KEYREC: BOUNDED VISUAL MEMORY FOR STREAMING AND LONG-VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models are increasingly used to understand long videos and continuous streams. However, dense visual tokens accumulate with video duration, making long-context inference prohibitively expensive. Existing training-free visual-token selection methods reduce this cost by retaining informative tokens, but may lose coherent event evidence and fail to distinguish detailed recent observations from long-range history. We propose **KeyRec**, a training-free framework for constructing bounded visual memory. During query-agnostic writing, KeyRec preserves fine-grained recent observations in a visual cache and organizes historical evidence into a structured event bank. Candidate events are proposed according to their novelty relative to previously stored events and maintained through an online add–merge–evict update. When a question arrives, a text-only router adaptively allocates a fixed readout budget between recent and event memory, without reprocessing historical frames. KeyRec operates on model-facing visual embeddings and supports both modular encoder–projector VLMs and the encoder- and projector-free NEO-ov architecture. Across four streaming and long-video benchmarks and three VLM backbones, KeyRec achieves the best compressed performance in 13 of 15 settings using only 10% of the dense decoder-facing visual-token budget. It outperforms the strongest compressed baseline by 2.21–18.37 points on real-time questions, achieves the best compressed result in five of six long-video settings, and performs best in every NEO-ov 2B setting.

1 INTRODUCTION

Recent advances in vision-language models (VLMs) have enabled increasingly capable reasoning over visual content, extending their applications from images and short clips to long-form recordings and continuous video streams (Zhang et al., 2024; Chen et al., 2025b; Zhang et al., 2025a; Cho et al., 2026). However, a VLM typically represents each frame with dozens or hundreds of visual tokens. Consequently, the visual sequence grows linearly with the number of frames, while dense self-attention during prefilling incurs quadratic computation in the sequence length (Ning et al., 2025; Wei et al., 2025; Zhang et al., 2025b). This accumulation is particularly problematic for streaming video, whose duration may be unknown and whose queries can arrive at any time (Di et al., 2025; Yang et al., 2025c), as well as for long videos, where dense visual context can exceed the available memory or context budget and may be repeatedly processed for multiple questions (Jin et al., 2025; Chen et al., 2026). Compressing the growing visual context is therefore essential for scalable video understanding (Yang et al., 2025b; Tao et al., 2025; Shao et al., 2026).

Recent training-free methods exploit redundancy among visual tokens by identifying informative regions through spatial-temporal scoring, retaining tokens that differ from adjacent frames or historical anchors, or maintaining a compact semantic representation of the observed stream (Wang et al., 2026b; Chen et al., 2026; Song et al., 2026). These approaches effectively remove repetitive backgrounds and other low-information content, substantially reducing the visual context. However, we observe three limitations when the retained representations serve as visual memory. First, novelty is commonly measured against a recent frame, coarse anchors, or a global semantic representation, without explicitly organizing history into temporally localized events. As a result, evolving events may be fragmented across updates, while visually similar but temporally distinct occurrences may be conflated. Second, independently selecting important tokens may discard the supporting context that connects an actor, action, object, and scene into coherent event evidence. Third, existing meth-

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.

ods often apply a common retention policy across the video, forcing detailed recent observations and long-range historical evidence to compete under the same policy. Uniformly sparse memory may therefore preserve broad semantics while losing the fine-grained evidence required by questions such as "What is happening right now?" These limitations motivate treating compression not only as token selection, but also as bounded visual-memory organization.

To address these challenges, we propose **KeyRec**, a training-free framework that separates query-agnostic memory construction from query-adaptive, fixed-budget readout. During writing, KeyRec maintains a dense *recent cache* and a structured *key event memory*. Incoming frames propose novel candidate events, which are maintained through an online add–merge–evict update that uses both semantic similarity and temporal proximity to consolidate related observations while keeping the historical memory bounded. When a question arrives, a text-only pass of the same frozen VLM allocates the readout budget between recent and event memory and retrieves historical events by query relevance or temporal coverage, without reprocessing previous frames. Because KeyRec operates on model-facing visual embeddings, the same mechanism applies to both modular encoder–projector VLMs and native one-vision models. We instantiate it on Gemma 4 E2B/E4B (Gemma Team, 2026) and NEO-ov 2B (Diao et al., 2026), preserving the temporal and spatial indices of selected NEO-ov tokens. Across four streaming (Niu et al., 2025; Lin et al., 2026a) and long-video benchmarks (Wu et al., 2024; Fu et al., 2026) and three VLM backbones, KeyRec achieves the best compressed performance in 13 of 15 settings using only 10% of the dense decoder-facing visual-token budget. Our contributions are summarized as follows:

- We propose KeyRec, a training-free framework for bounded visual memory that decouples query-agnostic memory construction from query-adaptive, fixed-budget readout.
- KeyRec organizes memory into detailed recent evidence and temporally localized historical events, maintained through an online semantic-temporal add–merge–evict update, and supports native one-vision VLMs by preserving token structure.
- Across four benchmarks and three VLM backbones, KeyRec achieves the best compressed result in 13 of 15 settings using only 10% of the dense decoder-facing visual-token budget.

2 METHOD

2.1 PROBLEM FORMULATION AND OVERVIEW

Video question answering. Consider a video stream $\mathcal{V} = \{x_1, x_2, \ldots\}$. The model-native visual embedding mechanism Φ maps each frame x_t to n visual tokens $H_t = \Phi(x_t) = \{\mathbf{h}_{t,i}\}_{i=1}^{n}$. A dense VLM answering a query q at time τ conditions on $[H_1; \ldots; H_\tau]$. This requires retaining and processing $n\tau$ visual tokens: storage grows linearly with τ, while dense self-attention cost grows quadratically. Although only streaming video imposes strict causality, both streaming and long-video understanding benefit from a bounded, query-agnostic memory that can be reused across queries and read adaptively under a fixed inference budget.

We formalize this as *query-agnostic memory writing with query-adaptive, fixed-budget reading*:

$$\begin{aligned} \mathcal{M}_t &= \mathrm{Update}_{\mathcal{M}}(\mathcal{M}_{t-1}, H_t), & |\mathcal{M}_t| &\le S, \\ Z_q &= \mathrm{Read}(\mathcal{M}_\tau, q), & |Z_q| &\le B, \end{aligned} \tag{1}$$

where S is the persistent visual-token storage budget and B is the per-query readout budget. The answer is generated as $\hat{y} = \mathrm{VLM}(Z_q, q)$ without revisiting the video.

KeyRec overview. KeyRec instantiates this interface as $\mathcal{M}_t = (\mathcal{C}_t, \mathcal{E}_t)$, where $\mathcal{C}_t$ is a *recent cache* preserving detailed evidence from the latest frames and $\mathcal{E}_t = \{e_k\}_{k=1}^{K_t}$ is a bounded *key event memory* retaining informative historical evidence. The cache stores at most N_C visual tokens, while the event bank contains at most K events of at most m tokens each, giving total storage $|\mathcal{C}_t| + \sum_{e_k \in \mathcal{E}_t} |P_k| \le N_C + Km \le S$. Both memories are constructed without access to the query. At query time, a text-only router allocates the fixed budget B between recent and event memory and selects the evidence presented to the VLM.

