Last But Not Least: Boundary Attention CalibratiON for Multimodal KV Cache Compression

Anonymous ACL submission

Abstract

Multimodal Large Language Models (MLLMs) achieve strong vision-language reasoning, but long visual contexts enlarge the KV cache and increase decoding latency. Existing compression methods rely on observation window attention for stable token-importance estimation, yet this aggregation can dilute sparse visual evidence and discard answer-critical tokens under aggressive compression. Therefore, we identify last-query attention as a complementary source for recovering such evidence, but its answer-irrelevant signals can mislead retention. We propose BACON, a plug-and-play method that calibrates observation window attention with last-query evidence and suppresses isolated noise via intra-layer coherence and inter-layer persistence. Across diverse benchmarks, models, budgets, and compression methods, BACON improves multimodal KV compression by 7.5% on average under the most aggressive budget, with gains up to 30.9%. Code is available at `https://anonymous.4open.science/r/BACON-A1106`.

1 Introduction

Multimodal Large Language Models (MLLMs) have become a central foundation for vision-language intelligence, extending the reasoning and generation capabilities of Large Language Models (LLMs) to multimodal scenarios through visual encoders and projection modules. Despite strong performance across multimodal tasks, efficient inference remains challenging as visual inputs become longer and denser. Visual tokens substantially increase the prefill length (Kwon et al., 2023), and the KV cache must retain their representations across layers and attention heads for subsequent decoding. KV cache compression has therefore become an important direction for reducing the memory and latency cost of MLLM inference.

Existing KV cache compression methods typically retain cached tokens according to their estimated importance. A common strategy is to estimate this importance from an observation window (Li et al., 2024b), i.e., the final segment of prompt queries, by averaging the attention each cached token receives from these queries. While this averaging reduces query-specific noise and stabilizes importance estimation, it also introduces a fundamental limitation: sparse answer-relevant visual evidence can be obscured by query-dependent attention signals unrelated to the target answer (Kang et al., 2025). Consequently, **window aggregation can dilute sparse visual evidence and bias retention away from answer-critical visual tokens**, especially under aggressive compression, as illustrated in Fig. 1 and Fig. 2(a).

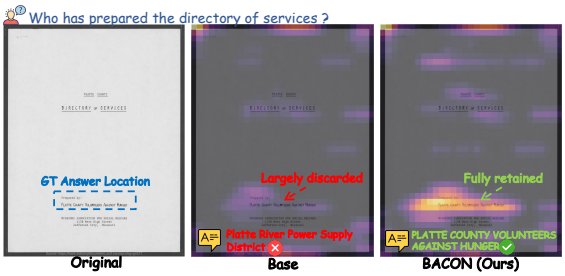


Figure 1: Visual Importance Estimation under low budget on SnapKV: Observation window vs. BACON

We investigate this limitation by analyzing attention dynamics in MLLM KV cache compression. Our analysis in Fig. 2(b) and 2(c) shows that the last prompt query can reveal important boundary evidence weakened by observation window aggregation. However, it is not sufficiently reliable as a standalone retention criterion: as shown in Fig. 3(a), nearly 70% of its high-attention tokens are non-evidential. Directly incorporating last-query attention into token retention can therefore assign excessive importance to answer-irrelevant high-attention tokens, while weakening the stability provided by observation window aggregation. This leads to a key insight: *the last query is not least for token retention; it reveals visual evidence missed by observation window aggregation, but this evidence*

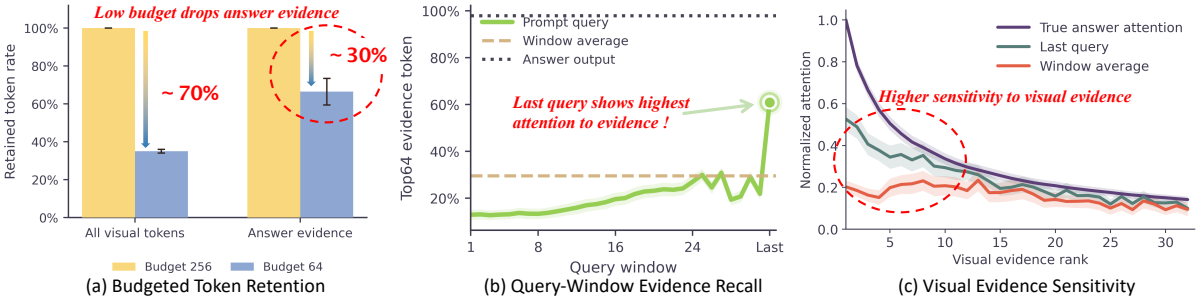


Figure 2: **Observation window aggregation can dilute sparse visual evidence, while the last query recovers these evidence.** (a) Under aggressive KV compression, SparseMM discard answer-critical visual evidence and produce incorrect predictions, suggesting observation window attention can miss sparse evidence under tight cache budgets. (b) The last query is more sensitive than earlier prompt queries to answer-relevant tokens, showing its ability to capture boundary-emergent evidence. (c) Compared with window-averaged attention, last-query attention better highlights visually important tokens and closely matches the answer's attention distribution over visual tokens.

should be calibrated against the window-based signal before guiding token retention.

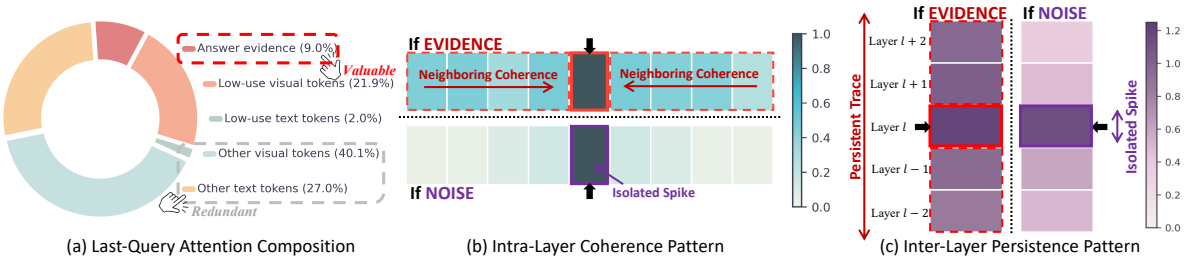
Motivated by this insight, we propose **B**oundary **A**ttention **C**alibrati**ON** (**BACON**), a plug-and-play token retention mechanism for MLLM KV cache compression. BACON keeps observation window attention as the stable basis of existing retention scores and uses last-query attention to calibrate it with boundary-emergent evidence. However, such boundary evidence must be distinguished from noisy high-attention signals before being directly incorporated into token retention. BACON therefore filters last-query signals through local coherence within each layer and persistence across adjacent layers, as discovered in Fig. 3(b) and 3(c). In this way, BACON recovers visual evidence diluted by window aggregation while avoiding the noise introduced by directly using last-query attention. As shown in Fig. 1, BACON successfully captures key visual information needed for questions compared to base counterparts. More visualization results are available in B.4.

In summary, our contributions are as follows. (1) We identify observation window aggregation as a key limitation why sparse but answer-critical visual evidence can be diluted and discarded during aggressive KV cache compression. (2) We propose BACON, a plug-and-play boundary attention calibration method that leverages evidence revealed by the last query to calibrate stable but diluted observation window attention, while suppressing isolated noise through intra-layer coherence and inter-layer persistence. (3) We conduct extensive experiments spanning multimodal understanding, video reasoning, GUI grounding, and long-context text tasks, with (M)LLMs of different scales and architectures, multiple KV compression methods, and diverse cache budgets, showing that BACON consistently improves existing compression methods without introducing extra inference cost.

2 Related Work

Multimodal Large Language Models. Recent Multimodal Large Language Models (MLLMs) extend LLMs with visual encoders and projection modules, aligning visual features with the language embedding space for joint vision-language reasoning. Representative models, such as LLaVA (Liu et al., 2024), InternVL (Zhu et al., 2025), and Qwen3-VL (Bai et al., 2025), have achieved strong performance across diverse multimodal tasks. Recent advances further support high-resolution inputs (Guo et al., 2024), dynamic image tiling (Li et al., 2024a), and multi-frame video reasoning (Maaz et al., 2024), improving fine-grained perception and long-context multimodal understanding. However, the resulting growth in visual context length substantially increases KV cache memory and decoding latency, making efficient inference critical for scalable deployment.

Efficient Inference and KV Cache Compression in MLLMs. Efficient MLLM inference has been explored mainly through visual-token reduction and KV cache compression. Visual-token reduction methods prune intermediate visual tokens (Chen et al., 2024), merge redundant tokens (Shang et al., 2025), or adapt input resolution (Wang et al., 2024), reducing computation but often modifying the visual processing pipeline or effective visual context. In contrast, KV cache compression reduces memory and decoding cost after prefill while preserving the original input representation and forward pipeline (Kim et al., 2026a; Fu et al., 2025). In LLMs, StreamingLLM (Xiao et al., 2024) and H2O



(a) Last-Query Attention Composition (b) Intra-Layer Coherence Pattern (c) Inter-Layer Persistence Pattern

Figure 3: **Why last-query attention needs calibration.** (a) The last query can highlight answer evidence, but most of its high-attention tokens are not useful: only about 9% correspond to truly important evidence, while around 70% are answer-irrelevant noise. (b) Important visual evidence usually appears as a local region, where neighboring tokens also receive high attention; in contrast, noise often appears as an isolated attention spike. (c) Important visual evidence also remains salient across adjacent layers, while noisy spikes are less consistent.

(Zhang et al., 2023) motivate retention through attention sinks and heavy hitters, while SnapKV (Li et al., 2024b), PyramidKV (Cai et al., 2024), and AdaKV (Feng et al., 2026) rely on observation window importance estimation and non-uniform layer/head budgets. Recent MLLM-oriented methods further adapt cache compression to multimodal inference (Wan et al., 2024): SparseMM (Wang et al., 2025) allocates modality-sensitive budgets, InfiniPotV (Kim et al., 2026b) uses value-norm-based visual KV selection, and MixKV (Liu et al., 2025) balances attention-based importance with semantic diversity. Despite these advances, most attention-based methods still use observation window attention as the primary retention signal. BACON is orthogonal to these methods: it calibrates window-based retention with last-query boundary evidence while preserving the original budget allocation and decoding pipeline.

