REDUNDANCY-AWARE VISUAL KV MEMORY FOR STREAMING VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming video understanding requires MLLMs to maintain **informative** visual context under a bounded memory budget as frames arrive continuously. Existing visual KV compression methods largely evaluate KVs as individual units, although feature-similar KVs can encode highly redundant visual content. Such redundancy affects both cross-layer capacity allocation and within-layer KV retention: repeated representations can distort token-level statistics used to estimate layer-wise memory demand and consume multiple retention slots while contributing limited additional information. This work introduces RAKV, a training-free framework for redundancy-aware visual KV memory management. RAKV defines a unified KV score that combines query-agnostic importance with feature representativeness and uses it for both memory decisions. For cross-layer allocation, we aggregate feature-similar KVs into visual groups and measure group-level entropy, yielding a layer-wise demand estimate that is less sensitive to repeated representations. For within-layer retention, the same redundancy-aware score favors KVs that are important and representative of distinct visual patterns. Experiments across streaming video benchmarks and MLLM backbones show that RAKV consistently improves streaming video understanding under a bounded visual KV budget, with gains of up to **9.50**% over the original backbone. Our code will be released upon acceptance.

1 INTRODUCTION

Recent MLLMs have made progress in video understanding (Maaz et al., 2024; Lin et al., 2024; Li et al., 2025; Bai et al., 2025), but most models and benchmarks assume an *offline* setting where the full video is available before inference (Li et al., 2024a; Liu et al., 2024; Fu et al., 2025; Mangalam et al., 2023). In streaming scenarios, frames arrive continuously and queries may occur at any time, requiring the model to answer from only the video observed so far (Chen et al., 2024; Qian et al., 2024; Niu et al., 2025). The model must therefore maintain historical visual memory without access to future user queries. Such memory can be maintained as frames, visual tokens, or model-internal KV states (Xie et al., 2026; Di et al., 2025; Zhang et al., 2026). KV memory is particularly suitable for streaming inference because it reuses previously computed layer-wise representations without re-encoding historical frames. However, retaining all historical visual KVs leads to continually growing memory and computation as the stream unfolds. Streaming inference therefore requires continual compression of the visual KV cache under a fixed memory budget (Yang et al., 2025).

Under a fixed total KV budget, maintaining this bounded memory entails two query-agnostic decisions: ❶ how to allocate capacity across decoder layers (Wan et al., 2025; Tu et al., 2025) and ❷ which visual KVs to retain within each layer (Zhang et al., 2026; Yang et al., 2026). Existing approaches mainly assess visual KVs as individual units, but visual representation redundancy makes such standalone evaluation insufficient. In particular, an additional KV may contribute little new information when other feature-similar KVs already encode the same visual content. Treating such KVs independently can therefore compromise both memory decisions. For ❶ ***cross-layer capacity allocation***, layer-wise demand is often estimated from token-level score distributions, such as attention entropy. When redundant KVs are treated as separate statistical units, token-level entropy reflects not only how score mass is distributed across distinct visual content, but also how it is divided among repeated representations of the same content. The latter can increase entropy even when no additional visual information is introduced, causing the estimated layer-wise demand to become sensitive to representation redundancy. For ❷ ***within-layer KV retention***, importance-based ranking may assign

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.

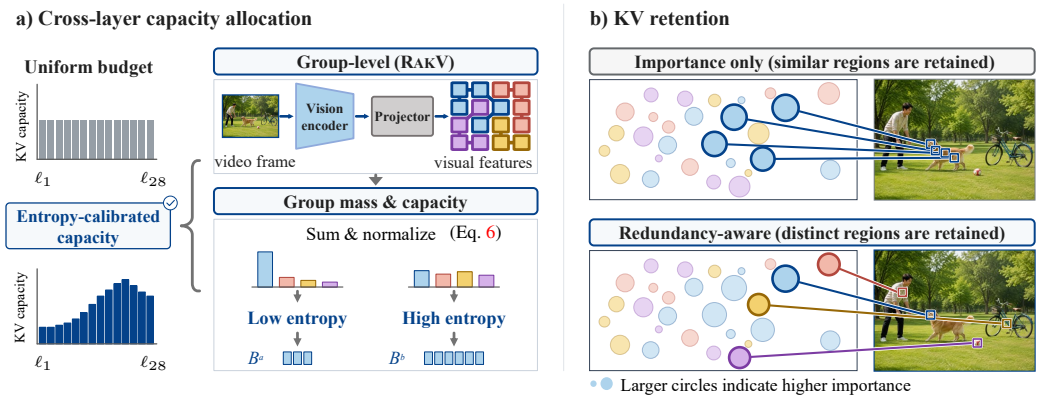

Figure 1: Illustration of redundancy-aware visual KV memory management in RAKV. a) Group-level entropy calibrates layer-wise capacity under a fixed total KV budget. b) Feature representativeness complements base importance to reduce repeated retention of feature-similar visual KVs.

high scores to several feature-similar KVs. Although each may be individually important, retaining them together provides limited additional information while consuming multiple memory slots that could otherwise preserve more informative visual patterns.

These observations motivate a redundancy-aware view of visual KV memory, where memory value depends on both individual importance and the distinct information represented relative to other KVs. Building on this view, RAKV introduces a unified ***redundancy-aware KV score*** that combines query-agnostic importance with feature representativeness and serves as a common basis for both memory decisions. For ❶ ***cross-layer capacity allocation*** (*cf.* Fig. 1a), we perform calibration on unlabeled videos and use group-level entropy to estimate layer-wise capacity demand from how score mass is distributed across visual groups. By aggregating feature-similar representations before measuring dispersion, this statistic is less sensitive to repeated representations of the same visual content. For ❷ ***within-layer KV retention*** (*cf.* Fig. 1b), we apply the same redundancy-aware KV score within each layer to rank candidate KVs, favoring those that are both important and representative of distinct visual patterns. Together, the two components reduce memory spent on repeatedly represented content while preserving distinct visual information under a fixed total KV budget.

We evaluate RAKV on StreamingBench and OVO-Bench using LLaVA-OneVision-7B and Qwen2.5-VL-7B (Lin et al., 2026; Niu et al., 2025; Li et al., 2025; Bai et al., 2025). RAKV consistently outperforms the evaluated training-free baselines under the same total visual KV budget. On OVO-Bench, RAKV improves average accuracy over the original Qwen2.5-VL-7B backbone by **9.50**%. Further experiments on VideoMME and EgoSchema (Fu et al., 2025; Mangalam et al., 2023) demonstrate its effectiveness in offline video understanding. Our code will be released upon acceptance.

Our contributions are summarized as follows:

- We provide a perspective on bounded visual KV memory: under representation redundancy, more visual KV representations do not imply more distinct visual information. This redundancy affects both layer-wise capacity allocation and within-layer KV retention.
- We propose RAKV, a training-free framework with a unified ***redundancy-aware KV score*** that combines query-agnostic importance with feature representativeness, explicitly accounting for whether a KV contributes distinct visual information beyond feature-similar candidates.
- We develop two complementary mechanisms: group-level entropy aggregates score mass over feature-similar KVs to estimate cross-layer capacity demand, while within-layer retention uses the unified score to rank candidate KVs under the calibrated layer-wise capacity.