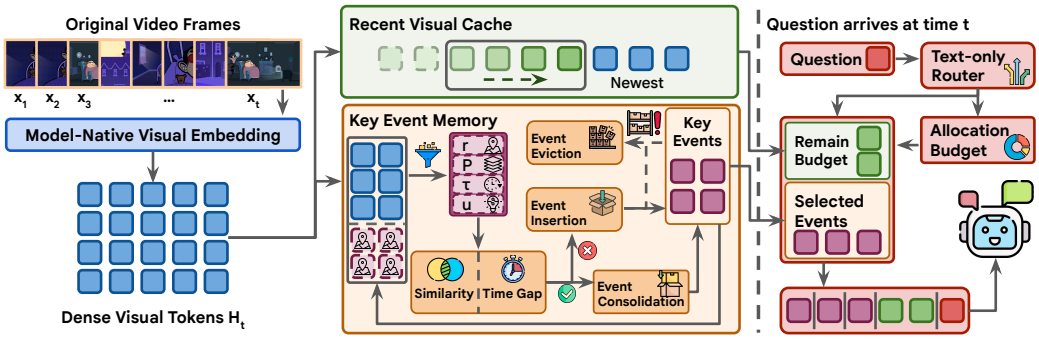


Figure 1: **Overview of KeyRec.** During query-agnostic writing, incoming frames update a recent cache and a bounded key-event memory through event insertion, consolidation, and eviction. At query time, a text-only router allocates a fixed readout budget between recent and historical evidence for query-adaptive inference without replaying previous frames.

2.2 Query-Agnostic Memory Construction

Given the visual tokens H_t of frame x_t, KeyRec updates the two memory components using only H_t and the memory state $\mathcal{M}_{t-1}$ from preceding frames.

Recent visual cache. The recent cache $\mathcal{C}_t$ is an ordered sequence of at most N_C visual tokens that preserves dense evidence from the latest observations. For each incoming frame, it appends H_t and retains only the most recent N_C tokens, $\mathcal{C}_t = \mathrm{Tail}_{N_C}(\mathcal{C}_{t-1} \| H_t)$; as new frames arrive, older tokens are evicted. Because this fixed-capacity cache only preserves recent evidence, KeyRec complements it with a bounded key-event memory for long-range history.

Key event memory. The key event memory is a bounded collection of historical visual evidence, initialized as $\mathcal{E}_0 = \varnothing$. At time t, it contains

$$\mathcal{E}_t = \{e_k\}_{k=1}^{K_t}, \quad e_k = (\mathbf{r}_k, P_k, \boldsymbol{\tau}_k, u_k), \quad K_t \le K, \tag{2}$$

where K is a fixed event capacity. For each event, $\mathbf{r}_k \in \mathbb{R}^d$ is a compact route embedding, $P_k \in \mathbb{R}^{m_k \times d}$ is its stored visual tokens with $m_k \le m$, $\boldsymbol{\tau}_k$ contains the corresponding token timestamps, and u_k denotes its event utility. The key event memory stores at most Km visual tokens.

▷ **Candidate construction.** Each incoming frame proposes a candidate event that is not sufficiently represented by the current event memory $\mathcal{E}_{t-1}$. For each token $\mathbf{h}_{t,i} \in H_t$, we define its novelty as

$$\nu_{t,i} = 1 - \max_{e_k \in \mathcal{E}_{t-1}} \cos(\mathbf{h}_{t,i}, \mathbf{r}_k), \tag{3}$$

with $\nu_{t,i} = 1$ when $\mathcal{E}_{t-1}$ is empty. $\nu_{t,i}$ measures whether a visual token contributes information not yet represented by the historical event memory, rather than merely measuring its within-frame saliency. A larger $\nu_{t,i}$ indicates that the token is less represented by the stored event routes.

Let $\mathcal{I}_t$ contain the indices of the m most novel tokens in frame t: $\mathcal{I}_t = \text{Top-m}(\{\nu_{t,i}\}_{i=1}^{n})$, we summarize the selected tokens into a candidate route using their normalized novelty scores:

$$\mathbf{r}_t = \mathrm{Norm}\left(\sum_{i \in \mathcal{I}_t} \frac{\exp(\nu_{t,i})}{\sum_{j \in \mathcal{I}_t} \exp(\nu_{t,j})} \mathbf{h}_{t,i} \right). \tag{4}$$

The same selected tokens constitute the candidate event tokens P_t, with their timestamps retained in $\boldsymbol{\tau}_t$. We define the candidate utility as their average novelty: $u_t = \sum_{i \in \mathcal{I}_t} \nu_{t,i} / |\mathcal{I}_t|$. Together, these quantities form the candidate event $\widetilde{e}_t = (\mathbf{r}_t, P_t, \boldsymbol{\tau}_t, u_t)$, which is subsequently incorporated into the bounded event bank.