3 Motivation Study

In this section, we analyze the attention patterns and paired failure cases of Qwen2-VL under SparseMM compression on DocVQA and present our key findings below:

❶ Low-budget compression discards visual evidence. To understand failures under tight cache budgets, we compare paired cases in which SparseMM succeeds with a larger budget but fails with a smaller budget. As shown in Fig. 2(a), the low-budget setting removes nearly 70% of the visual tokens retained by the high-budget setting and discards approximately 30% of the answer-aligned evidence. And many of the removed evidence tokens correspond to low-attention visual regions that are still essential for grounding the answer. This finding indicates that aggressive compression can eliminate not only redundant visual context but also weak yet semantically decisive evidence. As visual tokens generally receive lower attention than text tokens (Chen et al., 2024), they are especially vulnerable under low-budget selection, which motivates a retention signal beyond raw observation window attention.

❷ Last query captures boundary evidence. We next examine where the missing evidence appears in the attention dynamics by comparing prompt queries at different positions with observation window attention. Fig. 2(b) shows that earlier prompt queries cover only limited answer-aligned evidence, whereas the last query captures a much larger portion of tokens that are important for the answer. Fig. 2(c) further shows that this advantage is especially clear for visual tokens: last-query attention assigns higher saliency to visually important tokens and more closely matches the answer's attention distribution over visual tokens than observation window attention. These results suggest that while observation window aggregation can dilute sparse visual evidence, the last query remains sensitive to boundary-emergent visual signals that are closely tied to the forthcoming answer.

❸ Intra-layer coherence and inter-layer persistence calibrate noisy boundary evidence. Although last-query attention can discover boundary evidence weakened by observation window aggregation, Fig. 3(a) shows that its high-attention tokens are highly mixed: only 9% correspond to true answer evidence, whereas nearly 70% are answer-irrelevant noise. Directly using raw last-query attention for retention would therefore introduce irrelevant signals that interfere with important boundary evidence. As shown in Figs. 3(b) and 3(c), true evidence exhibits stronger intra-layer coherence and inter-layer persistence: high attention extends from evidence tokens to neighboring tokens within the same layer and remains concentrated on the same token positions across adjacent layers. In contrast,

answer-irrelevant high-attention tokens are more isolated within a layer and less stable across layers. These observations suggest that important boundary evidence should be salient to the last query while at the same time exhibiting intra-layer coherence and inter-layer persistence.

4 Methodology

4.1 Preliminaries: Attention-Based KV Cache Compression

Given a multimodal prompt $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{T}$, an MLLM computes the KV cache during prefill and reuses it during autoregressive decoding. For layer l and attention head h, we denote the cached keys and values by $\mathbf{K}_h^l, \mathbf{V}_h^l \in \mathbb{R}^{T\times d}$, where T is the sequence length and d is the head dimension. KV cache compression retains a compact subset of KV pairs under a budget $K_{l,h}$ by ranking candidate tokens for each layer-head pair.

Most attention-based methods estimate token importance by averaging attention over an observation window $\mathcal{W}$ near the end of the prompt. Let $\mathbf{A}^{l,h} \in \mathbb{R}^{T\times T}$ denote the causal attention matrix, where $A_{q,i}^{l,h}$ is the attention weight from query position q to key position i. The observation window score $B_i^{l,h}$ is

$$B_i^{l,h} = \frac{1}{|\mathcal{W}|}\sum_{q\in\mathcal{W}} A_{q,i}^{l,h}. \tag{1}$$

Although $B_i^{l,h}$ provides a stable relevance estimate, Sec. 3 shows that observation window aggregation can dilute sparse but answer-critical visual evidence. BACON therefore uses $B_i^{l,h}$ as the backbone score and calibrates it with boundary-emergent evidence exposed by the last query. Fig. 4 gives an overview of BACON.

4.2 Boundary-Emergent Evidence Modeling

BACON is motivated by the observation in Sec. 3 that the last prompt query can expose boundary-emergent evidence weakened by observation window aggregation, while raw last-query attention may also contain answer-irrelevant spikes. Let the last query position in the prompt be T. We define the last-query attention score as

$$Q_i^{l,h} = A_{T,i}^{l,h}. \tag{2}$$

We decompose last-query saliency into a stable window signal, a boundary-emergent evidence component, and a noise component:

$$Q_i^{l,h} = B_i^{l,h} + \Delta_i^{l,h} + \xi_i^{l,h}, \tag{3}$$

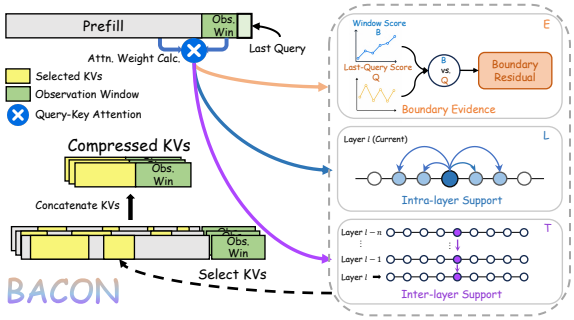


Figure 4: Overview of BACON. BACON extracts boundary evidence from observation window and last-query attention, then calibrates it with intra-layer coherence and inter-layer persistence to produce an evidence-aware score for head-wise KV cache compression.

where $\Delta_i^{l,h}$ denotes evidence revealed by the last prompt query, and $\xi_i^{l,h}$ denotes non-evidential noise. Since $\Delta_i^{l,h}$ is not directly observable, BACON estimates it as the boundary evidence estimate $E_i^{l,h}$ using the positive gap between last-query attention and observation window attention:

$$E_i^{l,h} = \left[Q_i^{l,h} - B_i^{l,h}\right]_+, \tag{4}$$

where $[x]_+ = \max(x, 0)$. The positive residual $E_i^{l,h}$ measures how much more token i is attended by the last query than by the observation window, thereby identifying evidence that may be diluted in the window-based score while preserving $B_i^{l,h}$ as the retention basis.

4.3 Structural Calibration of Boundary Evidence

As shown in Sec. 3, useful boundary evidence tends to exhibit intra-layer coherence and inter-layer persistence. BACON therefore refines the boundary residual with these two structural cues. For each layer-head pair, we write the residual scores as $\mathbf{e}^{l,h} \in \mathbb{R}^T$, whose i-th entry is $E_i^{l,h}$.

Intra-layer coherence. BACON measures intra-layer coherence by aggregating boundary residuals over neighboring token positions. For token i, we define a radius-r neighborhood, with $r = 5$ by default:

$$\mathcal{N}_r(i) = \{u \mid |u - i| \le r,\ 1 \le u \le T\}. \tag{5}$$

The row-normalized neighborhood operator $\mathbf{P}_r \in \mathbb{R}^{T\times T}$ is

$$(\mathbf{P}_r)_{i,u} = \frac{\mathbf{1}\{u \in \mathcal{N}_r(i)\}}{|\mathcal{N}_r(i)|}. \tag{6}$$

The intra-layer coherence score is

$$\mathbf{l}^{l,h} = \mathbf{P}_r\mathbf{e}^{l,h}, \qquad L_i^{l,h} = (\mathbf{l}^{l,h})_i. \tag{7}$$

Inter-layer persistence. BACON measures inter-layer persistence by tracing boundary residuals at the same token position across preceding layers. For layer l, the causal layer set is

$$\mathcal{P}_m(l) = \{j \mid \max(1, l-m) \leq j < l\}, \quad (8)$$

where m is the persistence depth, with $m = 4$ by default. The inter-layer persistence score is

$$T_i^{l,h} = \frac{1}{|\mathcal{P}_m(l)|} \sum_{j \in \mathcal{P}_m(l)} E_i^{j,h}. \quad (9)$$

If $\mathcal{P}_m(l) = \emptyset$, we set $T_i^{l,h} = 0$; using only preceding layers keeps the formulation causal and compatible with layer-by-layer compression.

Structurally calibrated boundary evidence. BACON combines the original boundary estimate with its intra-layer coherence and inter-layer persistence:

$$V_i^{l,h} = E_i^{l,h} + L_i^{l,h} + T_i^{l,h}. \quad (10)$$

In vector form,

$$\mathbf{v}^{l,h} = (\mathbf{I} + \mathbf{P}_r)\mathbf{e}^{l,h} + \mathbf{t}^{l,h}, \quad (11)$$

where $\mathbf{t}^{l,h}$ denotes the inter-layer persistence vector. BACON thus converts noisy last-query saliency into structurally calibrated boundary evidence without task-specific weighting.

4.4 Variance-Constrained Score Calibration

The structurally calibrated boundary evidence $V_i^{l,h}$ is used as a controlled calibration term rather than a replacement for the backbone score. To ensure comparability across layers and heads, BACON matches the variance of boundary evidence to the local variation of the observation window score.

For each layer-head pair, let $\mathcal{I}_{l,h}$ be the candidate set. We define the scale-matching region under the backbone score as

$$M_{l,h} = \min(\rho K_{l,h}, |\mathcal{I}_{l,h}|), \quad (12)$$

$$\mathcal{M}_{l,h} = \mathrm{TopK}_{i \in \mathcal{I}_{l,h}} \left(B_i^{l,h}, M_{l,h} \right), \quad (13)$$

where ρ is an expansion factor, set to $\rho = 2$ by default. BACON obtains the calibration coefficient by solving

$$\lambda^{l,h} = \arg\min_{\lambda \geq 0} \Big(\mathrm{Std}_{i \in \mathcal{M}_{l,h}}(\lambda V_i^{l,h}) - \gamma\, \mathrm{Std}_{i \in \mathcal{M}_{l,h}}(B_i^{l,h}) \Big)^2, \quad (14)$$

where γ controls the global calibration strength. This objective has the closed-form solution

$$\lambda^{l,h} = \gamma \cdot \frac{\mathrm{Std}_{i \in \mathcal{M}_{l,h}}(B_i^{l,h})}{\mathrm{Std}_{i \in \mathcal{M}_{l,h}}(V_i^{l,h}) + \epsilon}, \quad (15)$$

where ϵ is a small constant for numerical stability. The final BACON score is

$$S_i^{l,h} = B_i^{l,h} + \lambda^{l,h} V_i^{l,h}. \quad (16)$$

This preserves the stable observation window score while injecting boundary evidence at a calibrated scale.