2 Related Work

Visual KV Memory for Streaming Video. In streaming video understanding, the observed visual history grows continuously as the stream unfolds (Chen et al., 2024; Qian et al., 2024). Retaining

the full history becomes impractical as its memory and computation costs continue to accumulate. Efficient visual KV memory management is therefore essential for practical streaming video understanding under limited memory and computational resources. Existing visual KV cache methods for streaming video can be broadly categorized by whether historical visual states are retrieved after the user query becomes available. *Retrieval-based methods* preserve historical KV states in external storage and retrieve query-relevant content at answering time (Di et al., 2025; Ning et al., 2026; Chen et al., 2026). This enables query-dependent access to long-range visual history, but introduces additional historical storage and query-time retrieval. In contrast, *retrieval-free methods* continuously compress the observed stream into a bounded active KV memory and answer directly from the retained states (Yang et al., 2025; Kim et al., 2025; Zhang et al., 2026). Such methods avoid query-time historical retrieval, but their effectiveness depends on how well the fixed active memory preserves useful information from the observed visual history over long streaming horizons.

RAKV follows the retrieval-free bounded-memory setting and treats visual representation redundancy as a shared factor in cross-layer capacity allocation and within-layer KV retention. It uses a unified redundancy-aware KV score combining query-agnostic importance with feature representativeness to guide both decisions toward preserving distinct visual information under the fixed budget.

Visual KV Retention in Bounded Memory. General LLM KV cache compression methods exploit model-internal attention patterns to guide KV retention (Xiao et al., 2024; Zhang et al., 2023; Li et al., 2024b). In multimodal models, inter-modal relevance provides an additional signal for visual KV compression (Huang et al., 2025). For retrieval-free streaming video, the visual KV memory must be continually updated under a fixed budget before future user queries are known, requiring query-agnostic retention criteria. Existing methods draw on model-internal and temporal cues—including attention scores, value norms, temporal recency, and layer-specific attention patterns—to prioritize visual KVs for retention (Yang et al., 2025; Kim et al., 2025; Zhang et al., 2026).

These studies establish effective strategies for continuously updating a fixed-capacity visual KV memory from the observed video stream. RAKV takes a further step by incorporating feature representativeness into KV evaluation, complementing query-agnostic importance with information about the relation among candidate representations. This allows within-layer retention to reduce the impact of representation redundancy when selecting visual KVs under a limited per layer capacity.

Layer-Wise KV Cache Budget Allocation. Under a fixed KV budget, prior work has explored non-uniform allocation across decoder layers to account for layer-wise differences in KV memory demand. Such demand has typically been estimated from layer importance and attention statistics (Cai et al., 2025; Wang et al., 2025b), as well as information-theoretic criteria such as information loss and token-level entropy (Shen et al., 2025; Wan et al., 2025), to guide the allocation of layer capacity.

While these approaches capture layer heterogeneity, token-level statistics treat repeated visual KVs as separate statistical units, making layer-wise demand estimation sensitive to representation redundancy in streaming video. RAKV instead performs one-time capacity calibration with group-level entropy, aggregating score mass over feature-similar KVs before estimating layer-wise demand.

3 METHOD

Task Setup and Notation. We consider a streaming video setting in which frames arrive sequentially and are processed in streaming chunks (Zhang et al., 2026; Kim et al., 2025; Di et al., 2025). In this setting, RAKV maintains a bounded visual KV memory for a frozen MLLM without access to the future user query. Consider an MLLM with L decoder layers, where layer ℓ has visual KV capacity B^ℓ under a fixed total cross-layer budget B_{tot}, satisfying $\sum_{\ell=1}^{L} B^\ell = B_{\text{tot}}$. At streaming step t, the previously retained memory $\mathcal{M}_{t-1}^\ell$ and the visual KVs $\mathcal{C}_t^\ell$ from the incoming video chunk form the candidate set, from which the updated memory is retained under the layer capacity:

$$\mathcal{M}_t^\ell \subseteq \mathcal{M}_{t-1}^\ell \cup \mathcal{C}_t^\ell, \qquad |\mathcal{M}_t^\ell| \le B^\ell. \tag{1}$$

When a user query arrives after step t, the MLLM conditions on the query and directly uses $\{\mathcal{M}_t^\ell\}_{\ell=1}^{L}$ as its visual-history KV context to generate the answer. This setting gives rise to two query-agnostic memory-management decisions. *Cross-layer capacity allocation* determines the layer capacity

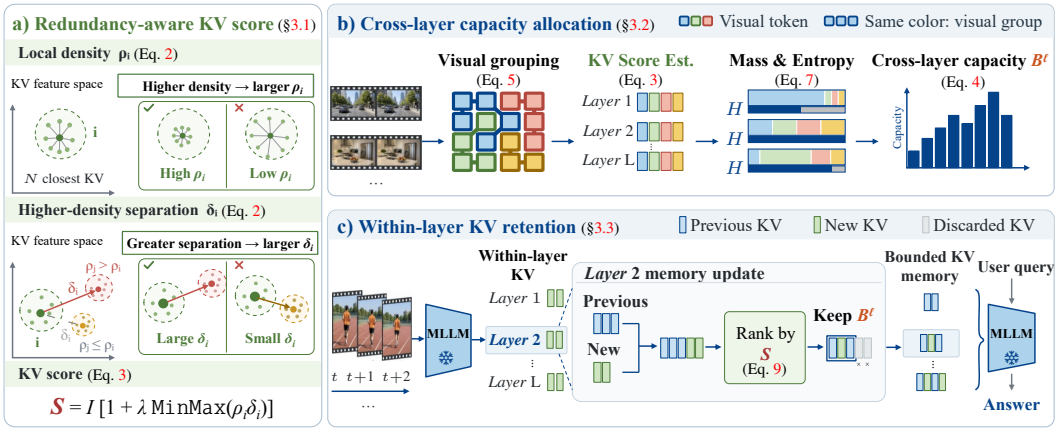

Figure 2: Overview of RAKV. a) The redundancy-aware KV score combines query-agnostic importance with density-peak representativeness. It supports b) cross-layer capacity allocation based on group-level entropy and c) within-layer retention under the calibrated capacity.

$\{B^\ell\}_{\ell=1}^{L}$, while *within-layer KV retention* determines which candidates in $\mathcal{M}_{t-1}^{\ell} \cup \mathcal{C}_t^{\ell}$ are retained under the corresponding capacity B^ℓ. Both decisions are made before the future user query is known.

Framework Overview. RAKV first defines a *redundancy-aware KV score* that combines query-agnostic importance with representativeness (§3.1). This unified score serves as the basis for the two memory management decisions. For cross-layer capacity allocation, we perform a one-time calibration using an unlabeled video set, where KV scores are aggregated over visual groups and summarized by group-level entropy to determine $\{B^\ell\}_{\ell=1}^{L}$ (§3.2). For within-layer KV retention, the same score formulation is used to rank visual KVs in $\mathcal{M}_{t-1}^{\ell} \cup \mathcal{C}_t^{\ell}$ under the calibrated capacity budget assigned to each decoder layer as new video chunks arrive during streaming inference (§3.3).