▷ **Bounded bank update.** Each candidate is either inserted as a new event or consolidated with an existing event; when insertion exceeds the capacity K, one stored event is evicted. To determine whether the candidate continues an existing event, we first identify the most similar stored route:

$$s_{t,k} = \cos(\mathbf{r}_t, \mathbf{r}_k), \qquad k^\star = \operatorname*{arg\,max}_{1 \le k \le K_{t-1}} s_{t,k}. \tag{5}$$

Because visually similar content may recur at distant times, we additionally require temporal proximity. Let $g_{t,k^\star} = \min(\boldsymbol{\tau}_t) - \max(\boldsymbol{\tau}_{k^\star})$ denote the temporal gap between the candidate and its matched event. The candidate is mergeable with $e_{k^\star}$ only if

$$s_{t,k^\star} \geq \gamma, \qquad g_{t,k^\star} \leq \delta, \tag{6}$$

where γ and δ control semantic similarity and temporal locality, respectively. This prevents temporally distant recurring observations from being collapsed into the same event.

Candidate insertion. If the event bank is empty or Eq. 6 is not satisfied, the candidate is appended as a distinct event, $\overline{\mathcal{E}}_t = \mathcal{E}_{t-1} \cup \{\widetilde{e}_t\}$. If this exceeds capacity K, the eviction rule below is applied.

Event consolidation. Otherwise, the candidate is consolidated with $e_{k^\star}$. Its route is updated by a normalized moving average:

$$\mathbf{r}_{k^\star} \leftarrow \mathrm{Norm}\left((1-\beta)\mathbf{r}_{k^\star} + \beta \mathbf{r}_t\right). \tag{7}$$

The stored visual tokens and timestamps are updated jointly:

$$\begin{aligned} P_{k^\star} &\leftarrow \mathrm{Uniform}_m\Big(\mathrm{TimeSort}\left[P_{k^\star}; P_t\right]\Big); \\ \boldsymbol{\tau}_{k^\star} &\leftarrow \mathrm{Uniform}_m\Big(\mathrm{TimeSort}\left[\boldsymbol{\tau}_{k^\star}; \boldsymbol{\tau}_t\right]\Big). \end{aligned} \tag{8}$$

$\mathrm{TimeSort}$ orders the visual tokens and timestamps by source time. To enforce the per-event storage capacity, $\mathrm{Uniform}_m$ leaves sequences with at most m tokens unchanged, while deterministically retaining m uniformly spaced entries. Such an operation preserves the temporal extent of the merged event. Event utility records the strongest candidate novelty and is updated as $u_{k^\star} \leftarrow \max(u_{k^\star}, u_t)$, which avoids diluting brief distinctive changes with redundant adjacent observations.

Capacity-based eviction. When the key event bank exceeds K events, KeyRec removes the event with the lowest retention score, which balances visual utility and temporal coverage. Let $c_k = (\min(\boldsymbol{\tau}_k) + \max(\boldsymbol{\tau}_k))/2$ be the center time of event e_k. We define

$$d_k^{\mathrm{temp}} = \min_{j \neq k} |c_k - c_j|, \qquad S_{\mathrm{ret}}(e_k) = u_k + \lambda d_k^{\mathrm{temp}}. \tag{9}$$

The utility term favors events containing distinctive visual changes, while the temporal term favors events that cover otherwise underrepresented portions of the video. KeyRec evicts $k_{\mathrm{evict}} = \arg\min_k S_{\mathrm{ret}}(e_k)$ to restore $|\mathcal{E}_t| \leq K$.

2.3 Query-Adaptive Fixed-Budget Readout

Different questions require different evidence from the same visual memory: current-state questions favor detailed recent observations, whereas retrospective questions require broader historical coverage. KeyRec therefore uses the same frozen VLM as a text-only memory router.

We define five readout states, ordered from recent- to event-oriented: AllRecent, HeavyRecent, Balanced, HeavyEvent, and AllEvent, collectively denoted by $\mathcal{Z}$. Given question q, the router predicts

$$z_q = \mathrm{Router}_{\mathrm{VLM}}(q), \qquad z_q \in \mathcal{Z}. \tag{10}$$

The text-only memory router observes only the question and a textual description of the two memory sources; it neither accesses nor reprocesses the video. Each state jointly determines the recent-event allocation and the strategy for selecting historical events.