Finally, BACON selects the retained token set

$$\mathcal{K}_{l,h} = \mathrm{TopK}_{i \in \mathcal{I}_{l,h}} \left(S_i^{l,h}, K_{l,h} \right), \quad (17)$$

and constructs the compressed cache by retaining $\{\mathbf{K}_{h,i}^l, \mathbf{V}_{h,i}^l \mid i \in \mathcal{K}_{l,h}\}$.

5 Experiment

5.1 Experiment Settings

Model Details. We evaluate BACON across diverse model architectures and scales. For multimodal understanding, we use LLaVA-NeXT-Mistral-7B (Liu et al., 2024), InternVL3-8B (Zhu et al., 2025), Qwen2-VL-7B-Instruct (Wang et al., 2024), and Qwen3-VL-30B-A3B-Instruct (Bai et al., 2025). For text-only evaluation, we use Mistral-7B-Instruct-v0.2 (Jiang et al., 2023) and Llama3.1-8B-Instruct (Grattafiori et al., 2024).

Benchmark Details. We evaluate BACON on image understanding, video understanding, GUI grounding, and long-context text tasks. For **image understanding**, we use DocVQA (Mathew et al., 2021), TextVQA (Singh et al., 2019), MMMU (Yue et al., 2024), ChartQA (Masry et al., 2022), and TextCaps (Sidorov et al., 2020). For **video understanding**, we use VATEX (Wang et al., 2019) and NextQA (Xiao et al., 2021). For **GUI grounding**, we use ScreenSpot (Li et al., 2025). For **text understanding**, we use LongBench (Bai et al., 2024).

Implementation Details. We apply BACON to various KV cache compression methods, including SnapKV, PyramidKV, AdaKV, and SparseMM, under different cache budgets. More experiment details are provided in A.

5.2 Main Results

Performance on image understanding benchmarks. Table 1 summarizes BACON integrated

Table 1: **Main results on image understanding benchmarks.** SparseMM is excluded in InternVL3-8B and Qwen3-VL-30B-A3B settings due to the unreleased profiling file. Arrows show deltas over Base, and "Full KV" denotes caching all KV pairs as the upper bound.

Method	Variant	DocVQA (%)			TextVQA (%)			ChartQA (%)			MMMU (%)			TextCaps		
		256	128	64	256	128	64	256	128	64	256	128	64	256	128	64
Qwen2-VL-7B-Instruct																
Full KV	Upper		93.7			–			71.3			49.9			1.473	
SnapKV	Base	88.6	82.1	70.1	80.6	77.0	70.3	70.0	69.6	66.2	49.9	49.8	49.6	1.361	1.141	0.787
	+MixKV	91.8	83.9	70.9	82.6	81.0	73.5	70.0	70.2	67.2	49.9	49.9	49.7	1.470	1.332	0.919
	+BACON	**93.1** ↑4.5	**91.5** ↑9.4	**85.5** ↑15.4	**83.1** ↑2.5	**82.6** ↑5.6	**78.2** ↑7.9	**70.2** ↑0.2	**70.2** ↑0.6	**69.6** ↑3.4	**49.9** →0.0	**49.9** ↑0.1	**50.0** ↑0.4	**1.488** ↑0.127	**1.426** ↑0.285	**1.178** ↑0.391
PyramidKV	Base	83.4	75.6	60.5	77.7	74.9	66.8	71.1	68.9	65.2	49.9	49.8	49.6	1.147	0.993	0.600
	+MixKV	85.0	77.5	61.3	80.9	77.2	69.4	71.0	**71.1**	66.4	49.9	49.8	49.6	1.383	1.145	0.662
	+BACON	**92.3** ↑8.9	**89.3** ↑13.7	**79.2** ↑18.7	**82.5** ↑4.8	**80.0** ↑5.1	**75.1** ↑8.3	**71.2** ↑0.1	70.6 ↑1.7	**69.2** ↑4.0	**49.9** →0.0	**49.9** ↑0.1	**49.8** ↑0.2	**1.461** ↑0.314	**1.344** ↑0.351	**1.073** ↑0.473
AdaKV	Base	88.4	81.3	69.4	80.5	75.9	70.8	69.8	69.6	66.6	49.9	49.7	49.6	1.300	1.099	0.771
	+MixKV	91.4	82.7	70.7	82.5	79.1	72.6	70.2	70.2	67.8	49.9	**49.9**	49.6	1.454	1.271	0.874
	+BACON	**93.0** ↑4.6	**91.1** ↑9.8	**86.2** ↑16.8	**82.8** ↑2.3	**81.2** ↑5.3	**78.5** ↑7.7	**70.2** ↑0.4	**70.2** ↑0.6	**69.6** ↑3.0	**49.9** →0.0	49.8 ↑0.1	**49.8** ↑0.2	**1.473** ↑0.173	**1.371** ↑0.272	**1.144** ↑0.373
SparseMM	Base	93.1	91.4	87.3	82.6	82.1	76.9	70.2	70.0	69.6	49.8	49.8	49.6	1.481	1.427	1.044
	+MixKV	**93.9**	92.9	88.6	82.5	**82.5**	80.9	69.6	69.8	**70.8**	49.8	49.8	49.7	1.480	1.456	1.303
	+BACON	93.8 ↑0.7	**93.2** ↑1.8	**92.0** ↑4.7	**82.6** →0.0	82.4 ↑0.3	**81.6** ↑4.7	**70.6** ↑0.4	**70.4** ↑0.4	70.2 ↑0.6	**49.9** ↑0.1	**49.8** →0.0	**49.8** ↑0.2	**1.506** ↑0.025	**1.511** ↑0.084	**1.431** ↑0.387
LLaVA-NeXT-Mistral-7B																
Full KV	Upper		62.7			68.4			51.8			34.7			0.704	
SnapKV	Base	58.1	55.2	46.2	66.0	63.0	58.9	49.8	48.4	47.2	34.7	34.9	34.8	0.651	0.560	0.442
	+MixKV	**60.4**	57.5	48.2	**67.5**	66.1	61.3	49.8	48.6	46.8	34.6	34.8	34.8	**0.710**	0.656	0.510
	+BACON	59.7 ↑1.6	**57.8** ↑2.6	**53.9** ↑7.7	67.1 ↑1.1	**66.2** ↑3.2	**62.8** ↑3.9	**50.0** ↑0.2	**49.6** ↑1.2	**49.2** ↑2.0	**34.8** ↑0.1	**34.9** →0.0	**34.8** →0.0	0.686 ↑0.035	**0.676** ↑0.116	**0.512** ↑0.070
PyramidKV	Base	57.5	54.3	43.8	65.2	63.1	55.7	43.6	42.7	38.6	34.8	35.0	34.7	0.652	0.581	0.436
	+MixKV	60.3	56.7	45.6	67.2	65.8	57.8	44.3	42.6	39.1	34.7	34.9	34.7	0.685	0.644	0.505
	+BACON	**60.6** ↑3.1	**59.0** ↑4.7	**52.4** ↑8.6	**67.4** ↑2.2	**66.0** ↑2.9	**59.8** ↑4.1	**45.0** ↑1.4	**43.9** ↑1.2	**39.9** ↑1.3	**35.0** ↑0.2	**35.0** →0.0	**34.8** ↑0.1	**0.685** ↑0.033	**0.650** ↑0.069	**0.505** ↑0.069
AdaKV	Base	58.3	56.1	47.6	65.4	62.7	57.8	49.6	48.8	47.4	34.7	34.9	34.8	0.645	0.568	0.441
	+MixKV	59.3	57.5	49.5	**67.2**	64.4	59.6	49.6	49.2	46.8	34.7	34.9	34.8	**0.701**	0.660	0.506
	+BACON	**59.3** ↑1.0	**58.4** ↑2.3	**55.2** ↑7.6	67.1 ↑1.7	**65.5** ↑2.8	**61.5** ↑3.7	**50.0** ↑0.4	**50.2** ↑1.4	**49.0** ↑1.6	**34.8** ↑0.1	**34.9** →0.0	**34.8** →0.0	0.694 ↑0.049	**0.671** ↑0.103	**0.510** ↑0.069
SparseMM	Base	58.5	58.9	57.5	67.2	67.4	65.2	50.6	49.8	49.4	34.7	34.7	34.8	0.670	0.600	0.489
	+MixKV	58.9	59.0	**58.7**	67.4	67.6	67.1	**51.0**	50.4	**50.0**	34.8	34.7	34.7	0.685	0.620	0.569
	+BACON	**59.7** ↑1.2	**59.3** ↑0.4	58.3 ↑0.8	**68.0** ↑0.8	**67.8** ↑0.4	**67.3** ↑2.1	50.6 →0.0	**50.4** ↑0.6	49.8 ↑0.4	**34.8** ↑0.1	**34.8** ↑0.1	**34.9** ↑0.1	**0.686** ↑0.016	**0.625** ↑0.025	**0.572** ↑0.083
InternVL3-8B																
Full KV	Upper		91.1			81.6			77.8			55.3			1.111	
SnapKV	Base	89.5	85.4	75.1	80.7	78.8	72.3	77.4	75.9	71.6	55.3	55.2	55.2	1.083	0.999	0.804
	+MixKV	89.5	86.4	75.5	81.2	79.2	73.6	**77.6**	76.0	72.2	55.3	55.2	55.1	**1.103**	1.016	0.821
	+BACON	**90.0** ↑0.5	**88.7** ↑3.3	**84.9** ↑9.8	**81.4** ↑0.7	**81.2** ↑2.4	**76.7** ↑4.4	77.4 →0.0	**77.3** ↑1.4	**75.6** ↑4.0	**55.4** ↑0.1	**55.3** ↑0.1	**55.2** →0.0	1.102 ↑0.019	**1.020** ↑0.021	**0.856** ↑0.052
PyramidKV	Base	87.6	82.5	69.2	78.4	76.1	68.8	76.4	74.8	71.1	55.2	55.2	55.1	0.977	0.902	0.688
	+MixKV	87.7	83.4	69.4	79.4	76.9	68.9	76.7	75.2	71.7	55.3	55.3	55.2	1.024	0.936	0.719
	+BACON	**89.6** ↑2.0	**87.7** ↑5.2	**80.7** ↑11.5	**81.1** ↑2.7	**79.7** ↑3.6	**74.4** ↑5.6	**77.5** ↑1.1	**76.9** ↑2.1	**74.3** ↑3.2	**55.3** ↑0.1	**55.4** ↑0.2	**55.3** ↑0.2	**1.027** ↑0.050	**0.945** ↑0.043	**0.782** ↑0.094
AdaKV	Base	89.5	85.7	76.9	80.7	78.5	72.4	**77.2**	75.5	72.6	55.6	55.2	55.2	1.092	0.983	0.847
	+MixKV	89.3	86.4	77.2	81.1	79.3	73.2	76.9	76.1	72.5	55.3	55.2	55.1	1.091	1.027	0.869
	+BACON	**89.5** →0.0	**88.6** ↑2.9	**85.8** ↑8.9	**81.4** ↑0.7	**80.4** ↑1.9	**76.8** ↑4.4	77.1 ↓0.1	**76.8** ↑1.3	**75.4** ↑2.8	**55.6** →0.0	**55.2** →0.0	**55.3** ↑0.1	**1.102** ↑0.010	**1.030** ↑0.047	**0.872** ↑0.025
Qwen3-VL-30B-A3B																
Full KV	Upper		95.5			84.3			74.4			52.56			0.328	
SnapKV	Base	94.7	91.5	78.9	83.4	81.8	75.8	74.7	73.2	70.4	52.33	52.67	51.56	0.354	0.348	0.273
	+MixKV	95.3	92.7	81.0	**84.0**	83.0	78.8	74.9	74.8	72.4	52.78	53.00	51.89	**0.357**	0.426	0.334
	+BACON	**95.4** ↑0.7	**95.1** ↑3.6	**90.3** ↑11.4	83.8 ↑0.4	**83.7** ↑1.9	**81.4** ↑5.6	**75.1** ↑0.4	**74.9** ↑1.7	**73.6** ↑3.2	**53.11** ↑0.78	**53.00** ↑0.33	**52.33** ↑0.77	0.347 ↓0.007	**0.435** ↑0.087	**0.355** ↑0.082
PyramidKV	Base	86.9	88.0	74.4	80.3	80.0	73.2	72.3	72.6	68.5	52.89	52.22	51.78	0.326	0.338	0.266
	+MixKV	89.3	90.2	76.3	82.3	82.9	77.2	74.0	74.5	71.8	53.44	52.89	**52.11**	0.401	**0.398**	0.370
	+BACON	**94.8** ↑7.9	**94.3** ↑6.3	**87.2** ↑12.8	**83.7** ↑3.4	**83.1** ↑3.1	**79.3** ↑6.1	**74.7** ↑2.4	**74.6** ↑2.0	**73.2** ↑4.7	**53.44** ↑0.55	**53.56** ↑1.34	52.00 ↑0.22	**0.425** ↑0.099	0.375 ↑0.037	**0.387** ↑0.121
AdaKV	Base	94.8	91.8	80.8	83.0	81.4	75.2	74.9	73.2	70.8	52.44	52.33	52.44	0.347	0.325	0.293
	+MixKV	95.5	93.5	83.3	**84.4**	82.8	78.5	74.8	**74.7**	72.9	52.89	52.56	51.78	0.351	0.414	0.353
	+BACON	**95.7** ↑0.9	**95.0** ↑3.2	**90.6** ↑9.8	83.3 ↑0.3	**83.7** ↑2.3	**80.7** ↑5.5	**75.0** ↑0.1	74.5 ↑1.3	**73.2** ↑2.4	**53.11** ↑0.67	**53.00** ↑0.67	**53.67** ↑1.23	**0.352** ↑0.005	**0.417** ↑0.092	**0.361** ↑0.068