3.1 REDUNDANCY-AWARE KV SCORE

Under visual representation redundancy, query-agnostic importance may assign similarly high scores to feature-similar visual KVs that encode the same visual content. To account for such redundancy, we first estimate representativeness for each visual KV candidate from feature-space density and separation, capturing its relation to other candidate representations in feature space, and then use this signal to modulate the base importance, yielding the redundancy-aware KV score S_i^ℓ (Fig. 2a).

Density-Peak Representativeness. We quantify representativeness using a density-peak score based on feature-space density and separation. Let $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ denote the unit-normalized value representations of the i-th and j-th visual KV candidates, respectively. Their cosine distance is $d_{ij} = 1 - \langle \boldsymbol{z}_i, \boldsymbol{z}_j \rangle$, where $\langle \boldsymbol{z}_i, \boldsymbol{z}_j \rangle$ denotes their cosine similarity. Let $\mathcal{N}(i)$ denote the set of N candidates with the smallest feature distances to the i-th visual KV candidate in feature space. Following the density-peak principle (Rodriguez & Laio, 2014; Du et al., 2016), for each visual KV candidate, we compute a density score ρ_i from these neighboring distances and a separation score δ_i measuring the feature space distance to the nearest candidate with higher density:

$$\rho_i = \exp\Big(-\frac{1}{N}\sum\nolimits_{j\in\mathcal{N}(i)} d_{ij}^2\Big), \quad \delta_i = \begin{cases} \min_{j:\rho_j>\rho_i} d_{ij}, & \exists j : \rho_j > \rho_i, \\ \max_j d_{ij}, & \text{otherwise.} \end{cases} \tag{2}$$

We combine the two terms and apply min–max normalization to obtain the representativeness score $R_i^\ell = \texttt{MinMax}(\rho_i\delta_i)$. A larger ρ_i indicates higher feature-space density around the i-th visual KV candidate, while a larger δ_i indicates greater separation from higher-density candidates. Consequently, a high R_i^ℓ identifies the candidate as representative of a dense and well-separated feature mode.

Redundancy-Aware KV Score. Let I_i^ℓ denote the base query-agnostic importance assigned to the i-th visual KV candidate at decoder layer ℓ. RAKV defines the redundancy-aware KV score by multi-

plicatively modulating this base importance with the candidate's feature-space representativeness:

$$S_i^\ell = I_i^\ell \left(1 + \lambda^\ell R_i^\ell\right), \tag{3}$$

where $\lambda^\ell \geq 0$ controls the contribution of representativeness. This residual multiplicative form preserves importance as the base score while using representativeness to modulate its contribution. The depth-specific definitions of I_i^ℓ and λ^ℓ are provided in Appendix A.1.

3.2 Cross-layer Capacity Allocation via Group-Level Entropy

Under the fixed total budget B_{tot}, layer-wise capacity calibration distributes the available visual KV memory across the L decoder layers according to their relative capacity demand. Let H^ℓ denote the calibrated entropy statistic of layer ℓ, where a larger value indicates a greater capacity demand. We map these layer-wise statistics to memory capacity using a temperature-controlled softmax:

$$B^\ell = B_{\text{tot}} \frac{\exp\big(H^\ell/\tau\big)}{\sum_{j=1}^{L} \exp(H^j/\tau)}. \tag{4}$$

Here, τ controls the degree of variation in capacity allocation across decoder layers. We use largest-remainder rounding to obtain integer capacity while preserving $B_{\text{tot}} = \sum_{\ell=1}^{L} B^\ell$. The key question is how to estimate H^ℓ under visual representation redundancy. A direct token-level entropy treats repeated visual KVs as separate statistical units, making the estimate of H^ℓ sensitive to redundancy and consequently affecting layer-wise capacity allocation. We therefore use group-level entropy, first aggregating the redundancy-aware KV scores S_i^ℓ from Eq. 3 within visual groups and then computing entropy over the resulting group masses to estimate layer-wise capacity demand (Fig. 2b).

Visual Group Construction. To provide a shared visual group across decoder layers, we construct visual groups from the projected visual-token features before the layer-specific decoder transformations (Xie et al., 2026; Jin et al., 2024). Let $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ denote the unit-normalized projected feature of the i-th and j-th visual tokens. We consider 8-neighborhood adjacent token pairs on the two-dimensional visual-token grid and apply Otsu's method (Otsu, 1979) to their cosine distances to obtain a threshold θ_f. Restricting (i, j) to these adjacent pairs, visual groups are constructed as follows:

$$\mathcal{G} = \texttt{UnionFind}\left(\{(i,j) \mid 1 - \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \leq \theta_f\}\right). \tag{5}$$

Here, `UnionFind` initializes each visual token as a singleton component and merges the selected token pairs, yielding $\mathcal{G} = \{G_1, G_2, \ldots\}$. Each $G \in \mathcal{G}$ contains the indices shared by the grouped visual tokens and their corresponding layer-specific visual KV candidates. The resulting groups can therefore be directly used to aggregate KV scores S_i^ℓ across decoder layers.

Group-Level Entropy Estimation. Given the visual groups constructed above, we aggregate the redundancy-aware KV scores of the visual KV candidates within each group $G \in \mathcal{G}$ and normalize the resulting score masses across all visual groups to obtain the group distribution:

$$P^\ell(G) = \frac{\sum_{i \in G} S_i^\ell}{\sum_{G' \in \mathcal{G}} \sum_{j \in G'} S_j^\ell}. \tag{6}$$

We then compute the group-level entropy as follows:

$$H_{\text{grp}}^\ell = -\sum\nolimits_{G \in \mathcal{G}} P^\ell(G) \log P^\ell(G). \tag{7}$$

We compute this statistic over the calibration set and average the resulting group-level entropy to obtain the layer-wise calibration statistic H^ℓ, which is then used in Eq. 4 to determine B^ℓ.

Proposition 1. *For a fixed decoder layer ℓ, let H_{tok}^ℓ denote the entropy obtained by normalizing the KV scores S_i^ℓ over individual visual KV candidates before group aggregation. Then*

$$H_{\text{tok}}^\ell = H_{\text{grp}}^\ell + H_{\text{within}}^\ell, \qquad 0 \leq H_{\text{within}}^\ell \leq \sum\nolimits_{G \in \mathcal{G}} P^\ell(G) \log |G|, \tag{8}$$

where H_{within}^ℓ is the group-mass-weighted within-group entropy. The upper bound is attained when the score mass is uniformly distributed within each visual group.

Proposition 1 separates the two sources of dispersion in token-level entropy. Specifically, H^{ℓ}_{grp} measures how score mass is distributed across visual groups, whereas $H^{\ell}_{\mathrm{within}}$ captures how the mass within each group is divided among its KV representations. For a fixed group mass $P^{\ell}(G)$, the within-group contribution increases as the mass is distributed more evenly across multiple representations, reaching $P^{\ell}(G)\log|G|$ under a uniform distribution. Thus, repeated representations of the same visual content can increase entropy purely through within-group redistribution, even when they introduce no additional distinct visual information. By aggregating KV scores within each visual group, H^{ℓ}_{grp} excludes this representation-dependent contribution and measures only the distribution of score mass across visual groups, making layer-wise capacity estimation less sensitive to repeated representations. A detailed derivation is provided in Appendix B.

