WHAT TO FORGET: CONTRIBUTION-AWARE MEMORY FOR STREAMING VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal Large Language Models (MLLMs) have made impressive strides in offline video understanding tasks. However, adapting them to streaming settings is far from straightforward, as the ever-growing visual context quickly exhausts GPU memory. While prior work mitigates this by compressing the cache and retaining tokens with high attention scores, we find that these scores do not always reflect a token's contribution to the model output. In this paper, we formalize token contribution through the Leave-one-out (LOO) reference, defined as the shift in model output distribution caused by removing a single token. Utilizing the self-prediction objective, we derive the first-order KV Taylor score that shows stronger agreement with LOO than attention-based scores. Building upon this, we thus propose CALM, i.e., Contribution-Aware Layer-wise Memory, a training-free caching strategy that allocates retention budgets based on KV Taylor scores and applies conditional novelty reranking to suppress redundant tokens. To retain long-range context, we further maintain a lightweight text memory that stores compact summaries of past video chunks. Experiments on StreamingBench and OVO-Bench demonstrate that CALM achieves state-of-the-art streaming video understanding results, with gains of 6.25% on StreamingBench and 11.82% on OVO-Bench (12.77% on the Real-Time and 10.87% on the Backward Tracing).

1 INTRODUCTION

Multimodal Large Language Models (MLLMs) have achieved remarkable performance in video understanding tasks (Hurst et al., 2024; Bai et al., 2025b;a). Streaming video understanding, however, requires the model to process a continuous stream and preserve potentially relevant visual evidence without knowing what a future query will ask. This forces the model to retain all potentially relevant evidence within a bounded key-value cache. Without compression, the cache length increases linearly with the number of observed frames (Chen et al., 2024a; Lin et al., 2026; Niu et al., 2025). Thus, the core challenge shifts to cache management: with a fixed budget, the model must continuously decide which tokens are worth keeping as new frames arrive.

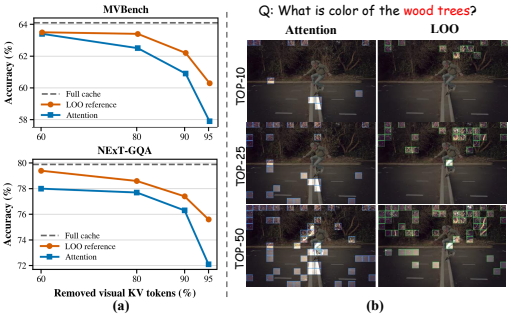


Figure 1: (a) Accuracy on MVBench (Li et al., 2024b) and NExT-GQA (Xiao et al., 2024) under different visual KV token removal ratios. (b) Visual tokens selected by attention and LOO, and LOO retains more question-relevant tokens than attention.

To address this, existing methods rely on proxy signals such as attention mass (Zhang et al., 2026; Ning et al., 2025; Eyzaguirre et al., 2026), value magnitude (Kim et al., 2025), feature similarity (Xie et al., 2026; Wang et al., 2026), and semantic segmentation (Chen et al., 2026) to decide which tokens to evict when the cache overflows. The problem is that none of these signals directly capture what matters: how much the model's output would change if a given token were removed. High attention does not imply high importance, and tokens that look redundant may carry the evi-

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

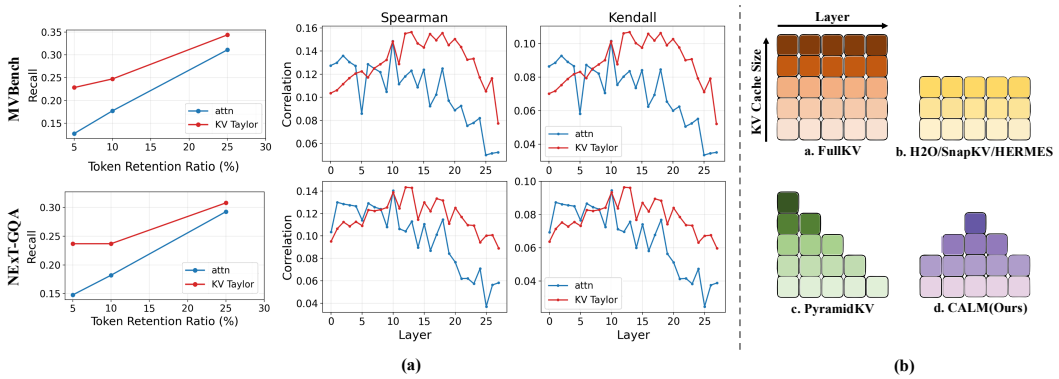


Figure 2: (a) Agreement of attention and KV Taylor with LOO on MVBench and NExT-GQA. The left column reports the recall of the top-$p\%$ LOO-ranked tokens using layer-averaged scores at the same token retention ratio. The middle and right columns report layer-wise Spearman and Kendall correlations. KV Taylor shows stronger agreement with LOO. (b) Comparison of CALM with traditional methods in per-layer KV cache allocation patterns.

dence a correct answer hinges on. Under aggressive compression, this gap widens: tokens that grab attention crowd out the ones that carry the answer.

In this paper, we construct a Leave-one-out (LOO) reference to define visual token importance by removing one token at a time and measuring KL divergence between original and perturbed model output distributions. LOO substantially outperforms attention-based selection on both benchmarks, with the gap widening to its maximum at a removal ratio of 95% (Figure 1a), and retains more tokens from question-relevant regions (Figure 1b). However, computing LOO directly requires a separate forward pass per token and is impractical at scale. We therefore introduce the KV Taylor score to measure the local sensitivity of the probe loss to each token's KV entries via only one forward-backward pass. KV Taylor achieves 1.5$\times$ the top-5% LOO recall compared with attention-based selection (Figure 2a). Notably, its agreement with LOO varies across layers, with particularly strong agreement in the middle layers, motivating a cache policy that adapts to model depth (Figure 2b).

Motivated by these observations, we propose the Contribution-Aware Layer-wise Memory (CALM), a training-free caching framework for streaming video understanding. CALM tailors both retention budgets and token scoring within individual layers. It allocates more cache capacity to middle layers, where KV Taylor aligns best with LOO, and less to shallow and deep layers, where the alignment weakens. For token scoring, CALM emphasizes recency in shallow layers and gradually transitions to KV Taylor scores in deeper layers, allowing the model to clearly capture ongoing events while preserving key information throughout the video stream. To prevent redundant tokens from dominating the cache, we introduce Conditional Novelty Reranking, which down-weights tokens that are similar to those already retained. Finally, a lightweight KV-Resident Text Memory stores compact textual summaries of past video chunks, preserving semantic context.

In summary, our main contributions are as follows:

- We construct the LOO reference to directly measure each token's contribution by its removal effect on output distributions, and introduce the KV Taylor, a first-order sensitivity score for computation efficiency via a single forward-backward pass.
- We propose CALM, a training-free caching framework that assigns layer-wise cache budgets and token scoring policies. CALM further uses conditional novelty reranking to suppress redundant high-scoring tokens, and maintains a KV-Resident Text Memory to preserve compact textual context across video chunks.
- We demonstrate that CALM consistently improves streaming video understanding across three model families on StreamingBench and OVO-Bench, achieving state-of-the-art results with gains of 6.25% on StreamingBench and 11.82% on OVO-Bench (12.77% on the Real-Time and 10.87% on the Backward Tracing).

2 Related Work

Streaming video understanding. Streaming methods differ in how they store past evidence and when they pay the cost of recovering it. VideoLLM-online (Chen et al., 2024a) formulates streaming perception as online dialogue. Flash-VStream (Zhang et al., 2024a) and Dispider (Qian et al., 2025) add dedicated memory modules for long streams—though these typically require additional training and broaden the memory interface with extra components or a separate recall step. ReKV (Di et al., 2025) avoids training by caching KV entries from all frames and retrieving relevant ones at answer time, but pays the cost in memory-intensive storage and added inference latency.

A more practical line of work compresses during encoding so the active cache stays bounded. Attention-based methods retain tokens with high probe attention: LiveVLM (Ning et al., 2025) preserves visual-to-visual attention sinks; HERMES (Zhang et al., 2026) adopts layer-specific policies scored by generic-prompt attention. Non-attention methods explore alternative cues—InfiniPot-V (Kim et al., 2025) uses value-vector norms with layer-adaptive pooling, StreamKV (Chen et al., 2026) segments the stream semantically and allocates layer-wise budgets, FluxMem (Xie et al., 2026) removes redundancy via feature similarity, and StreamingTOM (Chen et al., 2025) reduces pre-LLM tokens through frame-change detection. These methods make streaming practical, yet their retention decisions all rely on proxies for token utility rather than measuring the effect on model output directly.