Table 2: **Results on video understanding benchmarks with Qwen2-VL-7B.**

Method	Variant	VATEX												NextQA		
		CIDEr			BLEU-4			METEOR			ROUGE-L			WUPS		
		512	256	128	512	256	128	512	256	128	512	256	128	512	256	128
Full KV	Upper		49.61			21.91			23.83			44.15			26.30	
SnapKV	Base	48.14	46.51	**46.27**	21.22	20.60	20.26	23.70	23.22	22.18	43.80	43.53	42.68	25.97	25.69	25.84
	+MixKV	48.55	47.68	45.95	21.34	20.82	20.32	23.00	23.42	22.07	43.90	43.55	**42.93**	25.99	25.93	25.60
	+BACON	**48.67**	**48.02**	46.02	**21.38**	**21.26**	**20.38**	**23.79**	**23.42**	**22.63**	**43.96**	**43.57**	42.79	**26.15**	**25.94**	**26.02**
AdaKV	Base	48.20	46.20	45.37	21.55	20.43	20.28	23.62	23.29	22.85	43.85	43.26	42.67	25.72	25.70	25.65
	+MixKV	48.68	47.46	**45.39**	21.22	20.78	20.34	23.68	23.28	22.87	43.92	43.32	42.68	**25.99**	**26.01**	25.86
	+BACON	**48.82**	**47.85**	45.33	**21.55**	**21.24**	**20.54**	**23.68**	**23.45**	**23.15**	**44.06**	**43.41**	**42.76**	25.96	25.95	**25.86**
PyramidKV	Base	46.09	46.03	44.36	20.58	**20.29**	19.26	**23.28**	22.86	22.02	43.25	42.41	42.18	25.88	25.69	25.63
	+MixKV	46.13	45.67	44.50	20.48	20.29	19.55	23.22	22.87	22.11	43.30	**42.69**	42.16	25.87	25.72	25.52
	+BACON	**46.57**	**46.22**	**45.70**	**20.81**	20.25	**19.70**	23.26	**22.89**	**22.24**	**43.31**	42.65	**42.36**	**26.04**	**25.80**	**25.65**
SparseMM	Base	47.91	47.16	46.12	20.79	20.24	19.77	23.52	23.18	22.24	43.67	43.53	42.63	**26.37**	25.94	25.71
	+MixKV	48.65	47.44	45.99	20.99	20.54	19.82	23.46	23.10	**22.55**	43.85	43.22	42.66	26.13	25.98	25.83
	+BACON	**48.90**	**47.57**	**46.38**	**21.28**	**20.81**	**20.03**	**23.68**	**23.31**	22.38	**44.06**	**43.56**	**42.74**	26.33	**26.17**	**26.07**

with representative KV compression baselines across multiple MLLMs, benchmarks, and cache budgets. The results show three main advantages. (i) *Consistent effectiveness.* BACON consistently improves compressed inference, especially under aggressive budgets where sparse visual evidence is more likely to be discarded. For example, on Qwen2-VL-7B with PyramidKV at budget 64, BACON improves DocVQA by +18.7 points and TextVQA by +8.3 points; similar gains with SnapKV and AdaKV indicate that last-query calibration helps recover visual evidence weakened by observation window aggregation. (ii) *Broad applicability.* BACON benefits diverse compression paradigms, including attention-based selection, adaptive allocation, and head-/layer-wise budget allocation. On LLaVA-NeXT-Mistral-7B at budget 64, it improves PyramidKV by +8.6 points on DocVQA and SparseMM by +2.1 points on TextVQA. Since BACON only refines within-head token scores without changing the compression operator or budget allocation, it can be directly integrated into existing KV compression pipelines. (iii) *Model compatibility.* BACON generalizes from 7B-scale MLLMs to stronger architectures, including InternVL3-8B and the 30B-scale MoE model Qwen3-VL-30B-A3B. With PyramidKV at budget 64, it improves Qwen3-VL-30B-A3B by +12.8 points on DocVQA and +6.1 points on TextVQA, supporting its role as a plug-and-play calibration method across model families and scales.

Performance on video understanding bench-

Table 3: **Results on ScreenSpot GUI grounding benchmark with Qwen2-VL-7B.**

Method	Variant	Mobile Text		Mobile Icon		Desktop Text		Desktop Icon		Web Text		Web Icon		Average	
		128	64	128	64	128	64	128	64	128	64	128	64	128	64
Full KV	–	39.4		32.1		17.5		10.0		7.0		6.8		18.8	
SnapKV	Base	24.5	15.4	10.5	4.8	19.1	9.3	4.3	4.3	5.7	6.5	6.8	5.8	11.8	7.7
	+ MixKV	24.9	15.8	**10.8**	4.8	19.6	10.8	4.3	4.3	6.1	**6.5**	6.8	6.3	12.1	8.1
	+ BACON	**24.9**	**15.8**	10.7	**5.7**	**21.1**	**13.7**	**4.3**	**5.7**	**7.0**	6.4	**8.2**	**6.3**	**12.7**	**9.0**
AdaKV	Base	26.1	15.4	11.8	4.2	19.1	11.3	5.7	5.0	4.8	6.5	6.3	5.8	12.3	8.0
	+ MixKV	26.6	14.6	11.3	4.8	**20.1**	11.3	5.7	5.0	5.7	5.7	6.3	6.3	12.6	8.0
	+ BACON	**26.6**	**15.4**	**12.0**	**5.0**	19.5	**13.9**	**5.7**	**5.0**	**5.7**	**6.7**	**6.8**	**6.3**	**12.7**	**8.7**
PyramidKV	Base	24.5	**13.9**	8.2	6.6	20.1	9.8	3.6	5.0	6.1	6.1	6.2	5.3	11.5	7.8
	+ MixKV	24.5	13.2	8.7	**7.0**	19.1	10.3	3.6	5.0	6.5	6.1	**6.6**	5.8	11.5	7.9
	+ BACON	**24.8**	13.5	**8.9**	6.7	**20.1**	**10.3**	**4.3**	**5.7**	**7.8**	**6.2**	6.3	**5.8**	**12.0**	**8.0**
SparseMM	Base	22.7	15.8	9.2	4.4	18.6	9.8	4.6	4.3	7.0	4.5	**8.7**	5.8	11.9	7.4
	+ MixKV	21.5	14.6	10.2	4.4	17.0	10.3	2.9	4.3	7.4	3.9	8.2	5.8	11.2	7.2
	+ BACON	**22.7**	**16.9**	**10.2**	**4.5**	**19.6**	**10.3**	**4.9**	**5.0**	**7.6**	**5.2**	8.3	**8.2**	**12.2**	**8.4**