3.3 Redundancy-Aware Within-Layer KV Retention

Once the layer capacity is calibrated, RAKV performs online KV retention under the fixed capacity B^{ℓ} of each decoder layer. At streaming step t, the retained visual KVs $\mathcal{M}^{\ell}_{t-1}$ and the new visual KVs $\mathcal{C}^{\ell}_{t}$ are considered together for memory update (Fig. 2c). When their size exceeds B^{ℓ}, RAKV ranks the candidates using the depth-specific scoring procedure built upon the redundancy-aware KV score defined in §3.1. The updated memory retains the highest-scoring candidates:

$$\mathcal{M}^{\ell}_{t} = \mathrm{TopK}\left(S^{\ell}_{i}, B^{\ell}\right), \qquad i \in \mathcal{M}^{\ell}_{t-1} \cup \mathcal{C}^{\ell}_{t}. \tag{9}$$

Since S^{ℓ}_{i} jointly accounts for query-agnostic importance and representativeness, the selection favors visual KVs that are both important and representative of distinct feature modes. Feature-similar candidates are therefore less likely to repeatedly occupy the limited memory, while high-value historical information can remain retained as the stream evolves. Applying this update throughout streaming keeps the active visual KV memory bounded while continuously adapting its contents to newly observed visual information over time. Together with the layer capacity obtained from offline calibration, within-layer ranking completes the bounded memory management process.

4 Experiment

4.1 Experimental Setup

Datasets. We evaluate RAKV on two benchmarks for online streaming video understanding.

• **StreamingBench** (Lin et al., 2026). StreamingBench comprises 900 YouTube videos spanning eight categories of real-world scenarios. In line with the evaluation protocol adopted in prior work (Xie et al., 2026; Zhang et al., 2026; Liang et al., 2026), the Real-Time Visual Understanding subset is used for evaluation. It contains 2,500 questions over 500 videos and covers 10 tasks.

• **OVO-Bench** (Niu et al., 2025). OVO-Bench contains 644 videos and approximately 2,800 fine-grained questions covering a range of online video understanding tasks. Results are reported on the Real-Time Visual Perception and Backward Tracing subsets (Xie et al., 2026; Zhang et al., 2026; **?**). The former assesses understanding of currently observed video content, whereas the latter evaluates reasoning over previously observed visual content from earlier in the video stream.

Backbones. Following prior evaluation settings, we evaluate RAKV on two representative Video-LLM backbones, LLaVA-OneVision-7B (Li et al., 2025) and Qwen2.5-VL-7B (Bai et al., 2025), to assess whether the method generalizes across the two evaluated backbone architectures.

Evaluation Metrics. All questions in the evaluated subsets are multiple-choice; accordingly, we use accuracy as the evaluation metric and report it consistently across both benchmarks.

Implementation Details. Each video is encoded into 16-frame chunks, which are sequentially processed by the backbone LLM in temporal order. After each chunk, visual KV compression is triggered when the accumulated cache exceeds the corresponding layer capacity. LLaVA-OneVision-7B and Qwen2.5-VL-7B use sampling rates of 0.5 fps and 1.0 fps, respectively, with an average budget of 6K visual KV pairs per layer. For each backbone, the layer-wise capacity allocation is calibrated separately using a length-stratified set of 64 unlabeled videos sampled from the FineVideo training split (Farré et al., 2024), without using annotations or samples from the downstream evaluation

benchmarks. The resulting layer-wise capacity is then fixed throughout evaluation on StreamingBench and OVO-Bench, and all experiments are conducted on five NVIDIA RTX A6000 GPUs.

4.2 Quantitative Comparison Results

Table 1: Streaming video accuracy (%) on StreamingBench and OVO-Bench (§4.2). For each backbone, RAKV calibrates its layer-wise capacity using a 64-video subset of FineVideo. Results are reported as mean ± standard deviation over three runs.

Model	Sampling	StreamingBench	OVO-Bench		
		Real-Time ↑	Real-Time ↑	Backward ↑	Avg. ↑
Human	–	91.46	93.20	92.33	92.77
Proprietary MLLMs					
Gemini 1.5 Pro	1.0 fps	75.69	69.32	62.54	65.93
GPT-4o	64 frames	73.28	64.46	60.75	62.61
Claude 3.5 Sonnet	20 frames	72.44	–	–	–
Open-source Online MLLMs					
Flash-VStream-7B	1.0 fps	23.23	29.86	25.35	27.61
VideoLLM-online-8B	2.0 fps	35.99	20.79	17.73	19.26
Dispider-7B	1.0 fps	67.63	54.55	36.06	45.31
TimeChat-Online-7B	1.0 fps	75.36	61.90	41.70	51.80
StreamForest-7B	1.0 fps	77.26	61.20	52.02	56.61
Training-free Methods					
LLaVA-OneVision-7B	64 frames	71.34	63.06	43.64	53.35
+ ReKV (Di et al., 2025)[ICLR 2025]	0.5 fps	69.22	57.33	44.16	50.75
+ InfiniPot-V (Kim et al., 2025)[NeurIPS 2025]	0.5 fps	72.99	65.13	47.23	56.18
+ LiveVLM (Ning et al., 2026)[DAC 2026]	0.5 fps	72.92	–	–	–
+ StreamKV (Chen et al., 2026)[AAAI 2026]	0.5 fps	68.80	–	–	–
+ HERMES (Zhang et al., 2026)[ACL 2026]	0.5 fps	72.63	65.07	48.80	56.94
+ **RAKV (Ours)**	0.5 fps	**73.28**±0.04	**67.35**±0.11	**50.57**±0.09	**58.96**±0.10
Qwen2.5-VL-7B	1.0 fps	73.31	59.90	44.65	52.28
+ InfiniPot-V (Kim et al., 2025) [NeurIPS 2025]	1.0 fps	76.40	65.90	47.60	56.75
+ HERMES (Zhang et al., 2026)[ACL 2026]	1.0 fps	78.72	68.42	48.10	58.26
+ **RAKV (Ours)**	1.0 fps	**79.04**±0.07	**71.81**±0.29	**51.76**±0.29	**61.78**±0.04

To account for calibration-set sampling variability, we sample three duration-stratified FineVideo (Farré et al., 2024) calibration sets using different random seeds, repeat the calibration procedure separately, and report the mean and standard deviation across the resulting three evaluations.

StreamingBench (Lin et al., 2026). From Table 1, RAKV achieves the best performance among the evaluated training-free methods. On LLaVA-OneVision-7B, RAKV obtains 73.28%, outperforming ReKV (Di et al., 2025) and StreamKV (Chen et al., 2026) by **4.06**% and **4.48**%, respectively. It also outperforms HERMES (Zhang et al., 2026) by **0.65**%. On Qwen2.5-VL-7B, RAKV reaches 79.04%, surpassing HERMES by **0.32**%. These consistent gains across both backbones indicate that RAKV more effectively preserves the visual evidence under the same memory budget.

OVO-Bench (Niu et al., 2025). The results on OVO-Bench further validate the effectiveness of RAKV in supporting both current visual perception and reasoning over previously observed content. RAKV achieves 58.96% with LLaVA-OneVision-7B and 61.78% with Qwen2.5-VL-7B, surpassing HERMES by **2.02**% and **3.52**%, respectively. Concretely, on Qwen2.5-VL-7B, RAKV improves Real-Time Visual Perception from 68.42% to 71.81% and Backward Tracing from 48.10% to 51.76%, corresponding to gains of **3.39**% and **3.66**%, respectively. The comparable gains across both subsets indicate that the improvement extends to both current perception and reasoning over earlier events.

