluKV(Ours)	**22.53**	**57.84**	**24.54**	**55.81**	**24.04**	**45.14**	**26.19**	**40.03**	**25.86**	**37.53**	**27.15**	**29.73**	**28.58**	**20.69**	**28.13**	**19.95**	**25.88**	**38.34**
Llama-3.1-8B-Instruct	SnapKV	19.76	64.77	18.93	76.10	22.08	55.79	21.17	68.45	24.30	52.90	22.73	58.59	28.56	34.73	25.88	35.98	22.93	55.91
	PyramidKV	19.99	63.25	19.57	73.00	22.19	57.37	21.11	65.21	24.22	50.80	22.64	57.58	28.61	34.91	25.98	34.80	23.04	54.61
	HeadKV	23.66	51.12	22.27	57.16	25.7	40.81	24.16	43.73	28.18	32.93	25.69	33.37	32.11	16.60	26.67	21.24	26.06	37.12
	HalluKV(Ours)	**24.59**	**47.14**	**23.32**	**51.32**	**26.95**	**36.01**	**24.95**	**37.33**	**29.42**	**25.36**	**26.38**	**29.77**	**33.08**	**13.28**	**27.49**	**18.73**	**27.02**	**32.37**
Mistral-7B-Instruct-v0.2	SnapKV	18.62	73.23	19.21	85.59	20.46	58.97	21.78	72.50	22.16	46.16	22.89	59.09	26.70	24.75	26.35	29.04	22.27	56.17
	PyramidKV	18.42	70.14	19.43	83.20	20.38	58.58	21.44	71.45	22.06	46.75	22.97	56.40	26.31	26.46	25.78	30.32	22.10	55.41
	HeadKV	23.04	**40.58**	23.37	**48.51**	25.57	**28.12**	24.89	35.59	27.69	19.98	25.53	**25.62**	30.03	12.35	26.66	**16.93**	25.85	**28.46**
	HalluKV(Ours)	**23.91**	44.20	**24.15**	49.89	**26.66**	32.16	**25.31**	**34.19**	**28.44**	**19.46**	**25.80**	25.66	**30.49**	**10.39**	**27.08**	17.07	**26.48**	29.13
Mistral-7B-Instruct-v0.3	SnapKV	20.73	71.06	18.64	85.47	23.00	59.62	21.39	69.33	24.64	50.77	22.59	57.48	28.92	27.24	25.82	28.31	23.22	56.16
	PyramidKV	21.00	68.06	18.75	80.25	22.45	60.57	20.76	68.24	24.35	48.04	22.82	55.55	28.61	25.28	25.75	27.82	23.06	54.23
	HeadKV	23.79	42.36	22.19	52.11	26.85	34.70	24.02	36.21	29.24	20.82	25.44	24.77	32.64	9.79	26.70	**15.13**	26.36	29.49
	HalluKV(Ours)	**25.35**	**39.97**	**23.09**	**48.13**	**28.41**	**30.57**	**24.35**	**34.84**	**30.37**	**20.80**	**25.77**	**24.30**	**33.80**	**9.44**	**26.81**	15.57	**27.24**	**27.95**
Qwen2-7B-Instruct	SnapKV	16.67	70.99	14.84	87.09	19.32	60.54	17.82	70.06	21.83	49.05	19.69	56.21	27.11	22.37	23.36	26.78	20.08	55.38
	PyramidKV	16.77	71.32	14.93	82.51	18.28	61.73	16.50	71.12	20.41	50.28	18.54	59.47	24.93	29.81	22.97	30.15	19.17	57.05
	HeadKV	21.34	45.01	19.71	**47.54**	24.23	31.00	22.04	36.92	26.74	20.43	23.70	25.38	30.99	9.67	24.20	15.89	24.12	28.98
	HalluKV(Ours)	**21.88**	**43.10**	**20.28**	48.04	**25.08**	**29.82**	**22.78**	**33.60**	**27.64**	**19.96**	**23.90**	**23.79**	**31.36**	**9.63**	**24.90**	**15.61**	**24.73**	**27.94**
Qwen2.5-7B-Instruct	SnapKV	18.34	68.36	15.95	79.49	20.75	54.93	18.05	63.01	23.34	43.42	19.73	54.13	27.58	22.72	22.96	28.41	20.84	51.81
	PyramidKV	18.31	68.40	15.49	75.32	19.41	59.05	16.31	67.23	21.81	44.06	18.51	53.95	25.69	30.86	22.28	33.48	19.73	54.04
	HeadKV	21.53	**39.29**	20.04	50.55	24.75	28.20	21.06	37.42	27.39	**19.89**	22.49	29.57	30.41	**10.45**	23.89	18.18	23.95	29.19
	HalluKV(Ours)	**22.13**	40.35	**20.99**	**48.89**	**25.38**	**27.21**	**22.08**	**36.65**	**28.01**	21.02	**23.46**	**27.84**	**30.86**	10.71	**24.62**	**16.85**	**24.69**	**28.69**