Table 4: **Results on LongBench long context understanding benchmark with Mistral-7B-Instruct-v0.2.**

Methods	Single-Doc QA			Multi-Doc QA			Summarization			Few-shot			Synthetic		Code		Avg.
	NrtvQA	Qasper	MF-en	HotpotQA	2WikiMQA	Musique	GovReport	QMSum	MultiNews	TREC	TriviaQA	SAMSum	PCount	PRe	Lcc	RB-P	
Full KV	26.77	32.51	49.36	43.58	27.35	18.86	33.09	24.38	27.03	71.00	86.23	42.99	2.89	86.98	47.16	48.03	41.76
KV Cache Budget = 1024																	
SnapKV	25.06	28.92	49.17	**40.41**	26.11	18.14	25.91	23.99	25.73	67.00	86.24	41.57	2.98	**87.48**	46.13	46.09	40.06
+ MixKV	24.73	29.88	48.62	39.68	26.41	18.56	26.15	23.70	26.16	69.00	86.25	43.25	3.21	85.84	**46.27**	46.32	40.25
+ BACON	**25.89**	**31.15**	**49.42**	40.27	**26.47**	**18.94**	**26.32**	**24.38**	**26.37**	**71.00**	**86.43**	**43.56**	**3.36**	87.21	46.19	**46.68**	**40.85**
AdaKV	25.28	31.02	48.53	41.06	26.49	19.28	25.74	24.02	25.55	68.50	86.28	42.43	2.94	88.10	46.35	46.58	40.51
+ MixKV	25.60	30.94	48.79	40.72	26.56	19.25	**26.74**	23.98	25.68	69.00	86.25	**43.39**	2.96	86.88	46.78	46.01	40.60
+ BACON	**26.99**	**31.11**	**48.90**	**41.43**	**26.68**	**19.87**	26.65	**24.57**	**25.91**	**71.00**	**86.49**	43.33	**3.03**	**88.23**	**46.94**	**46.64**	**41.11**
PyramidKV	24.68	27.34	48.53	39.91	25.78	18.71	25.73	23.50	25.37	68.50	85.80	41.29	2.66	**86.73**	45.60	46.30	39.78
+ MixKV	23.72	29.85	48.12	39.17	**26.78**	19.29	26.67	23.59	26.47	70.00	85.63	**43.00**	3.13	84.05	46.10	45.50	40.07
+ BACON	**24.86**	**30.94**	**48.74**	**40.51**	26.73	**19.52**	**26.92**	**23.67**	**27.48**	**71.00**	**86.14**	42.98	**3.21**	86.28	**46.22**	**46.63**	**40.74**
KV Cache Budget = 512																	
SnapKV	24.15	26.67	48.94	37.38	25.90	17.18	23.80	22.83	24.32	65.50	85.88	42.07	3.17	87.13	45.01	45.80	39.11
+ MixKV	24.04	27.03	48.18	37.79	26.03	17.54	24.56	23.43	**25.20**	66.50	86.16	42.92	3.26	87.04	45.39	45.82	39.43
+ BACON	**25.21**	**27.81**	**49.17**	**38.47**	**26.30**	**17.74**	**24.83**	**23.95**	24.83	**69.00**	**86.36**	**42.92**	**3.44**	**87.20**	**45.85**	**45.96**	**39.94**
AdaKV	24.67	26.93	48.39	38.47	**26.07**	17.14	23.80	23.50	24.10	66.00	86.11	42.00	**3.29**	87.33	45.70	**46.02**	39.35
+ MixKV	24.30	27.85	48.40	38.33	25.76	**18.34**	24.39	23.61	24.80	67.50	85.92	42.49	2.99	87.12	**46.53**	45.77	39.63
+ BACON	**24.85**	**28.50**	**48.84**	**38.49**	25.83	18.30	**24.83**	**23.62**	**25.34**	**70.00**	**86.35**	**43.30**	3.14	**87.64**	45.96	45.81	**40.05**
PyramidKV	**23.39**	24.80	47.56	38.23	25.29	17.32	23.47	23.02	23.67	66.00	85.31	41.48	2.86	86.30	45.35	43.54	38.60
+ MixKV	22.99	25.38	47.89	38.21	24.09	17.76	23.98	22.78	**24.92**	67.50	85.41	41.73	3.19	86.41	45.24	44.35	38.86
+ BACON	23.06	**27.29**	**48.13**	**38.41**	**26.25**	**18.21**	**24.06**	**23.06**	24.76	**69.50**	**86.07**	**42.33**	**3.22**	**86.91**	**45.82**	**44.45**	**39.47**

marks. Beyond static image benchmarks, we evaluate BACON on video understanding tasks, where relevant evidence can be distributed across frames. As shown in Table 2, BACON improves compressed inference on both VATEX and NextQA across different compression backbones and cache budgets. On VATEX, BACON yields broad gains across methods and budgets, suggesting better preservation of fine-grained visual-temporal cues. On NextQA, it improves WUPS in most settings, indicating that the calibrated retention signal also benefits video question answering. These results show that BACON generalizes beyond image-level grounding while remaining compatible with diverse compression strategies. A few metric-level fluctuations suggest that some videos may benefit from broader temporal coverage retained by the base score. Overall, BACON provides a complementary calibration signal for compressed video-language inference.

Performance on GUI grounding benchmarks. GUI grounding requires models to identify fine-grained interface elements, making it sensitive to information loss under KV compression. We evaluate BACON on ScreenSpot with Qwen2-VL-7B to test whether it preserves interface evidence under compressed caches. As shown in Table 3, BACON improves average performance across different compression backbones under both 128 and 64 budgets, with stronger gains under the tighter budget. This suggests that BACON helps retain weak but critical GUI evidence that may be diluted by observation window aggregation. Minor drops on a few subsets reflect the sensitivity of GUI grounding to small retention changes under tight budgets, rather than a systematic limitation. These results validate BACON as a complementary calibration signal for memory-efficient GUI agent deployment.

Performance on long-context text benchmarks. To evaluate BACON beyond multimodal tasks, we further test it on LongBench with Mistral-7B-Instruct-v0.2. As shown in Table 4, BACON improves average performance across compression backbones and cache budgets, showing that

Table 5: **Ablation and efficiency analysis of BACON.** Left: component ablation on LLaVA-NeXT-Mistral-7B under budget 64. Right: efficiency comparison with fixed input length 32000.

Backbone	E	L	T	Budget 64				
				ChartQA	DocVQA	TextVQA	MMMU	TextCaps
SnapKV	✗	✗	✗	47.2	46.2	58.9	34.8	0.442
	✓	✗	✓	47.8	52.1	62.0	34.8	0.482
	✓	✓	✗	47.9	51.7	61.6	34.8	0.465
	✗	✓	✓	46.8	50.7	60.9	34.8	0.508
	✓	✓	✓	**49.2**	**53.9**	**62.8**	**34.8**	**0.512**
AdaKV	✗	✗	✗	47.4	47.6	57.8	34.8	0.441
	✓	✗	✓	48.7	52.3	59.3	34.8	0.495
	✓	✓	✗	47.9	54.8	60.9	34.8	0.477
	✗	✓	✓	47.7	51.4	59.5	34.8	0.469
	✓	✓	✓	**49.0**	**55.2**	**61.5**	**34.8**	**0.510**
SparseMM	✗	✗	✗	49.4	57.5	65.2	34.8	0.490
	✓	✗	✓	49.6	58.1	66.2	34.7	0.553
	✓	✓	✗	49.4	58.3	66.7	34.9	0.540
	✗	✓	✓	49.2	57.9	66.4	34.7	0.547
	✓	✓	✓	**49.8**	**58.3**	**67.3**	**34.9**	**0.572**

Backbone	Method	Lat.	Mem.
Full KV	Base	63.93	22.27
SnapKV	Base	28.60	18.27
	+ MixKV	28.66	18.27
	+ BACON	28.68	18.28
SparseMM	Base	28.62	17.74
	+ MixKV	28.15	17.74
	+ BACON	28.01	17.75
PyramidKV	Base	28.52	18.27
	+ MixKV	28.72	18.27
	+ BACON	28.21	18.29
AdaKV	Base	28.67	17.74
	+ MixKV	28.40	17.74
	+ BACON	28.34	17.76

boundary-aware calibration is not limited to visual tokens and can also refine token scoring in language-only settings. The rare regressions are mainly concentrated in Synthetic and Code tasks, where performance often depends on exact positions, copied strings, or short-range syntactic dependencies. In these cases, small changes in retained tokens can affect task-specific exactness, making boundary calibration less uniformly beneficial than in semantic understanding tasks. Nevertheless, the overall gains show that BACON remains effective in text-only settings, while its larger improvements on MLLM benchmarks further support our motivation that sparse multimodal evidence is particularly vulnerable to observation window aggregation. Additional results on Llama3.1-8B-Instruct are provided in B.1.

5.3 Ablation Study & Analysis

Ablation of BACON components. Table 5 ablates the three key components of BACON, including boundary evidence E, intra-layer coherence L, and inter-layer persistence T, under budget 64. Across different compression backbones, the full BACON consistently achieves the best performance, demonstrating the complementarity of these components under aggressive KV compression. The gains are especially pronounced on DocVQA, TextVQA, and TextCaps, indicating that OCR-intensive tasks are highly sensitive to the loss of sparse visual evidence caused by observation window aggregation. Here, E provides a boundary-aware correction for recovering evidence weakened by window aggregation, while L and T strengthen intra-layer coherence and inter-layer persistence to reduce answer-irrelevant high-attention responses. The weaker performance of partial variants further suggests that effective token retention requires both boundary-sensitive evidence discovery and structure-aware evidence calibration. Additional ablation results on Qwen2-VL-7B and other cache budgets are provided in B.2. We also provide detailed hyperparameter sensitivity analysis in B.3.

Efficiency Analysis of BACON. Table 5 evaluates the inference latency and peak memory consumption of BACON when integrated with representative KV cache compression backbones. BACON preserves the efficiency benefits of compressed inference and introduces negligible additional runtime or memory overhead over the original compression methods. This is because BACON only refines token-retention scores during prefill and does not change the compressed cache size or decoding procedure. Therefore, its accuracy gains are obtained without additional inference cost.