KV cache compression for video LLMs. For language models, H2O (Zhang et al., 2023) evicts tokens by accumulated attention mass, SnapKV (Li et al., 2024c) selects tokens using recent-window attention, and PyramidKV (Cai et al., 2024) allocates a pyramidal layer-wise budget. For vision-language models, FastV (Chen et al., 2024b) prunes visual tokens with attention after early layers, SparseVLM (Zhang et al., 2024b) uses cross-attention to the query, and VisionZip (Yang et al., 2025) keeps dominant and contextual tokens. DART (Wen et al., 2025), DivPrune (Alvar et al., 2025), and ToMe (Bolya et al., 2022) optimize diversity by preferring complementary tokens over merely salient ones. Most of these methods assume the full context or user query is available before compression, making them incompatible with streaming where frames arrive gradually and questions are unknown. They also typically use attention as an importance signal, despite prior work showing attention weights are unreliable contribution measures (Jain & Wallace, 2019; Abnar & Zuidema, 2020). CALM brings first-order Taylor estimation to dynamic KV entries at inference time, and combines it with a layer-wise retention policy.

3 Method

In this paper, we propose CALM, a training-free framework capable of processing streaming video and answering the questions raised at any moment of the video. Specifically, CALM uses KV Taylor as an efficient sensitivity score for cached visual tokens and adapts the retention policy across layers according to its agreement with the LOO reference. The overall framework of CALM is shown in Figure 3.

3.1 Streaming Video Understanding

A streaming video is an unbounded sequence of frames that arrives incrementally and must be processed online without access to future content. Let $\mathcal{V} = \{C_t\}_{t=1}^{\infty}$ denote a continuous video stream that arrives as a sequence of chunks. At an unknown time t_Q, the system receives a question Q about the current scene or an earlier event and must generate the answer A. Before Q becomes available, a guidance prompt q asks the model to describe the video and provides the signal for cache compression. The model Θ remains frozen throughout streaming and question answering.

Encoding chunk C_t appends visual keys and values to every transformer layer. Let $\mathcal{K}_t = \{(\boldsymbol{K}_{l,t}, \boldsymbol{V}_{l,t})\}_{l=1}^{L}$ denote the cache after chunk t. Since the cache grows with every chunk, we assign each layer l a budget B_l and fix their total to LB, where B is the average budget per layer.

After each chunk, an online policy π updates the cache as:

$$\mathcal{K}_t = \mathrm{Update}_{\pi}\big(\mathcal{K}_{t-1}, C_t, q; \{B_l\}_{l=1}^{L}\big)\,. \tag{1}$$

The update first appends the new entries and then compresses only the layers that exceed their budgets.

At query time, the frozen model answers Q using only the cache $\mathcal{K}_{t_Q}$. Therefore, the goal of the online policy is:

$$\max_{\pi} p_\Theta\big(A \mid Q, \mathcal{K}_{t_Q}\big)\,. \tag{2}$$

Thus, we construct the online policy π using the contribution score and retention mechanisms described below.

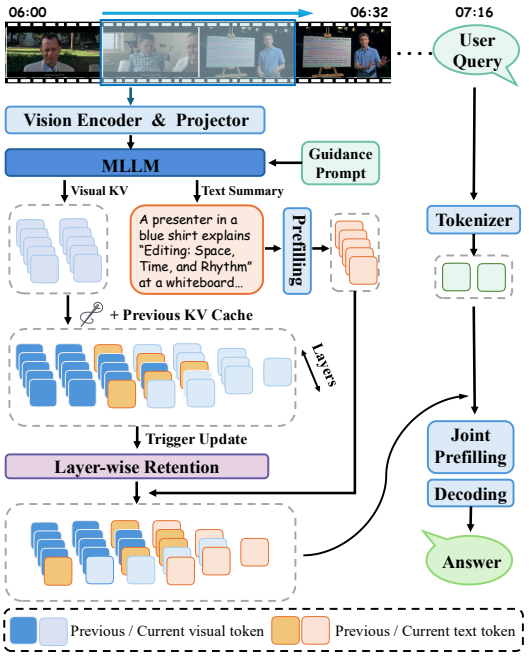


Figure 3: Overview of CALM. Each video chunk updates the visual KV cache and a compact text-memory cache. Layer-wise retention selects visual tokens when a layer exceeds its budget, and the retained visual and text KV entries support future queries.

3.2 Leave-one-out Reference

Most cache compression methods treat attention as token importance, although attention only measures how strongly the current probe reads a token. To obtain a direct reference for token contribution, we define the leave-one-out (LOO) reference as the output change caused by removing one visual position across all layers. The LOO score is defined as:

$$o(i) = D_{\mathrm{KL}}\big(p(\cdot \mid \mathcal{K}, q) \,\big\|\, p(\cdot \mid \mathcal{K}_{\setminus i}, q)\big)\,, \tag{3}$$

where $\mathcal{K}_{\setminus i}$ denotes the cache after the visual position i is masked by adding $-\infty$ to its attention logits in every layer and head. A large $o(i)$ indicates that removing token i causes a substantial change in the output distribution. Computing LOO over the cache requires one forward pass per token, making it impractical for online cache updates; it is used only as an offline reference.

3.3 KV Taylor Score

For efficiently estimating token contribution, we construct a differentiable self-prediction objective that provides a local sensitivity signal distinct from the KL-based LOO reference. Given $\mathcal{K}$ and q, the probe produces a next-token distribution and selects its most likely token as the target in one forward-backward pass as:

$$y^* = \arg\max_y p(y \mid \mathcal{K}, q), \qquad \mathcal{L} = -\log p(y^* \mid \mathcal{K}, q). \tag{4}$$

The target y^* is detached from the computation graph, with frozen model parameters throughout scoring. We compute gradients with respect to a detached copy of the KV cache. For token i at layer l, its first-order terms are defined as:

$$a^K_{l,i} = \left\langle \frac{\partial \mathcal{L}}{\partial \boldsymbol{K}_l[i]}, \boldsymbol{K}_l[i] \right\rangle, \qquad a^V_{l,i} = \left\langle \frac{\partial \mathcal{L}}{\partial \boldsymbol{V}_l[i]}, \boldsymbol{V}_l[i] \right\rangle, \tag{5}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product over all KV heads and feature dimensions. A first-order Taylor expansion approximates the change in the self-prediction loss as $-a^K_{l,i} - a^V_{l,i}$, with full derivation provided in Appendix A.1. Thus, we define the KV Taylor score as the sum of absolute values:

$$s_l(i) = \big|a^K_{l,i}\big| + \big|a^V_{l,i}\big|\,. \tag{6}$$

To that end, a single backward pass can compute the KV Taylor score for all visual tokens across the layers.

Algorithm 1 CALM update after one video chunk

Input: chunk C_t, cache $\mathcal{K}$, prompt q, budgets $\{B_l\}$
Output: updated bounded cache $\mathcal{K}$

```
Encode C_t and append its visual KV entries
Generate S_t and update the protected text memory with Eq. 11
Run one probe forward–backward pass to obtain KV Taylor scores
for each layer l do
    if layer l exceeds B_l then
        Compute the base scores with Eq. 8
        Apply conditional novelty reranking
        Keep text KV entries, then fill the remaining budget with the top-ranked visual entries
    end if
end for
Restore temporal order and update positional encodings
return K
```

3.4 Layer-wise Retention

The KV Taylor score provides stronger agreement with the LOO reference than attention and exhibits a clear, structured variation across model depth. Figure 2a shows that the agreement rises from shallow layers, peaks in middle layers, and remains informative in deep layers. Rather than applying one policy at every depth, CALM uses this measured depth profile to determine both cache capacity and token scoring.

Budgets across layers. Motivated by the depth profile, we adopt a simple center-weighted budget allocation in Figure 2b. We first set the budgets at the middle and edge layers, then interpolate the remaining budgets according to an arithmetic sequence. The hyperparameter γ controls the allocation shape through $B_{\text{edge}} = B/\gamma$ and $B_{\text{mid}} = 2B - B_{\text{edge}}$. Let $d_l \in [0, 1]$ be the normalized distance of layer l from the middle, then the budget of l is:

$$\widetilde{B}_l = B_{\text{mid}} - d_l \left(B_{\text{mid}} - B_{\text{edge}}\right). \tag{7}$$

Scores across layers. We divide the network into shallow, middle, and deep groups using boundaries fixed from the measured depth profile. Inspired by Ebbinghaus' memory decay theory (Ebbinghaus, 2013), we model token recency importance using an exponential forgetting curve based on temporal distance. For visual token i, let $\delta_i \in [0, 1]$ denote its relative temporal distance from the newest token, where $\delta_i = 0$ corresponds to the newest token. Its recency score is defined as $r(i) = \exp(-\kappa\delta_i)$. Then, the layer-wise base score is defined as:

$$b_l(i) = \begin{cases} \widehat{r}(i), & l \text{ is shallow}, \\ (1-\alpha_l)\widehat{s}_l(i) + \alpha_l \widehat{r}(i), & l \text{ is middle}, \\ \widehat{s}_l(i), & l \text{ is deep}, \end{cases} \tag{8}$$

where $\widehat{\cdot}$ denotes min–max normalization within one layer.