Table 1: Performance comparison across six LLMs, four KV cache budgets (64, 128, 256, 1024), and two datasets (Gov_Report and Multi_news). HalluKV consistently achieves the best or competitive performance in both ROUGE score and hallucination score (Hal.) across different settings. Bold values indicate the best performance in each configuration.

SnapKV (Li et al., 2024), PyramidKV (Cai et al., 2024), and HeadKV (Fu et al., 2024), ensuring all methods retain the same number of KV cache entries for fair comparison. Our proposed HalluKV is built upon HeadKV, where the hyper-parameter β from HeadKV, used to control the budget allocation strategy, is set to 1.05. The local window size w is set to 8.

The mask ratio α controls the trade-off between hallucination reduction and generation quality. We set $\alpha = 0.2$ in our experiments. We provide a detailed analysis of the impact of different mask ratio values in Section 5.4. Unless otherwise specified, all analysis experiments are conducted using Llama-3-8B-Instruct on the Multi_news dataset to ensure consistency and comparability across different analyses.

5.2 Main Results

Table 1 presents the comprehensive evaluation results of HalluKV across six LLMs and multiple KV cache budgets. Our method consistently outperforms all baseline approaches in both hallucination reduction and generation quality.

First, HalluKV achieves substantial hallucination reduction across all evaluated models. The hallucination reductions are particularly significant on Llama-3-8B-Instruct, where HalluKV reduces hallucination scores by an average of 5.48 points compared to the strong HeadKV baseline. Moreover, these reductions are consistent across different KV cache budgets, with hallucination scores dropping from 60.33% to 55.81% at KV = 64 (a 7.5% relative reduction) and from 22.50% to 19.95% at KV = 1024 (an 11.3% relative reduction) on the Multi_news dataset, indicating the robustness of HalluKV even when the preserved source information is limited.

Second, HalluKV achieves hallucination reduction without compromising generation quality. The method simultaneously improves ROUGE scores across all models, indicating better alignment with reference summaries. On Llama-3-8B-Instruct, HalluKV achieves an average ROUGE gain of 1.13 points over HeadKV, with consistent improvements observed across different KV cache budgets. This suggests that anchoring retrieval heads to the source context not only reduces factual errors but also enhances the informativeness and coherence of generated text.

5.3 Fine-grained Hallucination Evaluation

To systematically evaluate whether HalluKV effectively maintains model attention on source context throughout the generation process, we partition generated summaries into two equal segments corresponding to the first and second halves of the output. We compute hallucination scores for each

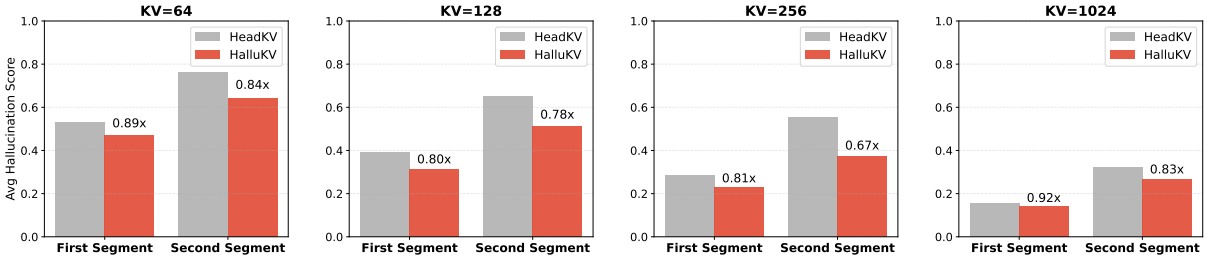

Figure 5: Comparison of hallucination scores between HeadKV and HalluKV across first and second halves of generated summaries under different KV cache budgets. HalluKV shows more substantial reduction in the second segment, effectively mitigating the "Hallucination at the last" phenomenon.

individual sentence and then calculate the average hallucination score within each segment to assess how our method's effectiveness varies across different generation stages.