4.3 Qualitative Comparison Results

Fig. 3 presents qualitative comparisons of RAKV, HERMES, and ReKV on two representative examples. The first is drawn from the Action Sequence Inference task under Backward Tracing and

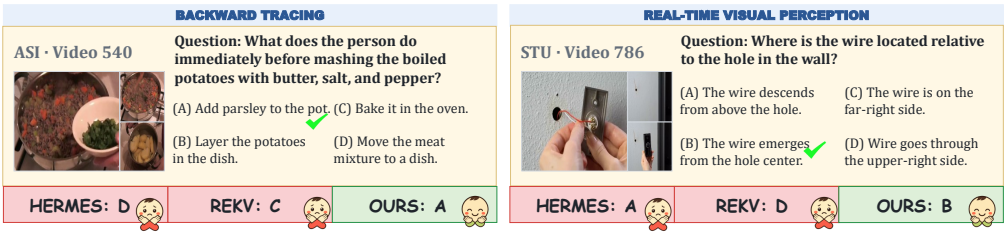

Figure 3: Qualitative comparison (§4.3) of backward tracing and real-time visual perception.

Table 2: Diagnostic experiments on OVO-Bench (§4.4). OVO, FV, and SB denote OVO-Bench, FineVideo, and StreamingBench. RT and Back. denote Real-Time Visual Perception and Backward Tracing, respectively. The adopted configurations are marked in red.

Configuration	Accuracy
LLaVA-OV-7B	53.35
+ Retention	57.87
+ Retention + Budget	**58.96**

(a) core components (§3.2, §3.3)

Formulation	RT	Back.	Avg.
$I \times R$	67.29	47.82	57.56
$I + R$	66.69	49.81	58.25
$I + \lambda R$	66.40	50.39	58.40
$I(1 + \lambda R)$	**67.35**	**50.57**	**58.96**

(b) score combination (Eq. 3)

Unit	RT	Back.	Avg.
Token	65.59	48.08	56.84
Group	67.35	50.57	58.96
Δ (G−T)	**+1.76**	**+2.49**	**+2.12**

(c) calibration granularity (§3.2)

Calib.	RT	Back.	Avg.
OVO	68.62	49.34	58.98
FV	67.35	50.57	58.96
SB	68.16	49.03	58.60

(d) calibration dataset

Size	RT	Back.	Avg.
D64	67.35	50.57	58.96
D96	67.22	50.40	58.81
D128	67.40	50.11	58.76
D256	67.22	50.24	58.73

(e) calibration-set size

Method	Budget	TTFT (s)	Peak (GB)
HERMES	6K	0.07	19.02
RAKV	6K	0.08	19.05

(f) system efficiency

requires reasoning over a previously observed sequence of actions. The second is drawn from the Spatial Understanding task under Real-Time Visual Perception and requires understanding spatial relations in the current scene. In contrast to HERMES and ReKV (Di et al., 2025), RAKV answers both questions correctly, suggesting that it more effectively retains the visual evidence needed for reasoning over earlier action sequences and understanding spatial relations in the current scene.

4.4 Diagnostic Experiments

Core Components. To assess the contributions of the two core components of RAKV, we introduce them incrementally. +Retention (§3.3) applies redundancy-aware retention with uniform layer-wise capacity, while adding +Budget (§3.2) further incorporates group-level capacity calibration, forming the full method. Following prior streaming evaluations (Di et al., 2025; Zhang et al., 2026), the original LLaVA-OV-7B baseline is reported under its standard fixed-frame protocol. As shown in Table 2a, our redundancy-aware retention achieves **57.87**% on OVO-Bench, compared with 53.35% for the original backbone. It also exceeds the HERMES result by 0.93%, supporting the effectiveness of the proposed retention strategy. Adding group-level capacity calibration further improves the average to **58.96**%, yielding an additional 1.09% gain by adapting the fixed total KV budget across decoder layers. Together, these results support the effectiveness of both redundancy-aware retention and group-level capacity calibration, the two core components of RAKV.

Token-Level *vs* Group-Level Calibration. To study the effect of calibration granularity, we compare capacity allocation based on token-level (Wan et al., 2025)and group-level entropy. As shown in Table 2c, group-level calibration improves Real-Time Visual Perception by **1.76**%, Backward Tracing by **2.49**%, and their average by **2.12**%. The consistent gains support the intended role of group-level entropy: reducing repeated representations yields more reliable layer-wise capacity estimates.

Score Formulation. To examine the residual multiplicative formulation in Eq. 3, we compare four combinations of importance I and representativeness R. As shown in Table 2b, the proposed $I(1 + \lambda R)$ achieves the highest average accuracy of **58.96**%, outperforming the strongest alternative

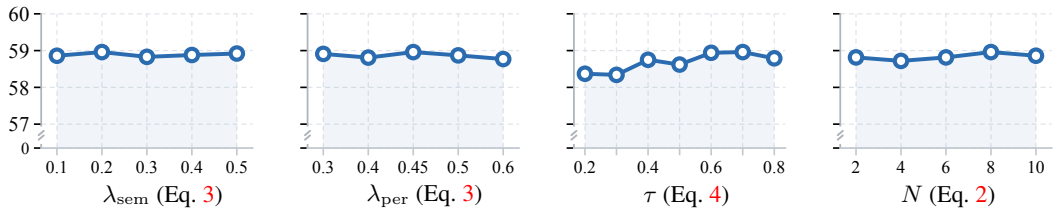

Figure 4: Hyperparameter sensitivity of RAKV on OVO-Bench (§4.4). From left to right, we vary λ_{sem}, λ_{per}, the capacity-allocation temperature τ, and the neighborhood size N . The vertical axis represents the average accuracy across Real-Time Visual Perception and Backward Tracing.

$I + \lambda R$ by 0.56%. This result supports the residual multiplicative design: preserving query-agnostic importance as the base score while modulating it with feature representativeness provides a more effective redundancy-aware scoring formulation than the evaluated alternatives.

Calibration Dataset. To assess the sensitivity of RAKV to the calibration source (Wang et al., 2025a),we calibrate the layer-wise capacity (Tu et al., 2025; Wan et al., 2025; Huang et al., 2025; Yang et al., 2026) separately on OVO-Bench, FineVideo, and StreamingBench, while evaluating on OVO-Bench. As shown in Table 2d, the three settings achieve scores of 58.98%, 58.96%, and 58.60%, respectively, with a maximum variation of only **0.38**%. This small variation indicates that RAKV is robust to the choice of calibration data and does not rely on a specific video source.

Calibration-Set Size. To assess the sensitivity of RAKV to calibration-set size, we construct four FineVideo calibration sets containing 64, 96, 128, and 256 videos. The videos are sampled from four duration ranges (1–3, 3–5, 5–7, and at least 7 minutes). As shown in Table 2e, the 64-video calibration set achieves the highest average score of **58.96**%, while variation across all four settings is only **0.23**%. This stability demonstrates that RAKV provides a robust and data-efficient calibration procedure, maintaining effective layer-wise capacity allocation across substantially different calibration-set sizes.