6 Conclusion

In this work, we analyze attention-based KV cache compression in MLLMs and identify a key limitation of observation window attention: its aggregation can dilute sparse visual evidence under tight budgets despite providing stable token-importance estimates. Motivated by this insight, we propose BACON, a plug-and-play boundary attention calibration method that refines observation window retention scores with last-query evidence. BACON further suppresses noisy boundary saliency through intra-layer coherence and inter-layer persistence, thereby producing a stronger retention signal without modifying the original compression pipeline. Extensive experiments spanning multimodal understanding, video reasoning, GUI grounding, and long-context text tasks across different model scales, architectures, compression methods, and cache budgets show that BACON consistently improves existing KV compression methods while requiring no extra method modification or additional inference cost.

7 Limitations

BACON is designed as a training-free and plug-and-play mechanism, enabling broad compatibility with existing KV cache compression pipelines without additional model updates. This design choice naturally focuses BACON on improving token-retention quality under a fixed compression policy. As a result, it does not explicitly exploit training signals to adapt retention behavior to sample-level evidence distributions, task characteristics, or input complexity. Incorporating BACON into training could further enable the model to learn more adaptive compression strategies for different samples and tasks. In addition, KV cache compression methods integrated with BACON can be applied to many latency-sensitive scenarios, such as real-time multimodal assistants, GUI agents, embodied systems, and edge-device deployment. However, their use should be carefully controlled in high-precision or safety-critical applications. If compression is applied too aggressively or without appropriate validation, answer-critical visual or textual evidence may be discarded, leading to degraded model performance and potentially affecting downstream functionality. Future deployment should therefore consider task-specific reliability requirements and provide safeguards, such as budget constraints, confidence checks, or fallback mechanisms to less compressed inference when necessary.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137.

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*.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S Kevin Zhou. 2026. Ada-kv: Optimizing kv cache eviction by adaptive budget allocation for efficient llm inference. *Advances in Neural Information Processing Systems*, 38:113152–113188.

Yu Fu, Zefan Cai, Abedelkadir Asi, Wayne Xiong, Yue Dong, and Wen Xiao. 2025. Not all heads matter: A head-level kv cache compression method with integrated retrieval and reasoning. In *International Conference on Learning Representations*, volume 2025, pages 99269–99290.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Zonghao Guo, Ruyi Xu, Yuan Yao, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, and Gao Huang. 2024. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. In *European Conference on Computer Vision*, pages 390–406. Springer.

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.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. 2025. See what you are told: Visual attention sink in large multimodal models. *arXiv preprint arXiv:2503.03321*.

Jang-Hyun Kim, Jinuk Kim, Sangwoo Kwon, Jae W Lee, Sangdoo Yun, and Hyun Oh Song. 2026a. Kvzip: Query-agnostic kv cache compression with context reconstruction. *Advances in Neural Information Processing Systems*, 38:167563–167591.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. 2026b. Infinipot-v: Memory-constrained kv cache compression for streaming video understanding. *Advances in Neural Information Processing Systems*, 38:138983–139013.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. 2025. Screenspot-pro: Gui grounding for professional high-resolution computer use. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 8778–8786.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024b. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. Llava-next: Improved reasoning, ocr, and world knowledge.

Xuyang Liu, Xiyan Gui, Yuchao Zhang, and Linfeng Zhang. 2025. Mixing importance with diversity: Joint optimization for kv cache compression in large vision-language models. *arXiv preprint arXiv:2510.20707*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. 2024. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12585–12602.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pages 2263–2279.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2025. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22857–22867.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. 2020. Textcaps: a dataset for image captioning with reading comprehension. In *European conference on computer vision*, pages 742–758. Springer.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. 2024. Look-m: Look-once optimization in kv cache for efficient multimodal long-context inference. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4065–4078.

Jiahui Wang, Zuyan Liu, Yongming Rao, and Jiwen Lu. 2025. Sparsemm: Head sparsity emerges from visual concept responses in mllms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23177–23187.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4581–4591.

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

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9777–9786.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9556–9567.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, and 1 others. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A Detailed Experiment Setting

A.1 Implementation Details.

All experiments were conducted on NVIDIA RTX 3090 and NVIDIA H800 GPUs. BACON is implemented as a plug-and-play modification to existing KV cache compression methods and is applied only during the prefill-stage cache selection. It does not require model retraining, additional supervision, or changes to the original model parameters. For each compression method, we keep its original layer-wise and head-wise cache budget allocation unchanged, and only modify the within-head token retention score. Unless otherwise specified, all results are obtained under the same decoding settings as the corresponding baseline.

A.2 Model Details.

We introduce more details on the models we used in our experiments:

LLaVA-NeXT-Mistral-7B. LLaVA-NeXT-Mistral-7B is a representative open-source MLLM built upon the Mistral-7B language backbone. It follows the common vision-language architecture that combines a pretrained visual encoder, a multimodal projector, and an autoregressive LLM. Compared with earlier LLaVA models, LLaVA-NeXT improves high-resolution image understanding, OCR ability, and visual reasoning by using dynamic high-resolution image processing and stronger instruction-tuning data. In our experiments, this model serves as a strong 7B-scale MLLM baseline for evaluating KV cache compression under image-text reasoning tasks.

Qwen2-VL-7B-Instruct. Qwen2-VL-7B-Instruct is an instruction-tuned vision-language model from the Qwen2-VL series. It supports flexible visual inputs, including images with different resolutions and aspect ratios, and is designed for fine-grained visual understanding, document comprehension, OCR-related reasoning, and general multimodal dialogue. The model adopts a unified multimodal generation framework in which visual tokens are injected into the language model and processed together with textual instructions. We use Qwen2-VL-7B-Instruct as a high-performing 7B-scale MLLM to evaluate whether BACON remains effective on modern multimodal architectures with stronger visual perception capability.

Qwen3-VL-30B-A3B-Instruct. Qwen3-VL-30B-A3B-Instruct is a larger-scale multimodal model from the Qwen3-VL family. It is designed for comprehensive vision-language understanding, including image understanding, video understanding, spatial reasoning, grounding, and long-form visual comprehension. Unlike dense 7B-scale MLLMs, this model adopts a larger mixture-of-experts-style architecture, with around 30B total parameters and a smaller number of active parameters during inference. We include Qwen3-VL-30B-A3B-Instruct to examine whether BACON can generalize beyond 7B-scale dense MLLMs and remain effective for larger multimodal backbones with stronger reasoning capacity.

InternVL3-8B. InternVL3-8B is an advanced open-source MLLM from the InternVL3 series. Different from many MLLMs that mainly adapt a pretrained text-only LLM to visual inputs, InternVL3 emphasizes native multimodal pretraining, where multimodal and linguistic abilities are jointly acquired during training. The model is designed to improve multimodal perception, reasoning, GUI understanding, document analysis, and other vision-language tasks. We use InternVL3-8B as another 8B-scale MLLM backbone to test the robustness of BACON across different multimodal training paradigms and architectural designs.

Mistral-7B-Instruct-v0.2. Mistral-7B-Instruct-v0.2 is an instruction-tuned text-only LLM based on the Mistral-7B architecture. It is a compact yet strong decoder-only transformer model and has been widely used in long-context language understanding and generation benchmarks. Since it does not contain a visual encoder or multimodal projector, all input tokens are textual. We include this model to evaluate whether the proposed boundary-aware KV retention strategy is also beneficial for text-only long-context inference, rather than being limited to multimodal visual-token compression.

Llama-3.1-8B-Instruct. Llama-3.1-8B-Instruct is an instruction-tuned text-only LLM from the Llama 3.1 family. It is an autoregressive decoder-only transformer optimized for multilingual dialogue and general text generation. The model uses grouped-query attention to improve inference scalability and supports long-context processing. In our experiments, Llama-3.1-8B-Instruct provides an additional text-only backbone for evaluating BACON on long-context language tasks, allowing us to test whether the proposed retention mechanism generalizes across different LLM families.

A.3 Benchmark Details.

We provide details for each benchmark used in our experiments:

DocVQA. DocVQA evaluates visual question answering over document images. Given a document image and a natural-language question, the model is required to locate relevant textual or structural evidence from the document and generate the correct answer. This benchmark emphasizes OCR ability, document layout understanding, and fine-grained evidence localization. We report ANLS as the evaluation metric.

ChartQA. ChartQA evaluates question answering over chart images. It requires the model to understand chart structures, recognize visual elements such as bars, lines, legends, axes, and labels, and perform numerical or logical reasoning based on the chart content. We report relaxed accuracy, which allows minor numerical deviations from the reference answer.

TextVQA. TextVQA evaluates question answering over natural images containing scene text. The model must jointly recognize visual content, read text appearing in the image, and reason over both visual and textual evidence to answer the question. This benchmark is useful for measuring whether compressed KV caches preserve sparse but answer-critical OCR evidence.

TextCaps. TextCaps evaluates image captioning with reading comprehension. Unlike standard image captioning benchmarks, many correct captions require the model to incorporate text appearing in the image. Therefore, TextCaps tests both visual description ability and scene-text understanding.

MMMU. MMMU is a multi-discipline multimodal reasoning benchmark built from college-level materials such as exams, quizzes, and textbooks. It covers diverse subjects including science, engineering, medicine, business, and the humanities. Solving MMMU requires models to combine visual perception, domain knowledge, and multi-step reasoning. We use the testmini split in our experiments.

VATEX. VATEX is a large-scale video captioning benchmark. Given a video clip, the model is required to generate a natural-language description of the visual content. Since useful evidence is distributed across multiple frames, VATEX is used to evaluate the effectiveness of BACON under longer multimodal contexts. We report standard captioning metrics, including CIDEr, BLEU-4, METEOR, and ROUGE-L.

NExT-QA. NExT-QA is a video question answering benchmark designed to evaluate temporal and causal reasoning. The questions often require understanding action sequences, object interactions, temporal order, and causal relations across video frames. This benchmark allows us to test whether BACON can preserve temporally distributed evidence in video-language reasoning.

ScreenSpot. ScreenSpot evaluates GUI grounding ability. Given a screenshot and a natural-language instruction, the model must identify the target user-interface element. The benchmark covers mobile, web, and desktop environments, and includes both text-based and icon/widget-based elements. This setting is challenging because the target evidence is often small, spatially localized, and visually similar to nearby interface components.