Figure 5 demonstrates that our method achieves more significant reductions in hallucination scores for the second segment compared to the first segment. By selectively removing generated KV caches, HalluKV consistently shows larger relative reductions in the latter portion of the generated summaries, with the second segment achieving 0.84x to 0.67x greater reduction across different KV cache budgets. This strategy is particularly effective at reducing factual errors in the latter portions of the summaries where the model tends to rely more heavily on its own generated content.

5.4 The impact of mask ratio

To investigate the optimal balance between hallucination reduction and generation quality, we conduct an ablation study on the mask ratio parameter α, which controls the proportion of retrieval heads selected for generated KV cache eviction. We systematically vary the mask ratio from 0% to 50% across different KV cache budgets (64, 128, 256, and 1024) and evaluate hallucination scores to understand how the degree of head masking affects factual consistency.

Figure 6 reveals a characteristic inverted U-shaped relationship between mask ratio and hallucination reduction. The results demonstrate that moderate mask ratios (15% to 30%) achieve optimal performance, with hallucination scores reduced by 9% to 16% compared to the baseline. As the mask ratio increases beyond 40%, performance begins to deteriorate, with the 50% mask ratio showing a notable increase in hallucination scores. This degradation occurs because excessive masking removes heads that may serve other important functions beyond source information retrieval.

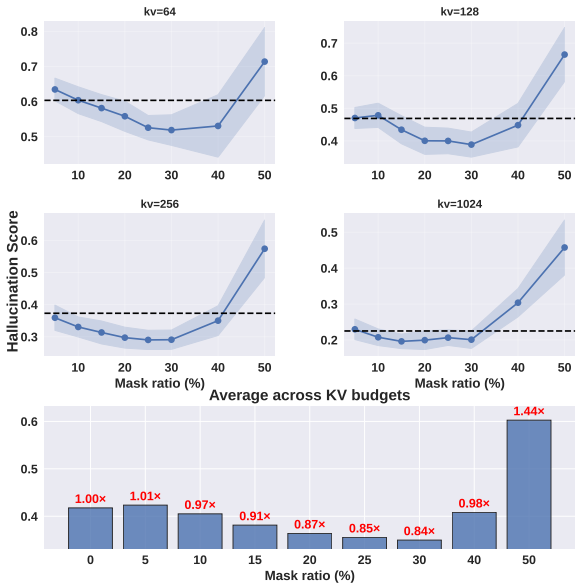

Figure 6: Impact of mask ratio α on hallucination reduction across different KV cache budgets.

5.5 Integration with Other Prefilling-Only Methods

To demonstrate the generalizability of our approach, we evaluate its effectiveness when combined with different prefilling-based KV cache compression methods. We integrate our decoding-phase strategy with two representative prefilling methods: SnapKV and HeadKV. Following the optimal mask ratio settings identified in Section 5.4, we select the best-performing configuration for each baseline method.

Table 2 reveals notable results regarding the integration of our method with prefilling-based approaches. First, our approach consistently reduces hallucination across all baseline methods and KV budgets, with average reductions of 5.33 points for SnapKV and 5.49 points for HeadKV. Second, our method demonstrates robustness even when the underlying prefilling method retains limited

Method	KV=64		KV=128		KV=256		KV=1024		Avg
	Gov_report	Multi_news	Gov_report	Multi_news	Gov_report	Multi_news	Gov_report	Multi_news	
SnapKV	73.47	82.32	64.68	71.92	55.97	61.55	38.44	35.74	60.52
+ Ours	**69.07**	**80.32**	**60.24**	**67.24**	**51.13**	**52.64**	**28.82**	**31.07**	**55.19**
HeadKV	59.39	60.33	50.73	46.89	45.78	37.32	27.70	22.50	43.83
+ Ours	**57.84**	**55.81**	**45.14**	**40.03**	**37.53**	**29.73**	**20.69**	**19.95**	**38.34**

Table 2: Hallucination scores when we integrate our method with SnapKV and HeadKV across different KV cache budgets. Our approach consistently reduces hallucination scores compared to baseline methods.

contextual information, as evidenced by the consistent reductions achieved with SnapKV despite its weaker baseline performance.