Hyperparameter Sensitivity. Fig. 4 evaluates the key hyperparameters in RAKV, including the representativeness weights λ_{sem} and λ_{per}, the capacity-allocation temperature τ, and the neighborhood size N for density estimation. The two representativeness weights yield consistently stable performance across their evaluated ranges, indicating limited sensitivity to the precise balance between base importance and representativeness. The temperature shows a mild preference for intermediate values: smaller τ concentrates the layer-wise allocation, whereas larger τ makes it more uniform. RAKV is also stable across neighborhood sizes from $N = 2$ to $N = 10$, with $N = \mathbf{8}$ achieving the highest average score of **58.96**%, indicating limited sensitivity to the neighborhood size. Together, these results show that both the redundancy-aware scoring and capacity-allocation components of RAKV remain effective without fine-grained hyperparameter tuning.

System Efficiency. Table 2f compares the system costs of RAKV and HERMES on OVO-Bench under the same 6K visual KV budget. RAKV requires 0.08s time to first token (TTFT) and 19.05 GB peak GPU memory, compared with 0.07s and 19.02 GB for HERMES, corresponding to only **0.01**s additional latency and **0.03** GB extra memory overhead. Despite these marginal overheads, RAKV consistently achieves higher streaming video understanding performance than HERMES.

5 CONCLUSION

This work studies the impact of visual representation redundancy on bounded KV memory for streaming video understanding, focusing on cross-layer capacity allocation and within-layer KV retention before the user query is known. We propose RAKV, a training-free framework built on a unified redundancy-aware KV score that combines query-agnostic importance with feature representativeness. RAKV uses group-level entropy to calibrate layer-wise capacity and applies the same redundancy-aware score for within-layer retention, reducing memory spent on repeatedly represented visual content. Consistent improvements across streaming video benchmarks and MLLM backbones demonstrate the effectiveness and generalizability of the proposed redundancy-aware memory management strategy. These results highlight that bounded-memory management should preserve distinct visual information rather than treating all retained KVs as independent evidence.

6 AI USE STATEMENT

Generative AI tools are used for language polishing of the manuscript. All AI-assisted edits are reviewed by the authors, who take full responsibility for the final content of this work.

7 ETHICS STATEMENT

This work aims to advance research on efficient streaming video understanding. We do not identify any ethical concerns specific to the proposed method. As with other streaming video understanding systems, general privacy considerations may apply in real-world deployment.

8 REPRODUCIBILITY STATEMENT

All experiments are conducted using publicly available pretrained MLLMs and video understanding benchmarks. We describe the model backbones, streaming protocol, calibration procedure, memory budget, and evaluation settings in the main paper. Additional implementation details and pseudocode are provided in Appendix A, with further analyses and experimental results in Appendix B and C. We will release the source code and evaluation scripts to facilitate reproduction of the reported results.

REFERENCES

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, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. In *COLM*, 2025.

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 *CVPR*, 2024.

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 *AAAI*, 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. In *ICLR*, 2025.

Mingjing Du, Shifei Ding, and Hongjie Jia. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *KBS*, 99, 2016.

Miquel Farré, Andi Marafioti, Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. FineVideo. `https://huggingface.co/datasets/HuggingFaceFV/finevideo`, 2024.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *CVPR*, 2025.

Kai Huang, Hao Zou, Bochen Wang, Ye Xi, Zhen Xie, and Hao Wang. AirCache: Activating inter-modal relevancy KV cache compression for efficient large vision-language model inference. In *ICCV*, 2025.

Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *CVPR*, 2024.

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

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *TMLR*, 2025.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. MVBench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, 2024a.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *NeurIPS*, 2024b.

Zhijia Liang, Jiaming Li, Weikai Chen, Yanhao Zhang, Haonan Lu, and Guanbin Li. OASIS: On-demand hierarchical event memory for streaming video reasoning. In *CVPR*, 2026.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVA: Learning united visual representation by alignment before projection. In *EMNLP*, 2024.

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.

Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. TempCompass: Do video LLMs really understand videos? In *Findings of ACL*, 2024.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-ChatGPT: Towards detailed video understanding via large vision and language models. In *ACL*, 2024.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2023.

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. In *DAC*, 2026.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. OVO-Bench: How far is your video-LLMs from real-world online video understanding? In *CVPR*, 2025.

Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE TSMC*, 9(1), 1979.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. In *NeurIPS*, 2024.

Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191), 2014.

Yiqun Shen, Song Yuan, Zhengze Zhang, Xiaoliang Wang, Daxin Jiang, and Cam-Tu Nguyen. LAVa: Layer-wise KV cache eviction with dynamic budget allocation. In *Findings of EMNLP*, 2025.

Cheng Shi, Yizhou Yu, and Sibei Yang. Vision function layer in multimodal LLMs. In *NeurIPS*, 2025.

Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. VL-Cache: Sparsity and modality-aware KV cache compression for vision-language model inference acceleration. In *ICLR*, 2025.

Zhongwei Wan, Hui Shen, Xin Wang, Che Liu, Zheda Mai, and Mi Zhang. MEDA: Dynamic KV cache allocation for efficient multimodal long-context inference. In *NAACL*, 2025.

Ao Wang, Hui Chen, Jianchao Tan, Kefeng Zhang, Xunliang Cai, Zijia Lin, Jungong Han, and Guiguang Ding. PrefixKV: Adaptive prefix KV cache is what vision instruction-following models need for efficient generation. In *NeurIPS*, 2025a.

Zihao Wang, Bin Cui, and Shaoduo Gan. SqueezeAttention: 2D management of KV-cache in LLM inference via layer-wise optimal budget. In *ICLR*, 2025b.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *ICLR*, 2024.

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

Yanlai Yang, Zhuokai Zhao, Satya Narayan Shukla, Aashu Singh, Shlok Kumar Mishra, Lizhu Zhang, and Mengye Ren. StreamMem: Query-agnostic KV cache memory for streaming video understanding. *arXiv preprint arXiv:2508.15717*, 2025.

Yaoxin Yang, Peng Ye, Xudong Tan, Chongjun Tu, Maosen Zhao, Jia Hao, and Tao Chen. Revisiting multimodal KV cache compression: A frequency-domain-guided outlier-KV-aware approach. In *CVPR*, 2026.

Qianhao Yuan, Qingyu Zhang, Yanjiang Liu, Jiawei Chen, Yaojie Lu, Hongyu Lin, Jia Zheng, Xianpei Han, and Le Sun. Shortv: Efficient multimodal large language models by freezing visual tokens in ineffective layers. In *ICCV*, 2025.

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

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *NeurIPS*, 2023.

SUMMARY OF THE APPENDIX

The appendix is organized as follows:

- §A provides implementation details, including depth-specific candidate scoring and the calibration and online KV retention algorithms.
- §B derives the decomposition of the entropy at the group-level and establishes the entropy bounds within the group.
- §C provides additional experiments and analyses.
- §D discusses the impact of RAKV.
- §E discusses the limitations of RAKV memory design.