Let $\pi_q = \pi(z_q) \in [0,1]$ denote the fraction of the readout assigned to event memory, with $\pi_q = 0$ and 1 corresponding to AllRecent and AllEvent, respectively. Given a total visual-token readout budget B and event size m, the numbers of selected events and recent tokens are

$$k_q = \min\left\{K, \mathrm{round}\left(\frac{\pi_q B}{m}\right)\right\}, \qquad r_q = B - k_q m. \tag{11}$$

The router state also determines how the k_q historical events are selected. For HeavyRecent and Balanced, KeyRec uses relevance-oriented retrieval, selecting the events whose routes have the highest cosine similarity to the mean-pooled question embedding $\mathbf{q}$. For HeavyEvent and

ALLEVENT, it instead favors temporal coverage by ordering events by center time and uniformly selecting k_q events across the sequence. ALLRECENT retrieves no historical events. Selected events are restored to temporal order before decoding.

The recent contribution is $C_q = \mathrm{Tail}_{r_q}(\mathcal{C}_\tau)$. Together, C_q and the selected event tokens form the fixed-budget readout Z_q, with $|Z_q| \leq B$, which is presented to the VLM.

Segmented memory presentation. We serialize the selected readout Z_q as separate memory segments with neutral numbered labels, preserving their boundaries without assigning semantic event types. The cached VLM-facing embeddings are inserted directly at the corresponding segment positions. Appendix A.1 compares this default serialization with alternative designs.

2.4 KEYREC ON NATIVE ONE-VISION MODELS

Existing visual-token compression methods have predominantly been developed for modular VLMs with a dedicated vision encoder and visual projector. Native one-vision models such as NEO-ov instead use lightweight patch embeddings and jointly process visual and textual tokens in a unified backbone (Diao et al., 2026), making methods that rely on ViT-specific features or attention signals difficult to transfer directly (Chen et al., 2026). We therefore study whether bounded visual memory can be applied to this emerging architecture with minimal adaptation.

Because KeyRec operates directly on model-facing visual embeddings, its memory construction and query-adaptive readout remain unchanged. The only architecture-specific adaptation for NEO-ov is to preserve the positional metadata of each retained token, which we represent as $\xi_i = (\mathbf{h}_i, \tau_i, p_i^h, p_i^w)$, where τ_i is the source frame and (p_i^h, p_i^w) is its original spatial location. KeyRec operates only on $\mathbf{h}_i$, while the metadata follows the same selection and subsampling indices. At read time, selected tokens are ordered by source time and restored to their native spatial and temporal positions before being inserted into the corresponding `<IMG_CONTEXT>` locations. Thus, adapting KeyRec to NEO-ov requires only structure-preserving positional bookkeeping, without modifying the underlying memory algorithm or reconstructing dense visual grids.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

Benchmarks and metrics. We evaluate KeyRec on both streaming and long video understanding benchmarks. For streaming evaluation, we use the nine backward-tracing and real-time tasks of OVO-Bench (Niu et al., 2025) and the Real-Time Visual Understanding subset of Streaming-Bench (Lin et al., 2026a), where each query accesses only its preceding video prefix. For long-video evaluation, we use the visual-only validation split of LongVideoBench (Wu et al., 2024) and the full Video-MME-v2 benchmark (Fu et al., 2026). We exclude subtitles and audio to isolate visual-memory compression and report accuracy following each benchmark's official evaluation protocol.

VLM backbones and baselines. We evaluate Gemma 4 E2B and E4B as modular encoder–projector VLMs (Gemma Team, 2026), and NEO-ov 2B as a native one-vision model (Diao et al., 2026). We compare against three representative training-free visual-token compression methods: **STC-Pruner**, which scores tokens against current and historical prototypes (Wang et al., 2026b); **StreamingTOM-CTR**, which combines adjacent-frame changes with ViT attention (Chen et al., 2026); and **CausalMem**, which maintains a fixed-budget low-rank semantic basis (Song et al., 2026). **Vanilla** retains all sampled visual tokens without compression. StreamingTOM-CTR is evaluated only on Gemma because it requires attention signals from a dedicated vision encoder. All methods use frozen backbones and the same frame-sampling protocol within each setting.