6 Related Work

6.1 KV Cache Compression Methods

The KV cache is widely used in autoregressive decoding to avoid redundant computation: at each time step, attention keys and values are stored and reused across future tokens, reducing decoding time from linear to constant per token.

However, as context length increases, the KV cache grows linearly, leading to significant computational efficiency challenges. Various KV cache compression methods (Xiao et al., 2023; Li et al., 2024; Cai et al., 2024; Fu et al., 2024) have been proposed to address this challenge. For example, StreamingLLM (Xiao et al., 2023) maintains attention sinks and a sliding window to enable infinite-length generation with bounded memory. SnapKV (Li et al., 2024) employs observation windows and leverages attention scores to select the most important KV cache entries for generation. Moreover, SCOPE (Wu et al., 2024a) introduces a framework that separately optimizes KV cache during both prefilling and decoding phases, preserving essential information during prefilling while using a sliding strategy to select important heavy hitters during decoding to handle long-context long-generation tasks.

While these methods achieve impressive efficiency improvements, they primarily focus on computational efficiency during the prefilling phase, with limited consideration of their impact on generation quality and factual consistency.

6.2 Hallucination Detection and Mitigation

Model hallucinations are often subtle and difficult to detect, significantly undermining the usability and trustworthiness of LLMs. Even powerful GPT models can fail to answer simple factual questions (Wei et al., 2024).

Recent work has identified specific patterns that cause hallucination and tried to mitigate it through representation editing (Shi et al., 2024; Chuang et al., 2024; Wang et al., 2025). For example, Lookback Lens (Chuang et al., 2024) introduces the lookback ratio, a metric that captures how much a model relies on the provided context versus its own generated tokens during decoding. This feature is used to identify hallucinations and guide the decoding process toward more faithful outputs. Building upon the lookback ratio concept, GAME (Wang et al., 2025) proposes a novel approach that dynamically adjusts attention maps to improve contextual relevance by employing a trained classifier to identify attention maps prone to inducing hallucinations and executing targeted interventions guided by gradient-informed edit directions.

While some work (Yang et al., 2024a) has recognized the hallucination problems caused by evicting KV cache, there has been limited systematic analysis. Our work addresses this gap by providing the first comprehensive study of how KV cache compression affects hallucination in long-context generation, revealing the hallucination snowballing effect under compression and proposing targeted mitigation strategies.

7 Conclusion

In this paper, we discovered that KV cache compression exacerbates contextual hallucination by causing retrieval heads to shift attention from source context to generated content. We proposed HalluKV, a decoding-phase KV cache eviction strategy that selectively removes generated KV pairs from retrieval heads while preserving their access to compressed source context, thereby anchoring these critical heads to source information throughout generation. Experimental results demonstrate that HalluKV consistently outperforms existing methods, achieving substantial hallucination reduction while maintaining generation quality.

Limitations

While our approach demonstrates significant reductions in contextual hallucination, several limitations warrant consideration. First, our method relies on the identification of retrieval heads based on the Needle-in-a-haystack experiment, which may not fully capture the complex and heterogeneous roles that different attention heads play in long-context generation. In reality, attention heads often serve multiple functions simultaneously, including retrieval, in-context learning (Olsson et al., 2022) and safety (Zhou et al., 2024), and our binary classification of heads as either "retrieval" or "non-retrieval" may oversimplify this complexity. This limitation could potentially lead to suboptimal head selection, where heads that serve dual purposes might be inappropriately masked, potentially affecting generation quality in ways not captured by our current evaluation metrics.

Second, our approach assumes that the importance score distribution of retrieval heads remains consistent across different types of long-context tasks and domains. However, the specific roles and importance of different heads may vary depending on the task characteristics, document types, and generation requirements. This domain-specific variation could limit the generalizability of our method across diverse long-context applications beyond summarization tasks.

References

Claude 3.7 sonnet system card.

Mousumi Akter, Erion Çano, Erik Weber, Dennis Dobler, and Ivan Habernal. 2025. A comprehensive survey on legal summarization: Challenges and future directions. *arXiv preprint arXiv:2501.17830*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and 1 others. 2024. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2023. Booookscore: A systematic exploration of book-length summarization in the era of llms. *arXiv preprint arXiv:2310.00785*.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James Glass. 2024. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. *arXiv preprint arXiv:2407.07071*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

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*.