A IMPLEMENTATION DETAILS

A.1 KV SCORE IMPLEMENTATION

The KV score in Eq. 3 combines the base query-agnostic importance I_i^ℓ with feature representativeness R_i^ℓ. In our implementation, the scoring procedure is instantiated according to decoder depth (Zhang et al., 2026; Shi et al., 2025; Yuan et al., 2025). In perception layers, I_i^ℓ is a frame-level recency prior that decays exponentially with the normalized temporal distance from the most recent frame, while R_i^ℓ is computed within each frame using Eq. 2. Transition layers interpolate the resulting perception-layer KV score with query-agnostic attention using layer-dependent weights. In semantic layers, the base importance is obtained from the attention induced by a fixed query-agnostic summarization prompt over the cached visual KVs. Following StreamMem and HERMES, we use the following guidance prompt for attention-based scoring: "Summarize the video narrative, identifying main characters, key events, timeline changes, and the overall theme." During online retention, we first shortlist the highest-importance candidates at $1.5\times$ the target retention size, compute R_i^ℓ within the resulting candidate pool, and obtain the final KV score using Eq. 3. We apply min–max normalization over the corresponding candidate sets.

For both LLaVA-OneVision-7B and Qwen2.5-VL-7B, layers 0–4, 5–18, and 19–27 correspond to the perception, transition, and semantic regimes, respectively. These are zero-based implementation indices; the algorithms use $\ell = 1, \ldots, L$ for the corresponding decoder layers. We set $\lambda_{\text{per}} = 0.45$ for perception and transition layers, $\lambda_{\text{sem}} = 0.20$ for semantic layers, and $N = 8$ in Eq. 2.

A.2 ALGORITHMIC PROCEDURES

Algorithms S1 and S2 summarize the offline capacity calibration and online KV retention procedures, respectively. During calibration, group-level entropy computed from current-chunk visual candidates is averaged to determine the layer-wise capacity. During online retention, compression is triggered when the accumulated visual KV candidates exceed the calibrated capacity; the previously retained and newly generated visual KVs are then scored and selected accordingly.

Algorithm S1 Pseudo-code for Layer-Wise Capacity Calibration.

```
"""
D_cal:unlabeled calibration videos
B_tot:total visual KV budget
tau: temperature for layer-wise allocation
P: fixed query-agnostic guidance prompts
"""

def Calibrate_Capacity(D_cal, B_tot, tau, P):
   #--- 1. Collect layer-wise entropy statistics ---
   H_record = {l: [] for l in layers}

   for video in D_cal:
      for chunk in Stream(video):
         # X: projected visual-token features of the current chunk.
         # C: visual KV candidates of the current chunk at each decoder layer.
         X, C = Encode_Chunk(chunk)
         G = Build_Visual_Groups(X)
         S = Calibration_Scores(C, P)

         for l in layers:
            H = Group_Entropy(S[l], G)
            H_record[l].append(H)

   #--- 2. Estimate layer-wise capacity demand ---
   H = {l: Mean(H_record[l]) for l in layers}

   #--- 3. Allocate the total visual KV budget ---
   B = Allocate_Capacity(H, B_tot, tau)

   return B
```

Algorithm S2 Pseudo-code for Online Redundancy-Aware KV Retention.

```
"""
M_prev: retained visual KV memory from the previous step
C_cur: visual KVs from the current video chunk
B: calibrated layer-wise KV capacity
P: fixed query-agnostic guidance prompts
"""

def RAR_Retention(M_prev, C_cur, B, P):
   #--- 1. Form current visual KV candidates ---
   C = Concatenate(M_prev, C_cur)
   I, R, A, I_pool, S = {}, {}, {}, {}, {}
   C_pool = {l: C[l] for l in layers}

   if Within_Capacity(C, B):
      return C

   #--- 2. Compute depth-specific KV scores ---
   for l in layers:
      if l in perception_layers or l in transition_layers:
         I[l] = Recency_Importance(C[l])
         R[l] = Representativeness(C[l])
         S[l] = KV_Score(I[l], R[l], lambda_per)

         if l in transition_layers:
            A[l] = Attention_Importance(C[l], P)
            S[l] = Interpolate(S[l], A[l], l)

      else:
         I[l] = Attention_Importance(C[l], P)
         C_pool[l], I_pool[l] = Select_Candidates(C[l], I[l], B[l], ratio=1.5)
         R[l] = Representativeness(C_pool[l])
         S[l] = KV_Score(I_pool[l], R[l], lambda_sem)

   #--- 3. Update the bounded visual KV memory ---
   M_cur = Retain_KV(C, C_pool, S, B)

   return M_cur
```

B Detailed Derivation of Group-Level Entropy

This section provides the complete derivation of Proposition 1 in §3.2.

Proof. Fix a decoder layer ℓ and consider the visual groups $\mathcal{G}$ constructed in Eq. 5. Define the normalized score mass as $p^\ell(i) = S_i^\ell / \sum_j S_j^\ell$. Since $\mathcal{G}$ forms a partition of the candidate indices, the group mass in Eq. 6 can equivalently be written as $P^\ell(G) = \sum_{i \in G} p^\ell(i)$. For each group with $P^\ell(G) > 0$, define the within-group conditional distribution as $p^\ell(i \mid G) = p^\ell(i)/P^\ell(G)$ for $i \in G$. Thus, $p^\ell(i) = P^\ell(G)p^\ell(i \mid G)$ and $\sum_{i \in G} p^\ell(i \mid G) = 1$. We adopt the standard convention $0 \log 0 = 0$. Zero-mass groups contribute zero and are omitted from sums involving conditional distributions.

Using this factorization, the token-level entropy can be expanded over the visual groups as:

$$\begin{aligned} H^\ell_{\text{tok}} &= -\sum\nolimits_{G \in \mathcal{G}} \sum\nolimits_{i \in G} p^\ell(i) \log p^\ell(i) \\ &= -\sum\nolimits_{G \in \mathcal{G}} P^\ell(G) \sum\nolimits_{i \in G} p^\ell(i \mid G) \log \big[P^\ell(G) p^\ell(i \mid G)\big] \\ &= -\sum\nolimits_{G \in \mathcal{G}} P^\ell(G) \log P^\ell(G) - \sum\nolimits_{G \in \mathcal{G}} P^\ell(G) \sum\nolimits_{i \in G} p^\ell(i \mid G) \log p^\ell(i \mid G) \\ &= H^\ell_{\text{grp}} + H^\ell_{\text{within}}. \end{aligned} \tag{10}$$

The first term is exactly the group-level entropy in Eq. 7, while the second term is the group mass weighted within-group entropy H^ℓ_{within}.

It remains to establish the bounds on H^ℓ_{within} in Proposition 1. For each positive-mass group G, the conditional entropy is nonnegative since $-x \log x \geq 0$ for $x \in [0, 1]$. For the upper bound, apply Jensen's inequality to the concave function $-x \log x$ with equal weights $1/|G|$. Since the conditional probabilities sum to one, their average is $1/|G|$, giving:

$$-\sum\nolimits_{i \in G} p^\ell(i \mid G) \log p^\ell(i \mid G) \leq -|G| \, \tfrac{1}{|G|} \log \tfrac{1}{|G|} = \log |G|. \tag{11}$$

Multiplying these conditional-entropy bounds by $P^\ell(G)$ and summing over the visual groups gives the bounds in Eq. 8. The lower bound is attained exactly when the mass of each positive-mass group is concentrated on a single representation. By strict concavity, the upper bound is attained exactly when $p^\ell(i \mid G) = 1/|G|$ for every candidate in each positive-mass group. In particular, for fixed $P^\ell(G)$, under uniform allocation the within-group contribution is $P^\ell(G) \log |G|$, and therefore grows logarithmically with the number of KV representations in G. □

C Additional Experiments

C.1 Generalization to Offline Video Understanding

Although RAKV is designed for bounded-memory streaming video understanding, its redundancy-aware KV retention is not specific to streaming benchmarks. We further evaluate its generalization to offline long-video understanding on VideoMME and EgoSchema.