Conditional novelty reranking. Ranking tokens independently may retain multiple high-scoring tokens with similar value features. However, enforcing diversity uniformly can displace tokens that are clearly preferred by the base ranking. Conditional novelty reranking therefore promotes complementary evidence when the base selection is redundant, while reducing this adjustment when the base ranking is well separated.

For layer l, we select the top M tokens under b_l as anchors $\mathcal{A}_l$. Let $\boldsymbol{x}_i$ be the value feature of token i and $\boldsymbol{\mu}$ be the mean anchor feature. We use SVD to obtain the leading anchor directions $\boldsymbol{U}$ and define the novelty score as:

$$e_l(i) = \left\| (\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\top)(\boldsymbol{x}_i - \boldsymbol{\mu}) \right\|_2. \tag{9}$$

A large $e_l(i)$ indicates that token i contains information less covered by the anchors. The novelty weight w_l grows when the anchors are more similar, and decreases when the top base scores are well separated from the overall average. We then combine the base and novelty scores as the final score:

$$f_l(i) = (1 - w_l) b_l(i) + w_l e_l(i). \tag{10}$$

Appendix A.2 gives the full definition of the novelty weight.

Table 1: Performance comparison (%) on StreamingBench and OVO-Bench. N denotes the number of sampled frames or the sampling rate (fps), as applicable. OVO-Bench results cover six Real-Time Visual Perception tasks and three Backward Tracing tasks; "Avg." in the final column averages the two task-group averages.

Model	N	Streaming Bench	OVO-Bench											Avg.
			Real-Time Visual Perception							Backward Tracing				
		Real-Time	OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.	
Human	-	91.46	93.96	92.57	94.83	92.70	91.09	94.02	93.20	92.59	93.02	91.37	92.33	92.77
Proprietary MLLMs														
Gemini 1.5 Pro	1 fps	75.69	85.91	66.97	79.31	58.43	63.37	61.96	69.32	58.59	76.35	52.64	62.53	65.93
GPT-4o	64	73.28	69.80	64.22	71.55	51.12	70.30	59.78	64.46	57.91	75.68	48.66	60.75	62.61
Open-source Offline MLLMs														
InternVL2-8B	16	63.72	68.46	58.72	68.97	44.94	67.33	55.98	60.73	43.10	61.49	27.41	44.00	52.37
Qwen2-VL-7B	64	69.04	69.13	53.21	63.79	50.56	66.34	60.87	60.65	44.44	66.89	34.41	48.58	54.62
LongVU-7B	1 fps	-	53.69	53.21	62.93	47.75	68.32	59.78	57.61	40.74	59.46	4.84	35.01	46.31
Open-source Online MLLMs														
Flash-VStream-7B	1 fps	23.23	25.50	32.11	29.31	33.71	29.70	28.80	29.86	36.36	33.78	5.91	25.35	27.61
VideoLLM-online-8B	2 fps	35.99	8.05	23.85	12.07	14.04	45.54	21.20	20.79	22.22	18.80	12.18	17.73	19.26
Dispider-7B	1 fps	67.63	57.72	49.54	62.07	44.94	61.39	51.63	54.55	48.48	55.41	4.30	36.06	45.31
TimeChat-Online-7B	1 fps	75.36	75.20	46.80	70.70	47.80	69.30	61.40	61.90	55.90	59.50	9.70	41.70	51.80
StreamForest-7B	1 fps	77.26	68.46	53.21	71.55	47.75	65.35	60.87	61.20	58.92	64.86	32.26	52.02	56.61
Training-free Offline-to-Online Methods														
LLaVA-OV-7B	64	71.34	67.79	55.05	72.41	48.31	72.28	62.50	63.06	57.24	55.41	18.28	43.64	53.35
+ ReKV	0.5 fps	69.22	52.35	54.13	69.83	43.26	67.33	57.07	57.33	57.58	56.08	18.82	44.16	50.75
+ LiveVLM	0.5 fps	72.92	-	-	-	-	-	-	-	-	-	-	-	-
+ StreamKV	0.5 fps	68.80	-	-	-	-	-	-	-	-	-	-	-	-
+ HERMES	0.5 fps	73.23	72.48	62.39	**74.14**	50.56	73.27	65.22	66.34	60.61	**61.49**	28.49	50.20	58.27
+ CALM	0.5 fps	**74.03**	**74.50**	**66.06**	71.55	**50.56**	**73.27**	**69.02**	**67.49**	**61.95**	59.46	**33.34**	**51.58**	**59.54**
Qwen2.5-VL-7B	1 fps	73.31	67.79	55.05	67.24	42.13	66.34	60.87	59.90	51.52	58.78	23.66	44.65	52.28
+ FluxMem	1 fps	76.40	81.20	59.60	70.70	53.40	**75.20**	63.00	67.20	-	-	-	-	-
+ HERMES	1 fps	79.44	85.23	64.22	71.55	53.37	74.26	65.22	68.98	48.48	62.16	37.63	49.43	59.21
+ CALM	1 fps	**79.56**	**87.92**	**70.64**	**75.86**	**55.06**	74.26	**72.28**	**72.67**	**54.21**	**70.95**	**41.40**	**55.52**	**64.10**
Qwen3-VL-8B	2 fps	78.92	83.89	61.47	75.86	59.55	71.29	59.78	68.64	**61.28**	69.59	10.22	47.03	57.84
+ HERMES	2 fps	81.28	87.92	74.31	**79.31**	56.18	70.30	71.74	73.29	55.56	67.57	24.73	49.28	61.29
+ CALM	2 fps	**83.01**	**91.28**	**77.06**	76.72	**68.54**	**74.26**	**76.09**	**77.32**	60.27	**72.30**	**35.48**	**56.02**	**66.67**
Qwen3-VL-4B	2 fps	78.32	83.89	66.97	76.72	57.30	72.28	66.85	70.67	60.27	**66.22**	23.66	50.05	60.36
+ HERMES	2 fps	78.24	81.88	72.48	**78.45**	59.55	75.25	66.30	72.32	55.89	63.51	45.70	55.03	63.68
+ CALM	2 fps	**81.20**	**83.89**	**79.82**	75.00	**59.55**	**79.21**	**77.72**	**75.87**	**61.95**	65.54	**51.08**	**59.52**	**67.70**

3.5 KV-Resident Text Memory

To strengthen the model's understanding of recent events, we maintain a compact text memory in the KV cache. After each chunk, the model summarizes visible entities, actions, scene changes, and readable text into a factual note S_t, and $\mathcal{T}_t$ is the KV cache of retained text tokens after chunk C_t. Given a text budget B_{text}, we update the memory as:

$$\mathcal{T}_t = [\mathcal{T}_{t-1} \oplus \mathrm{KV}_\Theta(S_t)]_{-B_{\text{text}}:}, \tag{11}$$

where KV_Θ encodes the note at every layer, $\oplus$ denotes concatenation, and $[\cdot]_{-B_{\text{text}}:}$ retains at most the last B_{text} text positions. Moreover, these text tokens are protected from visual token removal and excluded from visual token scoring. As shown in Algorithm 1, CALM jointly updates its visual and text KV memory after each chunk.

4 Experiments

4.1 Experimental Setup

Baselines and Benchmarks. To comprehensively evaluate CALM under diverse temporal dynamics and model capacities, we conduct experiments on three frozen model families: LLaVA-

OneVision (Li et al., 2024a), Qwen2.5-VL (Bai et al., 2025b), and Qwen3-VL (Bai et al., 2025a). For streaming video understanding, we evaluate on StreamingBench (Lin et al., 2026) and OVO-Bench (Niu et al., 2025), which rigorously test long-range event association and instantaneous dynamic response within continuous streams. To verify that CALM generalizes beyond the streaming setting, we additionally evaluate on three offline benchmarks: Video-MME (Fu et al., 2025), MVBench (Li et al., 2024b), and EgoSchema (Mangalam et al., 2023).

Our comparison involves two categories. The first is the full-cache reference, which processes all frames without compression. The second comprises recent training-free streaming compression methods operating under bounded-cache constraints: LiveVLM (Ning et al., 2025), ReKV (Di et al., 2025), StreamKV (Chen et al., 2026), FluxMem (Xie et al., 2026), and HERMES (Zhang et al., 2026).