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

Ambedkar Kanapala, Sukomal Pal, and Rajendra Pamula. 2019. Text summarization from legal documents: a survey. *Artificial Intelligence Review*, 51(3):371–402.

Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2021. Booksum: A collection of datasets for long-form narrative summarization. *arXiv preprint arXiv:2105.08209*.

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

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, and 1 others. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791.

Hwanjun Song, Hang Su, Igor Shalyminov, Jason Cai, and Saab Mansour. 2024. Finesure: Fine-grained summarization evaluation using llms. *arXiv preprint arXiv:2407.00908*.

Qwen Team and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2:3.

Yu Wang, Kamalika Das, Xiang Gao, Wendi Cui, Peng Li, and Jiaxin Zhang. 2025. Gradient-guided attention map editing: Towards efficient contextual hallucination mitigation. *arXiv preprint arXiv:2503.08963*.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*.

Jialong Wu, Zhenglin Wang, Linhai Zhang, Yilong Lai, Yulan He, and Deyu Zhou. 2024a. Scope: Optimizing key-value cache compression in long-context generation. *arXiv preprint arXiv:2412.13649*.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2024b. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Joonho Yang, Seunghyun Yoon, Hwan Chang, Byeongjeong Kim, and Hwanhee Lee. 2025. Hallucinate at the last in long response generation: A case study on long document summarization. *arXiv preprint arXiv:2505.15291*.

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

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024b. Qwen2.5 technical report. *ArXiv*, abs/2412.15115.

Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, Kun Wang, Yang Liu, Junfeng Fang, and Yongbin Li. 2024. On the role of attention heads in large language model safety. *arXiv preprint arXiv:2410.13708*.

A Out of Context Error Analysis

To further validate the effectiveness of HalluKV in reducing contextual hallucination, we analyze the proportion of out-of-context errors, which represent statements that cannot be supported by the source context. Figure 8 presents a comprehensive comparison between HalluKV and baseline methods across different KV cache budgets for all six LLMs used in Table 1.

The results demonstrate that HalluKV consistently achieves lower out-of-context error ratios compared to HeadKV across all tested models and KV cache sizes. For the Llama models, HalluKV shows substantial reductions: Meta-Llama-3-8B-Instruct reduces errors from 68.7% to 52.5% at KV=64, while Llama-3.1-8B-Instruct achieves similar reductions from 65.5% to 51.7% at KV=64. The Mistral models also benefit significantly, with Mistral-7B-Instruct-v0.2 reducing errors from 53.0% to 47.2% at KV=64, and Mistral-7B-Instruct-v0.3 showing reductions from 60.9% to 53.5% at KV=64. For the Qwen models, HalluKV consistently outperforms HeadKV, reducing out-of-context errors from 64.9% to 57.3% for Qwen2-7B-Instruct and from 38.6% to 34.3% for Qwen2.5-7B-Instruct at KV=64.

These results provide strong evidence that HalluKV's decoding-phase eviction strategy effectively anchors generation to the source context, significantly reducing the proportion of unfaithful statements that lack support from the input documents. The consistent reduction across all six models and various KV cache budgets demonstrates the robustness and generalizability of our approach in mitigating contextual hallucination.

B Computational Efficiency

To demonstrate that HalluKV maintains computational efficiency while reducing hallucination, we conduct efficiency evaluations using Mistral-7B-Instruct-v0.2. We measure peak memory usage across context lengths from 1K to 256K tokens and decoding latency across generation lengths from 10 to 4096 tokens. All experiments are conducted on a single H200 GPU with consistent hardware settings to ensure fair comparison. For latency measurements, we use a maximum context length of 32,768 tokens to maintain computational feasibility while still capturing the efficiency characteristics across different generation lengths.

Figure 7 shows that HalluKV maintains identical memory efficiency compared to HeadKV, achieving substantial memory savings over FullKV (34% reduction at 256K context length). The decoding latency analysis reveals that HalluKV introduces only minimal computational overhead (5-7% increase) compared to baseline compression methods while still providing significant improvements over FullKV (34% latency reduction at 4096 generation length).