Table S1: Offline video understanding accuracy (%) on VideoMME and EgoSchema. † denotes our reproduced HERMES results. Results of RAKV are reported as mean ± standard deviation over three runs.

Method	VideoMME (%)		EgoSchema (%)
	Overall	Long	
HERMES†	56.11	44.78	59.15
RAKV (Ours)	**57.31**± 0.19	**48.82**± 0.36	**59.42**± 0.06
Δ (RAKV − HERMES)	+1.20	**+4.04**	+0.27

As shown in Table S1, RAKV consistently outperforms HERMES on both benchmarks, improving VideoMME Overall by 1.20% and EgoSchema by 0.27%. The improvement is particularly pronounced on the Long subset of VideoMME, where RAKV achieves a **4.04**% gain, suggesting that our method more effectively uses the fixed memory budget to preserve distinct visual information over extended video contexts.

For HERMES, we use the released implementation and reproduce its results under the same evaluation protocol and hardware environment used for RAKV. The reproduced scores are lower than those reported in the original paper; a similar reproduction gap has also been reported in the public repository (Issue #10).

C.2 Visualization of Redundancy-Aware KV Retention

To better illustrate the effect of our redundancy-aware KV score, we visualize the visual KVs retained using importance alone and using our full KV score under the same retention budget.

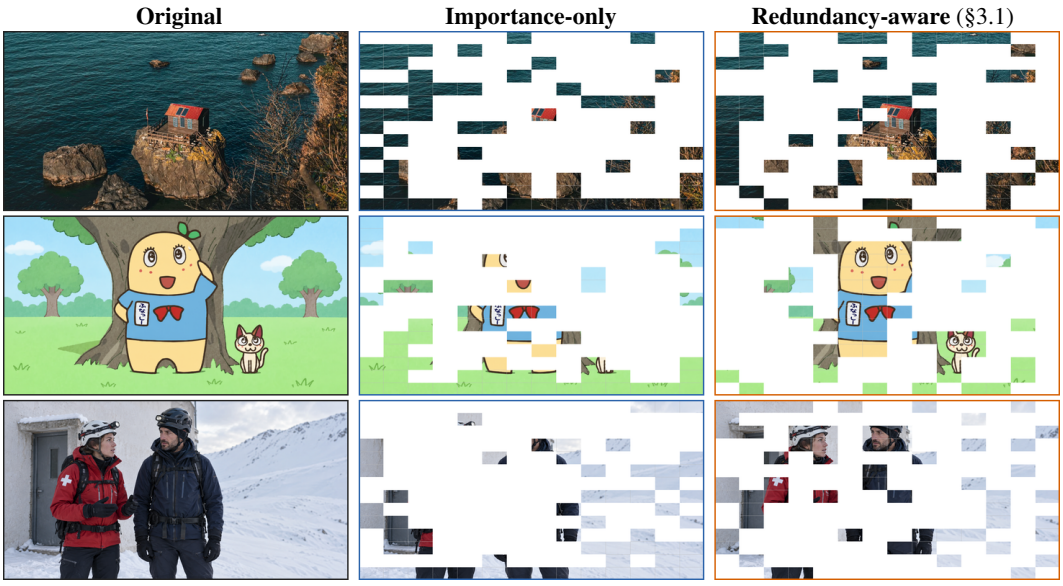

Figure S1: Qualitative comparison of visual KV retention under the same memory budget. Each row shows one example, with the original frame on the left, visual KVs retained by base importance in the middle, and those retained by the redundancy-aware KV score on the right. Visible patches correspond to retained visual KVs, while white regions indicate discarded KVs.

As illustrated in Fig. S1, importance-only selection tends to concentrate retained KVs on visually similar regions, resulting in repeated coverage of the same salient content. Incorporating feature representativeness favors the retention of visually informative content, such as recognizable object and facial details, under the same retention budget. This qualitative pattern is consistent with the intended role of the redundancy-aware score: reducing repeated retention and increasing the coverage of distinct visual information under the same memory budget.

C.3 Visualization of Layer-Wise Capacity Allocation

To further illustrate the effect of group-level entropy calibration on cross-layer capacity allocation, we visualize the calibrated layer-wise KV capacity of LLaVA-OneVision-7B.

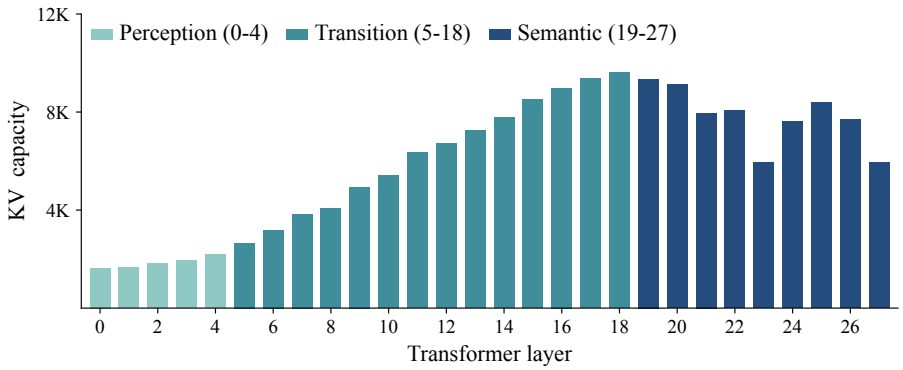

Figure S2: Layer-wise visual KV capacity allocation for LLaVA-OneVision-7B obtained from group-level entropy calibration under a fixed total memory budget.

As shown in Fig. S2, group-level entropy calibration produces a non-uniform capacity profile across decoder layers. The resulting allocation redistributes the fixed total KV budget according to the calibrated capacity demand of each layer while reducing the influence of repeatedly represented visual content on capacity estimation.

D Impact Statement

This work reveals that representation redundancy affects both key decisions in bounded visual memory for streaming video understanding: how a fixed memory budget is allocated across decoder layers and which visual KVs are retained within each layer. RAKV provides a practical approach to addressing these effects through group-level capacity calibration and redundancy-aware KV retention, respectively. More broadly, our findings suggest that future streaming memory systems should manage bounded memory according to the distinct visual information preserved by the memory as a whole, rather than evaluating visual KVs only as independent units. This perspective highlights representation redundancy as a shared design factor for both cross-layer capacity allocation and within-layer retention, and may inform more effective memory mechanisms for long-horizon streaming video understanding.

E Limitations

RAKV improves the use of bounded visual KV memory by favoring visual KVs that are both important and representative under a fixed budget. Nevertheless, the current retrieval-free design relies on irreversible KV eviction to maintain bounded active memory. Historical KVs removed during streaming are therefore no longer accessible when a future user query arrives. This limitation motivates future memory mechanisms beyond hard eviction. Instead of permanently discarding historical information, such mechanisms could consolidate information from evicted states into compact latent memory that remains accessible to future queries without explicit retrieval, while preserving bounded active memory and low streaming overhead.