LongBench. LongBench evaluates long-context understanding for text-only LLMs. It includes tasks from single-document question answering, multi-document question answering, summarization, few-shot learning, synthetic reasoning, and code completion. In our experiments, LongBench is used to assess whether BACON also generalizes to text-only long-context inference, where the input consists entirely of textual tokens rather than visual tokens.

A.4 Baseline Details.

We provide more details on the baseline used to compare with our methods in the experiment:

SnapKV. SnapKV is a training-free KV cache compression method that exploits the observation that attention patterns during prefilling can reveal tokens that are likely to remain important during decoding. It estimates token importance using attention from a local observation window near the end of the prompt and retains the most salient KV pairs under a given budget. SnapKV provides a simple and efficient attention-based compression baseline.

PyramidKV. PyramidKV compresses the KV cache with a pyramid-like budget allocation strategy across transformer layers. Instead of assigning the same retention budget to every layer, it allocates different budgets according to layer depth, reflecting the intuition that different layers contribute differently to context retention and generation. This baseline evaluates whether BACON can improve token selection under non-uniform layer-wise budget allocation.

Table 6: **Results on LongBench long context understanding benchmark with Llama3.1-8B-Instruct.**

Methods	Information Localization						Information Aggregation						Synthetic / Code				Avg.
	NrtvQA	Qasper	MF-en	HotpotQA	2WikiMQA	Musique	GovReport	QMSum	MultiNews	TREC	TriviaQA	SAMSum	PCount	PRe	Lcc	RB-P	
Full KV	28.29	45.53	54.94	56.02	46.66	31.34	35.19	25.28	27.16	72.50	91.65	43.59	8.91	99.50	52.73	49.20	48.03
KV Cache Budget = 1024																	
SnapKV	27.54	44.04	54.64	55.24	46.18	30.70	28.30	24.42	25.78	68.00	91.83	42.44	8.46	99.50	52.27	48.72	46.75
+ MixKV	27.61	44.06	54.47	55.66	45.76	31.11	28.48	24.59	26.41	69.50	91.74	42.58	7.76	99.50	**52.88**	48.64	46.92
+ BACON	**27.83**	**44.31**	**54.81**	**56.02**	**46.32**	**31.56**	**28.72**	**24.66**	**26.59**	**72.00**	**91.90**	**43.07**	**8.59**	**99.50**	52.51	**48.90**	**47.33**
AdaKV	28.03	43.34	**55.36**	55.67	45.90	31.23	28.47	24.25	25.94	68.50	91.62	42.25	**8.43**	99.50	52.28	49.23	46.88
+ MixKV	**28.21**	43.38	54.71	55.79	46.10	**31.55**	29.06	24.09	26.15	70.50	91.73	43.11	8.26	99.50	**52.68**	49.25	47.13
+ BACON	28.13	**44.05**	55.27	**56.04**	**46.64**	31.52	**29.31**	**24.28**	**26.31**	**72.50**	**91.73**	**43.63**	8.29	**99.50**	52.62	**49.34**	**47.45**
PyramidKV	28.06	42.63	54.19	55.65	46.23	31.46	27.67	24.85	25.79	68.00	91.79	41.57	8.34	99.50	51.35	48.01	46.57
+ MixKV	28.54	**43.18**	54.67	55.56	46.06	30.82	28.26	24.76	**26.35**	69.50	91.34	43.05	8.18	99.50	51.58	**48.47**	46.86
+ BACON	**28.57**	42.89	**54.81**	**55.75**	**46.38**	**31.73**	**28.45**	**25.12**	25.81	**72.50**	**92.34**	**43.14**	**8.49**	**99.50**	**51.63**	48.38	**47.22**
KV Cache Budget = 512																	
SnapKV	27.40	38.23	53.49	55.04	45.30	29.92	25.48	24.21	24.39	64.00	92.05	41.92	8.17	99.50	52.18	47.46	45.55
+ MixKV	27.46	38.58	53.18	55.66	45.47	30.48	25.63	24.47	24.97	68.00	92.04	**43.25**	7.67	99.50	51.86	47.52	45.98
+ BACON	**27.62**	**38.79**	**53.73**	**55.72**	**45.66**	**31.10**	**25.94**	**24.92**	**25.25**	**71.50**	**92.26**	43.05	**8.34**	**99.50**	**52.22**	**47.73**	**46.46**
AdaKV	27.38	40.76	**53.41**	55.67	45.53	29.81	25.96	24.38	24.61	66.00	92.35	42.27	8.01	99.50	52.75	48.26	46.04
+ MixKV	27.36	42.28	53.04	55.78	45.39	29.95	26.06	24.64	24.46	69.50	92.07	43.24	**8.09**	99.50	52.72	48.44	46.41
+ BACON	**27.62**	**42.53**	53.20	**55.99**	**45.68**	**30.04**	**26.19**	**24.83**	**24.84**	**72.00**	**92.71**	**43.37**	8.04	**99.50**	**53.09**	**48.59**	**46.76**
PyramidKV	27.15	40.52	53.15	55.19	45.57	29.90	25.29	24.19	24.31	62.50	91.30	42.08	8.57	99.50	50.79	46.31	45.40
+ MixKV	27.11	41.12	53.34	55.40	45.08	30.45	**25.64**	24.49	24.39	67.00	91.06	42.71	8.59	99.50	50.74	46.67	45.83
+ BACON	**27.89**	**41.37**	**53.67**	**56.07**	**46.46**	**30.62**	25.57	**24.70**	**24.42**	**71.50**	**91.77**	**43.65**	**8.62**	**99.50**	**51.26**	**46.82**	**46.49**

AdaKV. AdaKV performs adaptive KV cache compression by dynamically assigning retention budgets according to token or head importance. Compared with fixed-budget strategies, AdaKV aims to preserve more KV pairs in more informative regions while applying stronger compression to less important ones. We include AdaKV to evaluate whether BACON remains effective when the compression budget is adaptively distributed rather than uniformly assigned.

SparseMM. SparseMM is a multimodal KV cache compression baseline designed for efficient MLLM inference. It exploits sparsity in multimodal attention and applies head-wise KV retention to reduce redundant visual and textual tokens while preserving task-relevant evidence. Since SparseMM provides optimized head-wise budget allocation, it is a strong baseline for evaluating whether BACON can further improve within-head token selection without changing the original budget assignment.

MixKV. MixKV is a plug-and-play KV cache compression method for large vision-language models. It selects KV pairs by jointly considering token importance and diversity, aiming to preserve both highly attended tokens and semantically diverse contextual information. MixKV further adopts head-wise adaptive mixing to balance these criteria across attention heads. We use MixKV as a strong plug-in baseline and compare BACON against it under the same models, budgets, and evaluation settings.

A.5 Evaluation Metrics Details.

We follow the official evaluation protocol of each benchmark. Specifically, DocVQA is evaluated with ANLS, which measures normalized string similarity and accounts for minor OCR or formatting variations in document question answering. ChartQA uses relaxed accuracy, allowing small numerical deviations from the reference answer. TextVQA, MMMU, NExT-QA, ScreenSpot, and LongBench are evaluated with accuracy-based metrics according to their standard answer-matching or task-specific scoring rules. For captioning benchmarks, TextCaps and VATEX are evaluated with standard generation metrics, including CIDEr, BLEU-4, METEOR, and ROUGE-L, which measure caption quality from complementary perspectives such as n-gram precision, semantic overlap, and recall-oriented similarity. These metrics together assess answer correctness, caption generation quality, visual grounding accuracy, temporal reasoning, and long-context understanding.

B Additional Experiment Results

B.1 Additional results on LongBench

Table 6 provides additional LongBench results with LLaMA under KV cache budgets of 1024 and 512. The results further support the generality of BACON beyond multimodal benchmarks. Across different compression backbones, BACON consistently improves the average performance over the corresponding base methods and MixKV-enhanced variants, showing that boundary-evidence calibra-

Table 7: **Additional ablation results of BACON on LLaVA-NeXT-Mistral-7B under budget 128.**

Backbone	E	L	T	Budget 128				
				ChartQA	DocVQA	TextVQA	MMMU	TextCaps
SnapKV	✗	✗	✗	48.4	55.2	63.0	34.9	0.560
	✓	✗	✓	49.3	56.3	63.7	34.9	0.632
	✓	✓	✗	49.1	57.1	65.4	34.8	0.624
	✗	✓	✓	48.9	56.6	64.6	34.9	0.644
	✓	✓	✓	**49.6**	**57.8**	**66.2**	**34.9**	**0.676**
AdaKV	✗	✗	✗	48.8	56.1	62.7	34.9	0.568
	✓	✗	✓	45.0	57.6	65.1	34.9	0.606
	✓	✓	✗	44.5	57.4	65.4	34.7	0.626
	✗	✓	✓	45.3	57.0	63.2	34.9	0.619
	✓	✓	✓	**50.2**	**58.4**	**65.5**	**34.9**	**0.641**
SparseMM	✗	✗	✗	49.8	58.9	67.4	34.7	0.600
	✓	✗	✓	50.2	59.2	**67.8**	34.7	0.597
	✓	✓	✗	50.0	58.9	67.5	34.7	0.604
	✗	✓	✓	49.9	58.6	66.7	34.7	0.605
	✓	✓	✓	**50.4**	**59.3**	**67.8**	**34.8**	**0.626**

Table 8: **Additional ablation results of BACON on Qwen2-VL-7B under budgets 64 and 128.**