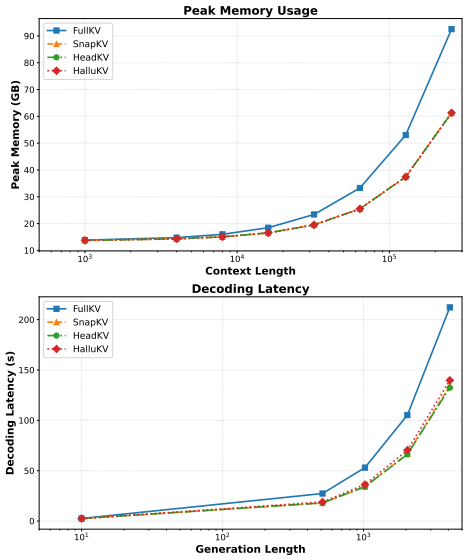

Figure 7: Computational efficiency comparison between HalluKV and baseline methods using Mistral-7B-Instruct-v0.2. Top: Peak memory usage across different context lengths. Bottom: Decoding latency across different generation lengths. HalluKV maintains the computational efficiency as other baseline methods.

These results demonstrate that HalluKV successfully maintains the efficiency gains from prefilling-phase compression while adding negligible computational overhead, making it practically viable for real-world applications.

C Case Analysis

To demonstrate HalluKV's effectiveness in hallucination reduction, we conduct a case analysis as shown in Figure 9. The results reveal the difference between HeadKV and HalluKV: HeadKV generates 5 out-of-context errors out of 7 statements (71.4%), while HalluKV can mitigate to zero out-of-context errors across all generated statements. This qualitative evidence confirms that HalluKV's strategy of anchoring retrieval heads to source context effectively prevents the introduction of information not present in the source context.