Implementation Details. For all benchmarks, we process each video in 16-frame chunks with up to 256 visual tokens per frame and sequentially prefill the frozen backbone. We divide the transformer layers into shallow, middle, and deep groups using a fixed 20%–60%–20% split. In middle layers, the recency weight decreases from 0.4 to 0.1, with $\kappa = 20$ controlling temporal decay. We set the layer allocation factor to $\gamma = 2$, while conditional novelty reranking uses up to 256 anchors and a rank-64 basis. KV-Resident Text Memory generates at most 32 tokens per chunk and retains the most recent 128 text tokens. We use "Summarize the video narrative, identifying main characters, key events, timeline changes, and the overall theme." as the guidance prompt. Following HERMES, we use a 4K KV cache budget, including both visual and text KV entries.

4.2 Main Results

Streaming Video Understanding. We conducted extensive experiments on streaming video datasets. As shown in Table 1, CALM consistently improves streaming performance across a wide range of backbones—LLaVA-OV-7B, Qwen2.5-VL-7B, Qwen3-VL-4B, Qwen3-VL-8B, and others. On both StreamingBench and OVO-Bench, it surpasses prior state-of-the-art methods. The gains are pronounced on tasks that require both short-horizon visual cues and temporal context; for example, action recognition and object recognition on OVO-Bench saw improvements of 15.59% and 11.41%, respectively. These results demonstrate that CALM effectively extracts key information from redundant video streams, enabling robust online reasoning across diverse model architectures.

Offline Video Understanding. As shown in Table 2, we further evaluate on the offline short-video benchmark (MVBench) and long-video benchmarks (EgoSchema and VideoMME). Under the same 4K budget, CALM improves the baseline on MVBench and EgoSchema and consistently outperforms HERMES across all three benchmarks. Although our method is specifically designed for streaming video scenarios, it also generalizes well to conventional offline video understanding tasks and uses significantly fewer tokens. CALM filters out repeated or distracting content while preserving complementary fine-grained and semantic evidence across layers, resulting in a compact yet informative context for offline video reasoning.

Table 2: Performance comparison (%) on offline benchmarks. Video-MME is evaluated without subtitles.

Method	Frames	MVBench	EgoSchema	VideoMME
Open-source Offline MLLMs				
Video-LLaMA2-7B	32	49.52	-	-
LongVA-7B	128	59.96	-	-
LLaVA-Video-7B	32	58.60	57.30	63.30
Qwen2-VL-7B	64	67.00	66.70	63.30
Open-Source Online MLLMs				
Dispider-7B	1 fps	-	55.60	57.20
TimeChat-Online-7B	1 fps	75.36	61.90	53.22
StreamForest-7B	1 fps	70.20	-	61.40
Training-free Offline-to-Online Methods				
Qwen2.5-VL-7B	1 fps	65.00	58.47	**64.52**
+ HERMES	1 fps	65.53	59.97	60.63
+ CALM	1 fps	**65.70**	**60.36**	61.33

Table 3: Cumulative component ablation on Qwen2.5-VL-7B. Each row adds the listed component. StrBench denotes StreamingBench; Real. and Bwd. denote real-time perception and backward tracing.

Configuration	StrBench	OVO-Bench		
	Real.	Real.	Bwd.	**Avg.**
(1) Qwen2.5-VL-7B	73.31	59.90	44.65	52.28
(2) + Layer-wise retention	77.11	70.44	49.26	59.85
(3) + Novelty reranking	77.63	71.68	51.48	61.58
(4) + Text memory	**79.56**	**72.67**	**55.52**	**64.10**

Table 4: Score-policy ablation on Qwen2.5-VL-7B. StrBench denotes StreamingBench; Real. and Bwd. denote real-time perception and backward tracing.

Configuration	Score policy	StrBench	OVO-Bench		
		Real.	Real.	Bwd.	**Avg.**
Qwen2.5-VL-7B	–	73.31	59.90	44.65	52.28
+ CALM	Attention	78.28	70.91	53.59	62.25
+ CALM	KV Taylor	**79.56**	**72.67**	**55.52**	**64.10**

Table 5: Layer-budget ablation on Qwen2.5-VL-7B. Results are accuracy in percent; the highlighted row denotes the default setting.

γ	StrBench	OVO-Bench		
	Real.	Real.	Bwd.	**Avg.**
1	78.08	71.87	55.24	63.55
2	**79.56**	**72.67**	**55.52**	**64.10**
4	78.36	71.95	55.03	63.49

Table 6: Efficiency under varying input lengths and chunk sizes. TTFT and TPOT denote time to first token and time per output token.

Metric	Frames			
	16	**64**	**256**	**512**
Chunk size: 8				
GPU mem. / GiB ↓	17.46	18.86	18.86	18.86
TTFT / ms ↓	23.74	23.54	23.97	24.23
TPOT / ms ↓	21.84	22.53	22.72	22.87
Chunk size: 16				
GPU mem. / GiB ↓	18.40	19.78	19.78	19.78
TTFT / ms ↓	22.89	23.68	24.66	24.31
TPOT / ms ↓	22.16	22.77	23.65	23.14

4.3 Ablation Studies

Component contributions. To evaluate the contributions of Layer-wise Retention, Conditional Novelty Reranking, and KV-Resident Text Memory components, we conduct ablations by selectively enabling them. As shown in Table 3, (2) Layer-wise Retention provides the largest improvement, increasing StreamingBench and OVO-Bench overall by 3.80% and 7.57%, confirming the importance of adapting token scores and cache budgets across layers. (3) Adding Conditional Novelty Reranking improves the OVO-Bench Backward Tracing by 2.22%. And (4) adding KV-Resident Text Memory further improves Backward Tracing by 4.04%, demonstrating its value in preserving earlier events.

Effectiveness of KV Taylor. As shown in Table 4, replacing attention with KV Taylor within the complete CALM pipeline improves StreamingBench by 1.28% and the OVO-Bench average by 1.85%. This controlled comparison isolates the incremental benefit of the score in the evaluated framework, whereas the larger base-to-CALM gains measure the complete architecture.

Budgets across layers. As shown in Table 5, a moderate center-weighted allocation ($\gamma = 2$) outperforms both the uniform allocation ($\gamma = 1$) and the more concentrated allocation ($\gamma = 4$). These results support the selected allocation, which favors middle layers while retaining capacity in other layers.

Efficiency analysis. To evaluate the efficiency of CALM, we utilize three metrics: peak GPU memory usage, Time to First Token (TTFT), and Time Per Output Token (TPOT) across varying numbers of input frames. All experiments are conducted using LLaVA-OV-7B as the base model with a 4K-token memory budget. We further examine the efficiency of CALM under varying encoded video chunk sizes, with the results shown in Table 6. GPU memory usage does not increase with longer video lengths due to the fixed memory budget. Moreover, TTFT and TPOT remain consistently low across varying video lengths and chunk sizes, confirming real-time responsiveness in practical streaming scenarios. Although KV Taylor requires a backward pass, its cost remains nearly constant at 365 GFLOPs for 16-frame chunks, and this computation is performed during online cache updates rather than answer generation. Overall, CALM maintains memory and stable latency as frames grow, while keeping the cost of online update practical for streaming. A detailed

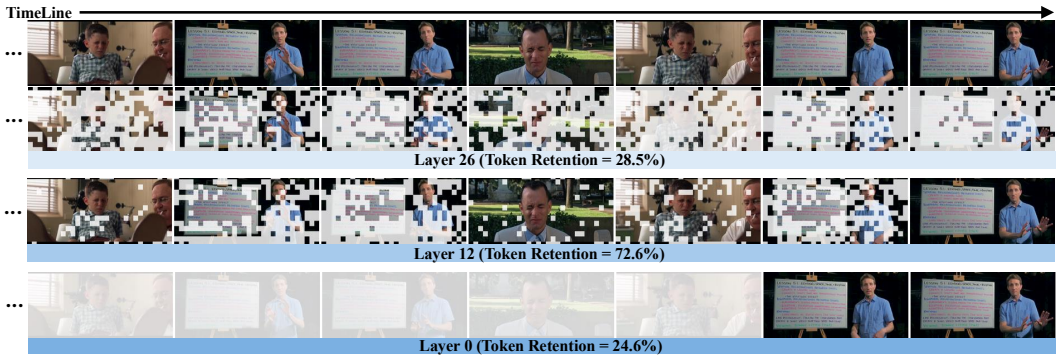


Figure 4: Layer-wise visual token retention example. The distinct patterns at layers 0, 12, and 26 reflect the depth-specific scoring rules and center-weighted cache allocation of CALM.

per-chunk wall-clock breakdown is provided in Appendix C, supporting the practicality of CALM's online updates under the evaluated settings.

4.4 Qualitative Analysis

Figure 4 visualizes how the layer-wise policy changes the retained KV cache. Layer 0 is governed by recency and therefore removes most earlier visual tokens while preserving the newest frames nearly intact. Layer 12 receives the largest cache allocation and combines recency with KV Taylor. In contrast, layer 26 uses KV Taylor and preserves a sparse set of contribution-sensitive regions across the full temporal span despite retaining only 28.5%. These qualitatively different patterns support using layer-wise scores and budgets rather than one uniform retention rule at every layer. Overall, these patterns illustrate how the layer-wise policy preserves complementary visual evidence at different layers.