Backbone	E	L	T	Budget 64					Budget 128				
				ChartQA	DocVQA	TextVQA	MMMU	TextCaps	ChartQA	DocVQA	TextVQA	MMMU	TextCaps
SnapKV	✗	✗	✗	66.2	70.1	70.3	49.6	0.787	69.6	82.1	77.0	49.8	1.141
	✓	✗	✓	68.1	84.0	77.1	50.0	1.114	**70.4**	90.6	81.9	49.8	1.410
	✓	✓	✗	68.2	83.5	76.4	49.9	1.151	69.7	88.3	81.4	**50.0**	1.401
	✗	✓	✓	67.5	84.7	77.3	50.0	1.133	69.9	87.7	81.8	49.8	1.408
	✓	✓	✓	**69.6**	**85.5**	**78.2**	**50.0**	**1.178**	70.2	**91.5**	**82.6**	49.9	**1.426**
AdaKV	✗	✗	✗	66.6	69.4	70.8	49.6	0.771	69.6	81.3	75.9	49.7	1.099
	✓	✗	✓	67.8	85.6	77.0	49.8	1.103	70.1	88.6	79.2	49.8	1.359
	✓	✓	✗	67.9	84.3	76.4	49.8	**1.146**	69.4	85.3	80.4	49.7	1.363
	✗	✓	✓	68.3	81.7	77.6	49.6	1.111	69.8	86.5	81.1	49.7	1.348
	✓	✓	✓	**69.6**	**86.2**	**78.5**	**49.8**	1.144	**70.2**	**91.1**	**81.2**	**49.8**	**1.371**
SparseMM	✗	✗	✗	69.6	87.3	76.9	49.6	1.044	70.0	91.4	82.1	49.8	1.427
	✓	✗	✓	69.9	90.5	81.1	49.8	1.388	**70.4**	92.3	82.3	49.6	1.471
	✓	✓	✗	70.0	91.9	80.7	49.8	1.415	70.1	93.0	82.1	49.8	1.503
	✗	✓	✓	69.8	89.4	81.2	49.6	1.407	69.7	92.3	82.3	49.8	1.498
	✓	✓	✓	**70.2**	**92.0**	**81.6**	**49.8**	**1.432**	**70.4**	**93.2**	**82.4**	**49.8**	**1.511**

tion remains effective for long-context language understanding. The gains are particularly stable under the smaller budget of 512, where aggressive compression makes evidence retention more challenging. This suggests that BACON can better preserve tokens that become important near the generation boundary, rather than relying only on window-averaged attention signals.

We also observe that the improvements are not limited to a single task group. BACON achieves consistent gains on information localization tasks, such as QA over single-document and multi-document inputs, indicating that calibrated boundary evidence helps recover fine-grained relevant context. Meanwhile, it also improves many information aggregation and few-shot tasks, suggesting that the proposed calibration does not sacrifice global context modeling. A few metrics show smaller gains or occasional non-best results, especially in tasks where the baseline already performs close to the Full KV upper bound or where the score is saturated, such as passage retrieval. Overall, these appendix results demonstrate that BACON is not specific to visual-token compression, but provides a broadly applicable score calibration mechanism for KV cache compression.

B.2 Additional ablation results

We provide additional ablation results in Tables 7 and 8 to further validate the component design of BACON beyond the main setting. The full BACON variant consistently achieves the best or near-best performance across different models, compression backbones, benchmarks, and cache budgets, confirming the complementary roles of boundary evidence E, local coherence L, and cross-layer trace support T. Removing any component often weakens performance or leads to less stable results, especially on evidence-sensitive tasks such as DocVQA, TextVQA, and TextCaps. These additional results further demonstrate that BACON's component design remains robust across broader model and budget settings.

B.3 Hyperparameter Sensitivity Analysis

Table 9 reports the sensitivity of BACON to the inter-layer persistence range m and the intra-layer neighborhood radius r on SnapKV under compression budget at 64. Overall, BACON exhibits stable performance across a moderate range of hyperparameter values, indicating that its gains do not rely on a narrowly tuned configuration. For the inter-

Table 9: **Parameter sensitivity analysis of BACON on SnapKV.** Left: sensitivity to the inter-layer persistence range m. Right: sensitivity to the intra-layer neighborhood r. The default settings used in our main experiments are marked with *.

m	Budget 64		
	DocVQA	**ChartQA**	**TextVQA**
0	51.7	47.9	61.6
2	53.4	48.2	62.1
4^*	53.9	49.2	62.8
6	53.9	48.9	62.9
8	53.5	48.8	62.7
10	52.0	47.7	62.0

r	Budget 64		
	DocVQA	**ChartQA**	**TextVQA**
0	52.1	47.8	62.0
3	53.7	48.8	62.8
5^*	53.9	49.2	62.8
7	53.7	49.1	62.9
9	53.7	48.9	62.5

layer range m, performance improves substantially when moving from $m = 0$ to moderate values, suggesting that inter-layer persistence provides useful structural validation for boundary evidence. The results remain close when m varies from 2 to 8, with only minor fluctuations across DocVQA, ChartQA, and TextVQA. Similarly, for the neighborhood radius r, BACON consistently outperforms the no-neighborhood setting and maintains comparable results for $r = 3, 5, 7$, showing that local coherence is beneficial as long as the neighborhood is not overly restricted or overly broad. Performance degradation appears mainly at excessively large ranges, such as $m = 10$ or $r = 9$, where the support signals may become less discriminative by incorporating irrelevant surrounding or cross-layer information. These results demonstrate that BACON is robust to reasonable hyperparameter choices, while the default setting achieves a favorable balance between evidence recovery and noise suppression.

B.4 Additional visualization results

Here we provide more visualization results on how BACON improve visual evidence retention compared to baseline methods, as shown in Fig. 5

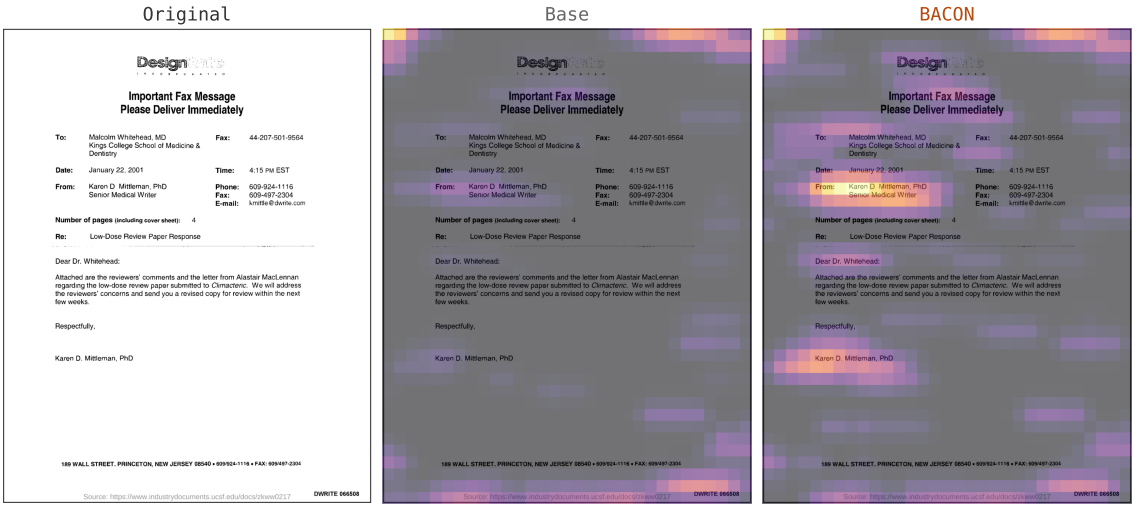
Q: What was the final event ?

GT: questions and answers | Questions and Answers

Base: Questions (Wrong)

BACON: Questions and Answers (Correct)

Figure 5: Additional visualizations comparing evidence importance estimation between window attention and BACON.



Q: Who is the sender of the Fax?

GT: Karen D Mittleman, PhD

Base: Karen D. Mccormick (Wrong)

BACON: Karen D. Mittleman (Correct)

Figure 5: Additional visualizations comparing evidence importance estimation between window attention and BACON.

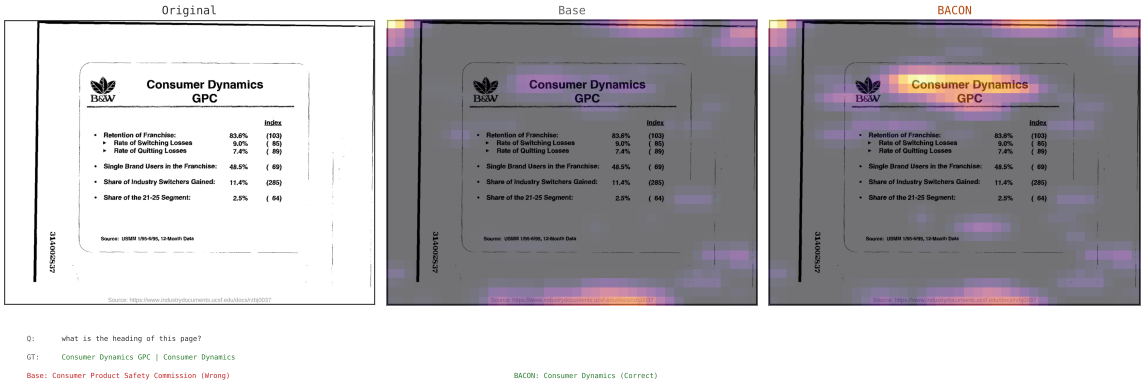


Figure 5: Additional visualizations comparing evidence importance estimation between window attention and BACON.

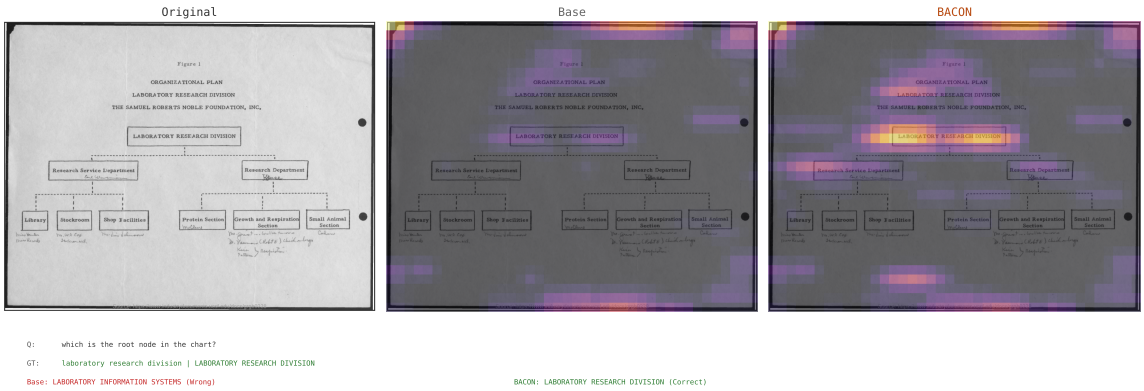


Figure 5: Additional visualizations comparing evidence importance estimation between window attention and BACON.