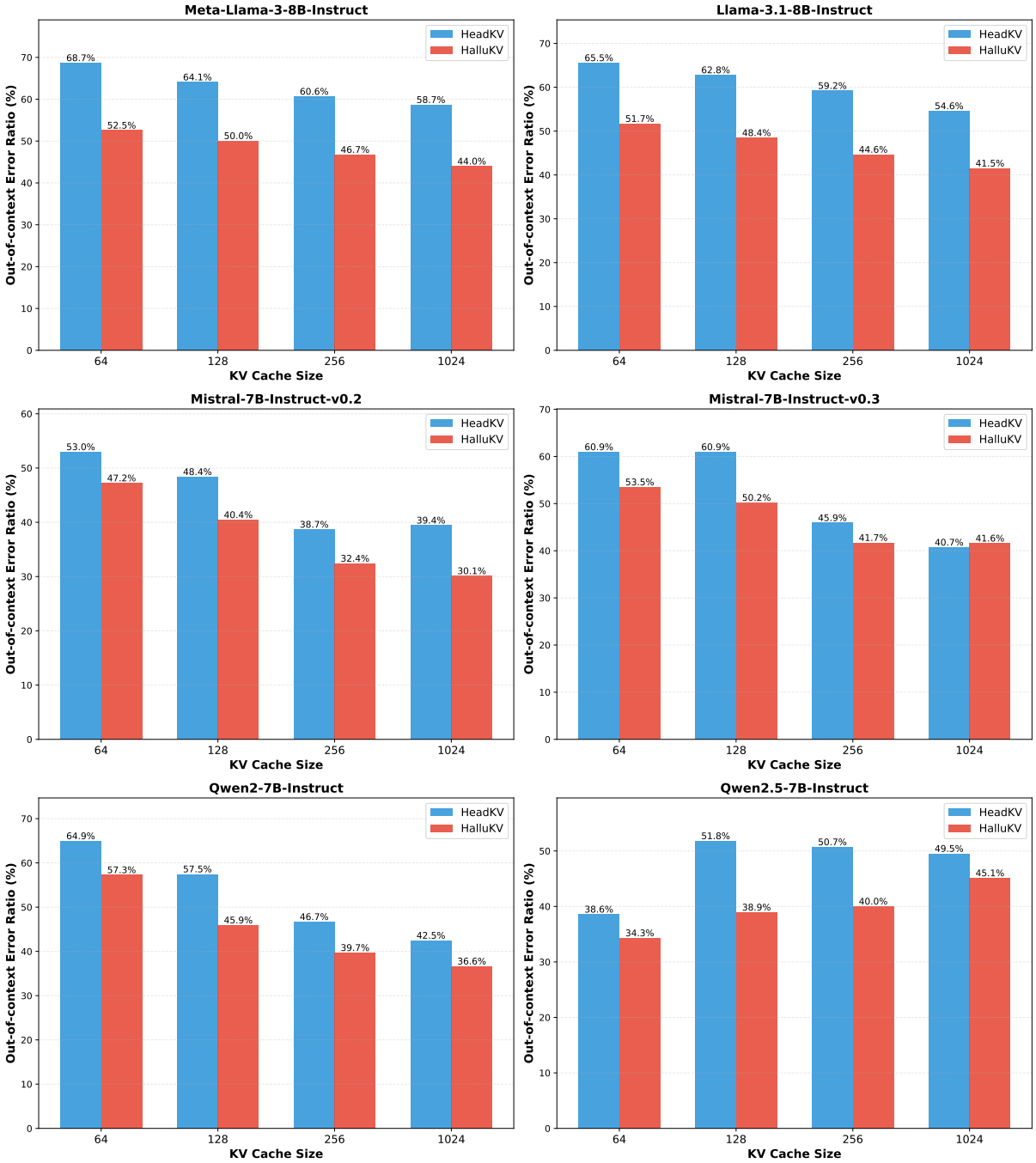

Figure 8: The proportion of out-of-context errors across different KV cache budgets. HalluKV consistently maintains a lower proportion of out-of-context errors compared to baseline methods.

Reference Summary	The LDA, as amended, requires lobbyists to file quarterly disclosure reports and semiannual reports on certain political contributions. The law also includes a provision for GAO to annually audit lobbyists' compliance with the LDA. GAO's objectives were to (1) determine the extent to which lobbyists can demonstrate compliance with disclosure requirements, (2) identify challenges to compliance that lobbyists report, and (3) describe the resources and authorities available to USAO in its role in enforcing LDA compliance, and the efforts USAO has made to improve enforcement. This is GAO's 11th report under the provision. GAO reviewed a stratified random sample of 98 quarterly disclosure LD-2 reports filed for the third and fourth quarters of calendar year 2016 and the first and second quarters of calendar year 2017. GAO also reviewed two random samples totaling 160 LD-203 reports from year-end 2016 and midyear 2017. This methodology allowed GAO to generalize to the population of 45,818 disclosure reports with $5,000 or more in lobbying activity, and 30,594 reports of federal political campaign contributions. GAO also met with officials from USAO to obtain status updates on its efforts to focus resources on lobbyists who fail to comply. GAO is not making any recommendations in this report. GAO provided a draft of this report to the Department of Justice for review and comment. The Department of Justice provided technical comments, which GAO incorporated as appropriate. For the 2017 reporting period, most lobbyists provided documentation for key elements of their disclosure reports to demonstrate compliance with the Lobbying Disclosure Act of 1995, as amended (LDA). For lobbying disclosure (LD-2) reports and political contributions (LD-203) reports filed during the third and fourth quarter of 2016 and the first and second quarter of 2017, GAO estimates that 87 percent of lobbyists filed reports as required for the quarter in which they first registered; the figure below describes the filing process and enforcement; 99 percent of all lobbyists who filed (up from 83 percent in 2016) could provide documentation for income and expenses; and 93 percent filed year-end 2016 LD-203 reports as required. These findings are generally consistent with prior reports GAO issued for the 2010 through 2016 reporting periods. However, in recent years GAO's findings showed some variation in the estimated percentage of reports with supporting documentation. For example, an estimated increase in lobbyists who could document expenses is notable in 2017 and represents a statistically significant increase from 2016. As in GAO's other reports, some lobbyists were still unclear about the need to disclose certain previously held covered positions, such as paid congressional internships or certain executive agency positions. GAO estimates that 15 percent of all LD-2 reports may not have properly disclosed previously held covered positions. On the other hand, over the past several years of reporting on lobbying disclosure, GAO found that most lobbyists in the sample rated the terms associated with LD-2 reporting as “very easy” or “somewhat easy” to understand. The U.S. Attorney's Office for the District of Columbia (USAO) stated it has sufficient resources and authority to enforce compliance with the LDA. USAO continued its efforts to bring lobbyists into compliance by reminding them to file reports or by applying civil penalties.