4.5 Validation of the LOO and Depth Profile

To validate LOO and examine whether the observed depth profile depends on the evaluation data, we conduct four robustness studies. We first compare LOO-based retention with KV Taylor, attention, and uniform sampling across several compression ratios. When 95% of the visual tokens are removed, LOO outperforms attention by 2.4% on MVBench and 3.5% on NExT-GQA. This result supports LOO as a reliable reference under severe compression. Using this reference, we then compare the layer-wise agreement between LOO and KV Taylor across datasets, sample scales, and question types. Across video dataset pairs, the Pearson correlation between depth profiles remains above 0.792, even as the question-type TV distance reaches 0.792. Moreover, profiles estimated from only 10% of the samples retain a Pearson correlation above 0.986 with the full-data profiles. Overall, these results show that the depth profile is stable across diverse data conditions. Detailed settings, numerical results, and further analysis are provided in Appendix B.

5 Conclusion

This paper proposes CALM, a training-free framework for efficient streaming video understanding. Guided by the Leave-one-out (LOO) reference of formalizing token contribution, we introduce the KV Taylor score, an efficient sensitivity score that enables contribution-aware token retention. By introducing layer-wise retention and KV-resident text memory, CALM further enhances understanding performance for long streaming input. Extensive experiments demonstrate that CALM delivers accurate performance under continuously growing video streams, while consistently maintaining extremely low response latency and compact GPU memory usage, supporting practical deployment in real-world streaming scenarios.

AI USE STATEMENT

We disclose our use of generative AI tools along the categories defined by the ICLR 2027 AI policy. **Required disclosure.** Generating synthetic datasets: not used; Helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in writing proofs: not used. Proposing or refining hypotheses: not used. Designing or providing feedback on research methodology or experiments: not used. Implementing methods: not used. Assisting with translation: not used. Cleaning and reformatting datasets: not used. Supporting qualitative and thematic data analysis: not used. Interpreting results: not used. **Recommended disclosure**. Drafting parts of the research paper and editing it to improve readability: used; generative AI tools assisted in drafting and polishing the manuscript text. Formatting references: not used. We have reviewed all AI-assisted work. The authors take responsibility for the final content of this work, including its text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

Section 3 and Algorithm 1 describe the cache-update and token-retention procedures. Section 4 reports the evaluated backbones, benchmarks, chunk sizes, cache budgets, guidance prompt, and principal hyperparameters. Appendix A.1 provides the KV Taylor derivation, and Appendix A.2 details novelty reranking. We commit to publicly releasing the code upon publication.

REFERENCES

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4190–4197, 2020.

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. DivPrune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9392–9401, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025b.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster, 2022.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, et al. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling, 2024.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. VideoLLM-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18407–18418, 2024a.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 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*, pp. 19–35. Springer, 2024b.

Xueyi Chen, Keda Tao, Kele Shao, and Huan Wang. StreamingTOM: Streaming token compression for efficient video understanding, 2025.

Yilong Chen, Xiang Bai, Zhibin Wang, Chengyu Bai, Yuhan Dai, and Ming Lu. StreamKV: Streaming video question-answering with segment-based KV cache retrieval and compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3120–3128, 2026.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Tao Zhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video KV-cache retrieval, 2025.

Hermann Ebbinghaus. Memory: A contribution to experimental psychology. *Annals of neurosciences*, 20(4):155, 2013.

Cristobal Eyzaguirre, Jiajun Wu, and Juan Carlos Niebles. Linear scaling video VLMs for long video understanding, 2026.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24108–24118, 2025.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Drew A Hudson and Christopher D Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card, 2024.

Sarthak Jain and Byron C. Wallace. Attention is not explanation, 2019.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. InfiniPot-V: Memory-constrained KV cache compression for streaming video understanding, 2025.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. LLaVA-OneVision: Easy visual task transfer, 2024a.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. MVBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22195–22206, 2024b.

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

Junming Lin, Zheng Fang, Chi Chen, Haoxuan Cheng, Zihao Wan, Fuwen Luo, Ziyue Wang, Peng Li, Yang Liu, and Maosong Sun. StreamingBench: Assessing the gap for MLLMs to achieve streaming video understanding. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 12147–12151. IEEE, 2026.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3195–3204, 2019.

Zhenyu Ning, Guangda Liu, Qihao Jin, Chengwei Li, Wenchao Ding, Minyi Guo, and Jieru Zhao. Livevlm: Efficient online video understanding via streaming-oriented kv cache and retrieval, 2025.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, et al. OVO-Bench: How far is your video-LLMs from real-world online video understanding? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18902–18913, 2025.

Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24045–24055, 2025.

Chao Wang, Xudong Tan, Jianjian Cao, Kangcong Li, and Tao Chen. Curvestream: Boosting streaming video understanding in mllms via curvature-aware hierarchical visual memory management, 2026.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9972–9991, 2025.

Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. Can i trust your answer? visually grounded video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13204–13214, 2024.

Yiweng Xie, Bo He, Junke Wang, Xiangyu Zheng, Ziyi Ye, and Zuxuan Wu. FluxMem: Adaptive hierarchical memory for streaming video understanding, 2026.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. VisionZip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. ActivityNet-QA: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 9127–9134, 2019.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-VStream: Memory-based real-time understanding for long video streams, 2024a.

Haowei Zhang, Shudong Yang, Jinlan Fu, See-Kiong Ng, and Xipeng Qiu. HERMES: KV cache as hierarchical memory for efficient streaming video understanding, 2026.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. SparseVLM: Visual token sparsification for efficient vision-language model inference, 2024b.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2O: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

A METHOD DERIVATIONS

A.1 TAYLOR APPROXIMATION

To derive the KV Taylor score, consider visual token i at layer l and introduce separate scalar gates for its key and value. The gated entries are defined as

$$\boldsymbol{K}_l^{g_K}[i] = g_K \boldsymbol{K}_l[i], \qquad \boldsymbol{V}_l^{g_V}[i] = g_V \boldsymbol{V}_l[i]. \tag{12}$$

Let $g_K, g_V \in [0,1]$. The original cache corresponds to $g_K = g_V = 1$, while $g_K = g_V = 0$ jointly zeroes the selected key and value. Let $\ell(g_K, g_V)$ denote the objective under these gates. A first-order expansion around the original cache gives

$$\ell(0,0) - \ell(1,1) \approx -\left.\frac{\partial \ell}{\partial g_K}\right|_{1,1} - \left.\frac{\partial \ell}{\partial g_V}\right|_{1,1}. \tag{13}$$

Applying the chain rule to each gate yields

$$\left.\frac{\partial \ell}{\partial g_K}\right|_{1,1} = \left\langle \frac{\partial \mathcal{L}}{\partial \boldsymbol{K}_l[i]}, \boldsymbol{K}_l[i] \right\rangle = a_{l,i}^K, \qquad \left.\frac{\partial \ell}{\partial g_V}\right|_{1,1} = \left\langle \frac{\partial \mathcal{L}}{\partial \boldsymbol{V}_l[i]}, \boldsymbol{V}_l[i] \right\rangle = a_{l,i}^V. \tag{14}$$

Therefore, the change in the self-prediction loss under joint KV gating can be approximated as

$$\Delta \mathcal{L}_{l,i} = \ell(0,0) - \ell(1,1) \approx -a_{l,i}^K - a_{l,i}^V. \tag{15}$$

The two signed terms may cancel even when one path has a large individual effect. We avoid this cancellation by summing their magnitudes, and the triangle inequality gives

$$\left|a_{l,i}^K + a_{l,i}^V\right| \le \left|a_{l,i}^K\right| + \left|a_{l,i}^V\right| = s_l(i). \tag{16}$$

The KV Taylor score therefore upper-bounds the magnitude of the joint first-order gating estimate and serves as an efficient local sensitivity proxy.

A.2 DETAILS OF CONDITIONAL NOVELTY RERANKING

Novelty from the anchor subspace. For layer l, we flatten the value entry of token i and normalize it as

$$\boldsymbol{x}_i = \frac{\mathrm{vec}(\boldsymbol{V}_l[i])}{\|\mathrm{vec}(\boldsymbol{V}_l[i])\|_2}. \tag{17}$$

Let $\mathcal{A}_l$ contain the M tokens with the largest base scores. Their mean feature is

$$\boldsymbol{\mu} = \frac{1}{M} \sum_{i \in \mathcal{A}_l} \boldsymbol{x}_i. \tag{18}$$

We stack the centered anchor features as the rows of $\boldsymbol{X}_l$ and compute

$$\boldsymbol{X}_l = \boldsymbol{P} \boldsymbol{\Sigma} \boldsymbol{U}_{\mathrm{all}}^\top. \tag{19}$$

The matrix $\boldsymbol{U}$ contains the first r columns of $\boldsymbol{U}_{\mathrm{all}}$. These directions capture the main information shared by the anchors.

The columns of $\boldsymbol{U}$ are orthonormal. The closest point to $\boldsymbol{x}_i - \boldsymbol{\mu}$ in their span has coefficients

$$\boldsymbol{c}^* = \boldsymbol{U}^\top (\boldsymbol{x}_i - \boldsymbol{\mu}). \tag{20}$$

Substituting these coefficients into the projection gives

$$\min_{\boldsymbol{c}} \|\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{U}\boldsymbol{c}\|_2 = \|\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{U}\boldsymbol{c}^*\|_2 \tag{21}$$

$$= \left\|(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\top)(\boldsymbol{x}_i - \boldsymbol{\mu})\right\|_2. \tag{22}$$

This distance is exactly the novelty score. A large value therefore identifies a token whose information is not represented by the dominant subspace of high-scoring anchors.

Conditional novelty weight. We first rescale the base and novelty scores within each layer. For $z_l \in \{b_l, e_l\}$, we use

$$\widehat{z}_l(i) = \frac{z_l(i) - \min_j z_l(j)}{\max_j z_l(j) - \min_j z_l(j) + \epsilon}, \tag{23}$$

where ϵ avoids a zero denominator. In Eq. (10) of the main text, b_l and e_l denote these rescaled values.

We measure anchor similarity as

$$R_l = \frac{2}{M(M-1)} \sum_{\substack{i,j \in \mathcal{A}_l \\ i<j}} \left[\boldsymbol{x}_i^\top \boldsymbol{x}_j \right]_+, \tag{24}$$

where $[z]_+ = \max(z, 0)$. A large R_l means that the anchors contain repeated information.

Let k_l be the number of visual tokens retained at layer l. We measure the separation of the base ranking as

$$F_l = \mathrm{Avg}\Big(\mathrm{Top}_{k_l}(\widehat{\boldsymbol{b}}_l)\Big) - \mathrm{Avg}\Big(\widehat{\boldsymbol{b}}_l\Big), \tag{25}$$

where $\widehat{\boldsymbol{b}}_l$ contains the base scores of all candidate tokens. A large F_l means that the top tokens already stand out from the full set.

We combine the two signals through

$$w_l = \eta_l \left[\frac{R_l - \tau_0}{\tau_1 - \tau_0} \right]_0^1 [1 - F_l]_{g_{\min}}^1, \tag{26}$$

where $[z]_a^b = \min(\max(z, a), b)$. The parameter η_l sets the largest novelty weight at layer l. The thresholds τ_0 and τ_1 control when novelty starts and when its redundancy factor reaches its maximum. The lower bound $g_{\min}$ keeps a small novelty contribution when the anchors remain highly similar.

When $R_l \le \tau_0$, we have $w_l = 0$, so the final ranking follows the base score. As R_l grows, novelty receives more weight. A clear base ranking gives a larger F_l and reduces this change.

B Validation of the LOO and Depth Profile

We examine the LOO reference and compare the layer-wise agreement curves across datasets, sample scales, and question types.

B.1 Validity of the LOO Reference

To validate LOO as a meaningful reference for token contribution, we compare token retention based on LOO, KV Taylor, attention, and uniform sampling. Each method retains the same number of visual tokens at a given removal ratio. Tables 7 and 8 show that LOO gives the best accuracy under severe compression. At a 95% removal ratio, LOO exceeds attention retention by 2.4% on MVBench and 3.5% on NExT-GQA. This result supports LOO as a reference for token contribution.

Table 7: MVBench accuracy under KV token removal. ↓ denotes the removal ratio, and full-cache accuracy is 64.1%.

Method	**↓ 60%**	**↓ 80%**	**↓ 90%**	**↓ 95%**
LOO	63.5	**63.4**	**62.2**	**60.3**
KV Taylor	**64.0**	62.8	61.8	59.3
Attention	63.4	62.5	60.9	57.9
Uniform	62.5	62.2	58.1	54.3

Table 8: NExT-GQA accuracy under KV token removal. ↓ denotes the removal ratio, and full-cache accuracy is 79.9%.

Method	**↓ 60%**	**↓ 80%**	**↓ 90%**	**↓ 95%**
LOO	**79.4**	78.6	**77.4**	**75.6**
KV Taylor	79.3	**78.8**	76.8	74.1
Attention	78.0	77.7	76.3	72.1
Uniform	78.6	78.5	74.7	72.1

Table 9: Rank correlation with LOO using layer-averaged token scores across image and video benchmarks. For each sample, token scores are averaged across layers before computing correlations; entries report the mean ± standard deviation across samples.

Benchmark	N	Score	Spearman	Kendall
Image benchmarks				
GQA	3,253	Attention	0.278±0.148	0.191±0.103
		KV Taylor	**0.452±0.132**	**0.319±0.099**
VQAv2	5,000	Attention	0.315±0.155	0.215±0.108
		KV Taylor	**0.474±0.141**	**0.337±0.107**
OK-VQA	5,046	Attention	0.346±0.149	0.237±0.105
		KV Taylor	**0.448±0.142**	**0.318±0.106**
Video benchmarks				
NExT-GQA	1,000	Attention	0.132±0.041	0.089±0.027
		KV Taylor	**0.166±0.039**	**0.113±0.034**
ActivityNet-QA	1,000	Attention	0.148±0.054	0.107±0.040
		KV Taylor	**0.294±0.044**	**0.213±0.043**
MVBench	1,000	Attention	0.147±0.072	0.099±0.051
		KV Taylor	**0.268±0.060**	**0.186±0.049**

B.2 Agreement with LOO across Benchmarks

We extend the comparison between KV Taylor and attention to three image benchmarks (GQA (Hudson & Manning, 2019), VQAv2 (Goyal et al., 2017), and OK-VQA (Marino et al., 2019)) and three video benchmarks (ActivityNet-QA (Yu et al., 2019), MVBench (Li et al., 2024b), and NExT-GQA (Xiao et al., 2024)). For every sample and layer, we compute the Spearman and Kendall correlations between each proxy ranking and the LOO token ranking. We additionally measure top-$p\%$ LOO recall, namely the fraction of the top-$p\%$ tokens under LOO that are recovered when the proxy retains the same proportion. For Table 9, we first average each token's proxy scores across layers, compute rank correlations with LOO for each sample, and then report the mean and standard deviation across samples. Figure 5 reports correlations separately at each layer, shows LOO recall across retention ratios, and compares the key-only, value-only, and joint KV Taylor scores.

KV Taylor outperforms attention in both rank metrics on all six benchmarks. The improvement holds for both image and video inputs and is largest on GQA overall; among the video benchmarks, ActivityNet-QA shows the largest gains, improving Spearman from 0.148 to 0.294 and Kendall from 0.107 to 0.213. Thus, the agreement reported on MVBench and NExT-GQA in the main paper is not specific to those two datasets.

Figure 5 shows that the benefit is consistent across model depth rather than being caused by a few isolated layers. The joint K+V score generally matches or exceeds either single-path variant, with the clearest separation from attention in the middle and deep layers. Its recall advantage is also most visible at retention ratios from 5% to 25%, where selecting the right tokens is hardest. These results support including both key and value effects in KV Taylor and using it under the aggressive cache budgets targeted by CALM.

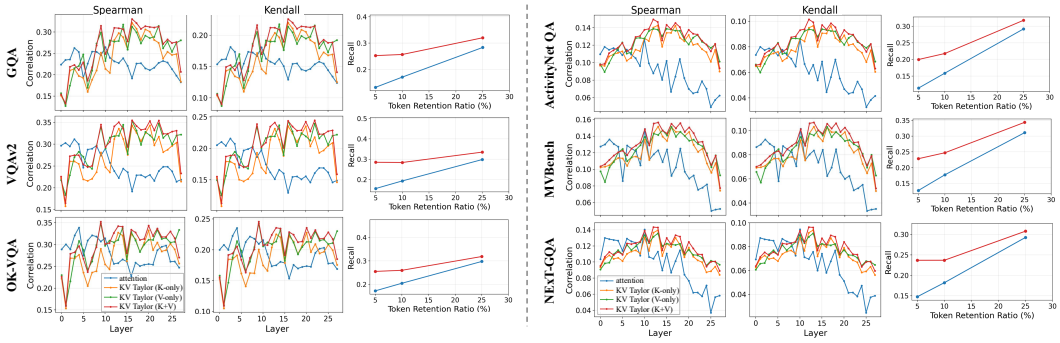


Figure 5: Agreement with the LOO reference across image and video benchmarks. The left block shows image benchmarks, and the right block shows video benchmarks. Within each block, rows correspond to benchmarks, and columns report Spearman correlation, Kendall correlation, and top-$p\%$ LOO recall, respectively. KV Taylor (K+V) uses both key and value Taylor terms, while K-only and V-only isolate their individual contributions.