HeadKV

The report, \\\"Lobbying Disclosure: Observations on Lobbyists' Compliance with New Disclosure Requirements,\\\" examines the extent to which lobbyists comply with new disclosure requirements.	This sentence accurately reflects the general purpose of the transcript, which examines lobbyist compliance with disclosure requirements under the LDA.	No Error
The report finds that while some lobbyists have made efforts to comply, many still struggle to provide accurate and complete information.\	his statement is consistent with the transcript, which shows varying levels of compliance across different reporting requirements.	No Error
he report highlights several challenges that lobbyists face in complying with the new requirements, including difficulties in identifying and reporting lobbying activities, and a lack of clarity on what constitutes a \\\"lobbying contact.\\\"	While the transcript mentions some challenges lobbyists face, it does not specifically mention difficulties in identifying what constitutes a \\\"lobbying contact\\\" as a key challenge.	Out-of-Context Error
Additionally, the report notes that some lobbyists may be intentionally evading disclosure requirements, which can undermine the transparency and accountability of the lobbying process.	Additionally, the report notes that some lobbyists may be intentionally evading disclosure requirements, which can undermine the transparency and accountability of the lobbying process.	Out-of-Context Error
To address these challenges, the report recommends that the government agency responsible for enforcing the disclosure requirements provide clearer guidance on what constitutes a \\\"lobbying contact\\\" and offer additional training and resources to help lobbyists comply with the new requirements.\	The transcript does not contain these specific recommendations about providing clearer guidance on \\\"lobbying contact\\\" or offering additional training and resources.	Out-of-Context Error
The report also suggests that the agency consider implementing a system for tracking and monitoring lobbying activities, which could help to improve compliance and increase transparency.\	The transcript describes existing tracking systems used by USAO but does not suggest implementing a new system for tracking and monitoring lobbying activities.	Out-of-Context Error
Overall, the report concludes that while there are challenges to complying with the new disclosure requirements, many lobbyists are making efforts to comply and that with additional guidance and resources, the process can be improved.	While the transcript does indicate that many lobbyists find compliance relatively easy and are making efforts to comply, it does not specifically conclude that additional guidance and resources would improve the process.	Out-of-Context Error

HalluKV

The report, \\\"Lobbying Disclosure: Observations on Lobbyists' Compliance with New Disclosure Requirements,\\\" examines the extent to which lobbyists comply with the Lobbying Disclosure Act (LDA) of 1995.	This statement is consistent with the transcript, which describes a report examining lobbyist compliance with the LDA.	No Error
The report finds that while the majority of lobbyists are able to demonstrate compliance with the law, there are still challenges and areas for improvement.	This statement accurately reflects the transcript's findings about general compliance with some challenges.	No Error
The report notes that the Lobbying Disclosure Act (LDA) requires lobbyists to register and report their activities, but many lobbyists are not in compliance with the law.	While the transcript mentions non-compliance issues, it doesn't state that 'many lobbyists' are not in compliance - this overstates the findings.	Predicate Error
The report finds that 87% of lobbyists are able to demonstrate compliance with the law, but there is still a significant gap in reporting and transparency.	The 87.2% figure in the transcript refers specifically to matching new registrations to corresponding LD-2 reports, not overall compliance with the law.	Entity Error
The report also notes that the lobbying disclosure process is complex and often confusing, which can lead to errors and inconsistencies.	This is consistent with the transcript, which mentions confusion about requirements like rounding errors and covered positions.	No Error
The report recommends that the government agency responsible for enforcing the law, the Federal Election Commission (FEC), provide clearer guidance and training to lobbyists on the reporting requirements.	The transcript identifies USAO (not FEC) as responsible for enforcement, and doesn't mention this specific recommendation.	Entity Error
Overall, the report suggests that while there are some areas of improvement, the lobbying disclosure process is generally effective in providing transparency and accountability.\	This is a reasonable interpretation of the transcript's findings about general compliance with some areas needing improvement.	No Error
However, there are still some areas where improvement is needed, such as the reporting of certain types of lobbying activities.	This is consistent with the transcript, which identifies specific areas needing improvement like reporting covered positions and contributions.	No Error

Figure 9: One Case example of HalluKV from Meta-Llama-3-8B-Instruct on Gov_report dataset. (KV = 64)