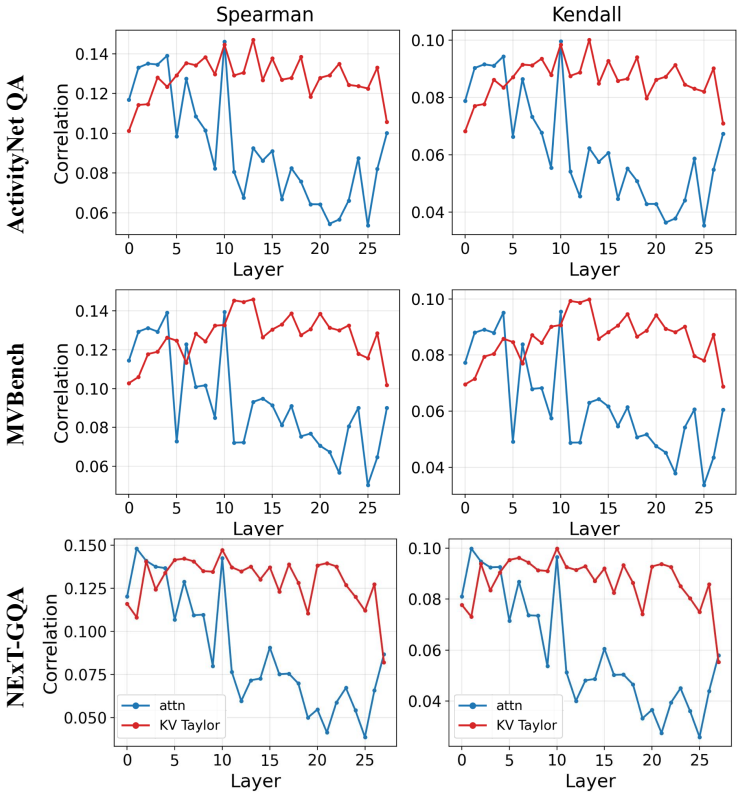


Figure 6: Query-agnostic agreement with LOO using CALM's guidance prompt. Rows correspond to ActivityNet-QA, MVBench, and NExT-GQA; columns report layer-wise Spearman and Kendall correlations.

B.3 Consistency across Datasets

To show that the depth profile is consistent across datasets, we compare it across three image datasets and three video datasets. For each group, we compute the Spearman and Kendall correlations between KV Taylor and LOO at every layer. Then we rescale each curve to the unit interval and align the corresponding layers. Pearson correlation measures shape similarity across two curves, while Spearman and Kendall correlations measure ordering consistency. We also use KL and JS divergence to compare their distributions across depth. Table 10 shows strong agreement across the

Table 10: Pairwise consistency of layer-wise KV Taylor–LOO agreement profiles across datasets. For each dataset pair, P_1 and P_2 denote the normalized layer-wise profiles of the first and second datasets, respectively. Pearson, Spearman, and Kendall measure agreement between profiles, while $D_{\mathrm{KL}}(P_1 \| P_2)$, $D_{\mathrm{KL}}(P_2 \| P_1)$, and JS measure their distributional divergence across layers. Higher correlation and lower divergence indicate greater consistency.

Metric	Dataset Pair	Pearson	Spearman	Kendall	$D_{\mathrm{KL}}(P_1 \| P_2)$	$D_{\mathrm{KL}}(P_2 \| P_1)$	JS
			Image Benchmarks				
Spearman	GQA and OK-VQA	0.9270	0.9048	0.7354	0.0233	0.0265	0.0062
Spearman	GQA and VQAv2	0.9834	0.9721	0.8730	0.0082	0.0091	0.0021
Spearman	OK-VQA and VQAv2	0.9738	0.9475	0.8201	0.0050	0.0048	0.0012
Kendall	GQA and OK-VQA	0.9241	0.8785	0.6984	0.0240	0.0273	0.0063
Kendall	GQA and VQAv2	0.9832	0.9737	0.8783	0.0083	0.0092	0.0022
Kendall	OK-VQA and VQAv2	0.9725	0.9288	0.7884	0.0053	0.0050	0.0013
			Video Benchmarks				
Spearman	ActivityNet QA and MVBench	0.9146	0.9070	0.7566	0.0262	0.0335	0.0072
Spearman	ActivityNet QA and NExT-GQA	0.8088	0.8221	0.6614	0.0636	0.0636	0.0154
Spearman	MVBench and NExT-GQA	0.7926	0.7794	0.6085	0.0541	0.0485	0.0126
Kendall	ActivityNet QA and MVBench	0.9144	0.9075	0.7566	0.0263	0.0337	0.0072
Kendall	ActivityNet QA and NExT-GQA	0.8092	0.8188	0.6561	0.0637	0.0638	0.0154
Kendall	MVBench and NExT-GQA	0.7921	0.7942	0.6138	0.0542	0.0486	0.0126

Table 11: Agreement between subset and full dataset layerwise KV Taylor agreement curves. "Frac." denotes the percentage of the full dataset used, and MAE denotes mean absolute error.

Metric	Frac.	Samples	Pearson	Spearman	Kendall	MAE
			VQAv2			
Spearman	10%	500	0.9931	0.9841	0.9153	0.0048
Spearman	25%	1,250	0.9980	0.9880	0.9365	0.0033
Spearman	50%	2,500	0.9993	0.9934	0.9683	0.0010
Spearman	80%	4,000	0.9998	0.9995	0.9947	0.0007
Kendall	10%	500	0.9929	0.9830	0.9101	0.0036
Kendall	25%	1,250	0.9980	0.9880	0.9365	0.0023
Kendall	50%	2,500	0.9992	0.9940	0.9630	0.0007
Kendall	80%	4,000	0.9998	0.9989	0.9894	0.0005
			NExT-GQA			
Spearman	10%	100	0.9864	0.9797	0.9153	0.0031
Spearman	25%	250	0.9958	0.9918	0.9471	0.0010
Spearman	50%	500	0.9986	0.9929	0.9471	0.0005
Spearman	80%	800	0.9994	0.9989	0.9894	0.0020
Kendall	10%	100	0.9861	0.9797	0.9153	0.0021
Kendall	25%	250	0.9958	0.9907	0.9418	0.0007
Kendall	50%	500	0.9985	0.9918	0.9418	0.0004
Kendall	80%	800	0.9993	0.9984	0.9841	0.0013

datasets, with Pearson correlations above 0.924. ActivityNet QA and MVBench also show high agreement, with Pearson and Spearman correlations above 0.90. Pairs involving NExT-GQA vary more, but Pearson remains above 0.792, and Spearman remains above 0.779 across all video pairs. JS divergence stays below 0.016 for every video pair. These results show that the datasets share a common depth profile, while some dataset-specific variation remains.

B.4 Consistency across Sample Scales

To confirm that the depth profile can be estimated from limited samples, we recompute each curve on nested random subsets. All subsets follow one fixed sample order, so each larger subset contains the smaller ones. We compare every subset curve with the full-data curve. Pearson correlation measures shape agreement, while Spearman and Kendall correlations measure ordering consistency. Mean absolute error measures the pointwise difference between the two curves. Table 11 shows that the estimated curves remain stable with limited samples. At a 10% sample share, Pearson correlation is above 0.986, and Spearman correlation is above 0.979 on both datasets. Kendall correlation remains

Table 12: Question type distribution distances across datasets, and larger values indicate greater differences.

Dataset pair	TV distance	JS divergence
Image datasets		
GQA and OKVQA	0.464	0.148
GQA and VQAv2	0.181	0.081
OKVQA and VQAv2	0.537	0.190
Video datasets		
ActivityNet QA and MVBench	0.485	0.190
ActivityNet QA and NExT-GQA	0.792	0.407
MVBench and NExT-GQA	0.707	0.398

Table 13: Online update efficiency of CALM on LLaVA-OneVision-7B with a 4K-token average cache budget.

Metric	Chunk 8	Chunk 16
Full chunk update / ms	1415.8	1765.9
Video encode / ms	364.9	683.3
Text memory / ms	688.0	724.9
Compression / ms	363.9	357.9
Probe forward / ms	38.4	37.4
Probe backward / ms	31.2	28.8
Other compression / ms	291.1	290.8
Backward / GFLOPs	343	365
Probe F+B / GFLOPs	702	739
Amortized update / ms per frame	177.0	110.4

above 0.91, while mean absolute error stays below 0.005. At a 50% sample share, both Pearson and Spearman correlations exceed 0.99. These results show that a small subset of the sample is sufficient to recover the overall depth profile.

B.5 Variation across Question Types and Prompts

To test robustness to question composition, we group questions into twelve semantic categories and compare their normalized distributions. As shown in Table 12, TV distance reaches 0.537 for image datasets and 0.792 for video datasets, while the corresponding depth profiles remain strongly correlated in Table 10.

The agreement results above use benchmark questions as diagnostic probes. We additionally repeat the experiment using CALM's guidance prompt without benchmark questions. Figure 6 shows improved agreement with LOO under the guidance prompt. This evaluates the scoring signal without benchmark questions; downstream streaming accuracy provides separate evidence about performance on subsequent questions.

C Online Update Cost

Importantly, CALM's online maintenance is performed off the query-critical path and completes well before the next chunk arrives. As shown in Table 13, at 0.5 FPS, a 16-frame chunk arrives every 32 seconds, whereas the complete update requires only 1.766 seconds, consuming 5.52% of the available streaming interval. This supports an effective processing rate of 9.06 FPS, equivalent to 18.1$\times$ the evaluated input rate. Once the cache is saturated, the backward computation remains bounded at 365 GFLOPs per chunk, while query-time TTFT remains approximately 24 ms as the stream grows from 64 to 512 frames. These results demonstrate that CALM provides bounded and lightweight online maintenance without compromising real-time query responsiveness.

D ADDITIONAL QUALITATIVE VISUALIZATIONS

Figures 7–9 provide three additional examples of the layer-wise visual tokens retained by CALM. White overlays indicate removed visual tokens, while unobscured regions indicate retained tokens. Across the examples, shallow layers concentrate on the newest frames, middle layers preserve denser contribution-aware evidence under their larger budgets, and deep layers retain a sparse set of contribution-sensitive regions distributed across the timeline.

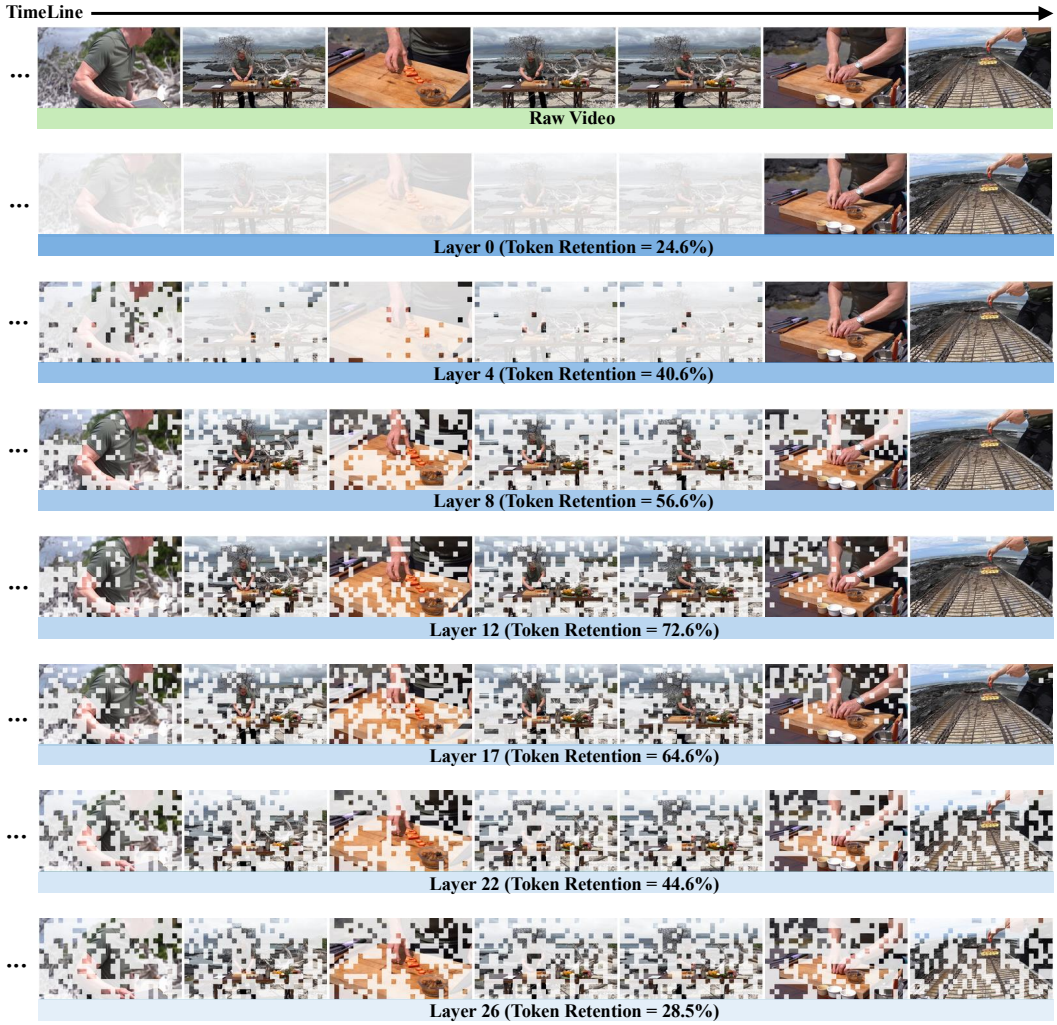


Figure 7: Additional layer-wise token-retention visualization on streaming video.



Figure 8: Additional layer-wise token-retention visualization on streaming video.

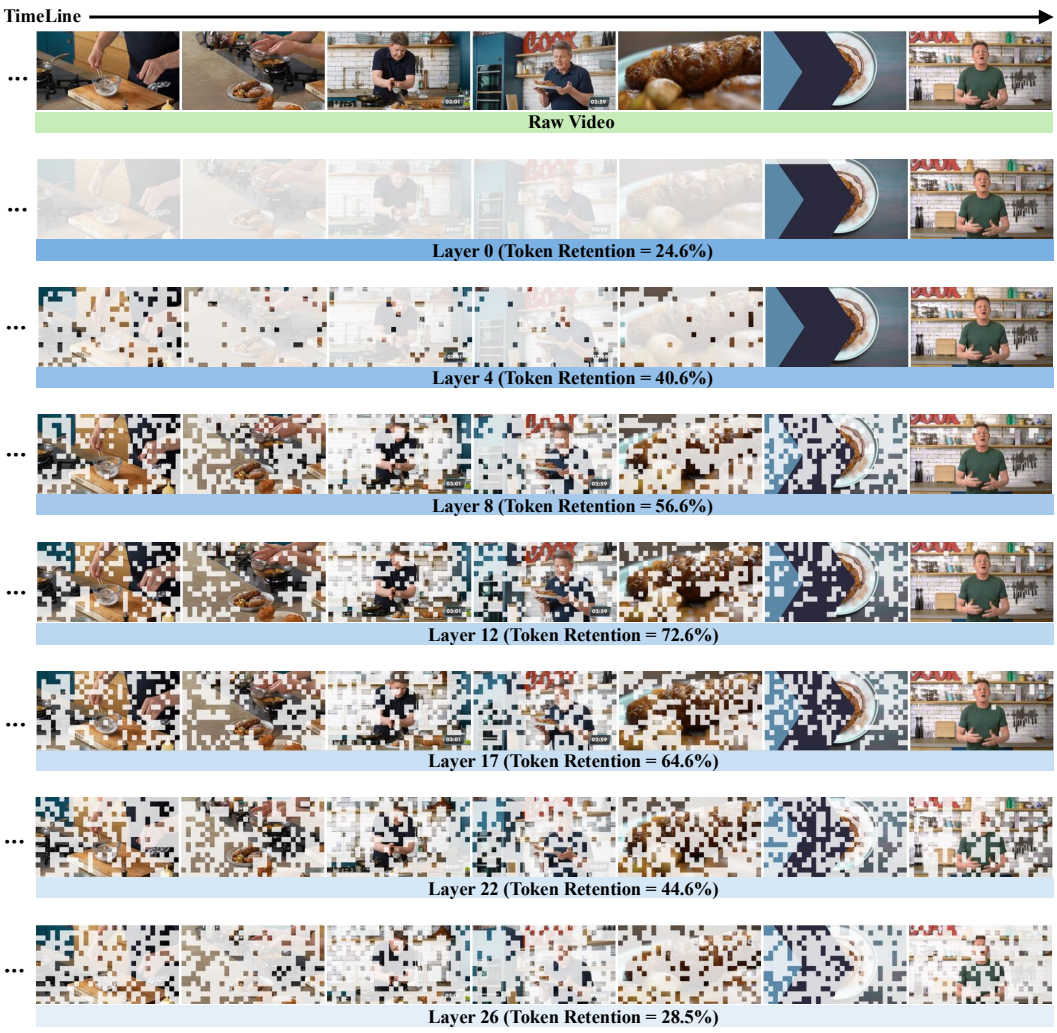


Figure 9: Additional layer-wise token-retention visualization on streaming video.