Implementation details. For the main experiments, we sample 32 frames from each streaming prefix[1] and 64 frames from each complete long video, processing frames chronologically. All com-

[1]STC-Pruner and StreamingTOM-CTR retain a fixed token fraction per frame, so their retained context grows with stream length. We therefore cap the main comparison at 32 frames to match all methods under the same decoder-facing budget, and separately evaluate fixed-budget full-prefix 1-fps streaming in Section 3.7.

Table 1: **Main results on streaming and long-video understanding benchmarks.** Vanilla retains all visual tokens, whereas compressed methods use a matched decoder-facing budget of approximately 10% of the dense visual input. Bold indicates the best compressed result in each row; "–" denotes an unsupported architecture (STOM-CTR: StreamingTOM-CTR).

Benchmark	Model	Subset	Vanilla	STC-Pruner	STOM-CTR	CausalMem	KeyRec
OVO-Bench	Gemma 4 E2B	Backward	43.22	41.20	42.30	**45.31**	43.83
OVO-Bench	Gemma 4 E2B	Realtime	50.72	48.02	47.23	55.49	**57.70**
OVO-Bench	Gemma 4 E4B	Backward	53.86	53.01	50.49	54.45	**54.45**
OVO-Bench	Gemma 4 E4B	Realtime	59.83	51.67	53.26	62.51	**64.81**
OVO-Bench	NEO-ov 2B	Backward	38.66	35.22	–	39.09	**41.90**
OVO-Bench	NEO-ov 2B	Realtime	58.07	36.84	–	44.41	**62.78**
StreamingBench	Gemma 4 E2B	Realtime	51.28	52.91	51.36	58.01	**61.29**
StreamingBench	Gemma 4 E4B	Realtime	63.73	56.44	60.25	63.61	**66.49**
StreamingBench	NEO-ov 2B	Realtime	69.06	50.68	–	55.56	**69.10**
LongVideoBench	Gemma 4 E2B	Validation	35.90	33.06	34.64	34.11	**35.60**
LongVideoBench	Gemma 4 E4B	Validation	49.14	43.38	43.98	43.08	**45.77**
LongVideoBench	NEO-ov 2B	Validation	52.51	44.43	–	42.93	**49.29**
Video-MME-v2	Gemma 4 E2B	Overall	18.47	14.16	**15.69**	15.31	15.41
Video-MME-v2	Gemma 4 E4B	Overall	20.09	14.34	15.91	14.84	**17.63**
Video-MME-v2	NEO-ov 2B	Overall	22.06	17.56	–	17.47	**20.03**

pressed methods use approximately 10% of the dense decoder-facing visual-token budget. KeyRec maintains a larger query-agnostic persistent memory of approximately 20% of the dense visual input and selects a 10% readout for each query, whereas the baselines decode their complete retained buffers. Exact memory configurations, router codebooks and prompts, and baseline hyperparameters are provided in Appendix B.

3.2 Main Experimental Results

Table 1 compares KeyRec with uncompressed inference and representative training-free compression methods under the same frame-sampling protocol. **Observation 1: KeyRec is particularly effective for streaming video understanding.** KeyRec achieves the best compressed performance in eight of nine streaming settings and outperforms the strongest compressed baseline by 2.21–18.37 points across OVO-Bench Realtime and StreamingBench. It also matches or exceeds Vanilla in all nine settings, with gains of up to 10.01 points. Although Vanilla retains all visual evidence, it is not an oracle upper bound: redundant streaming prefixes may dilute decisive recent evidence with repeated or irrelevant context. KeyRec instead forms a structured evidence bottleneck in which the event bank compresses redundant history, the recent cache preserves fine-grained latest observations, and the router adapts their allocation to the query. **Observation 2: KeyRec provides a strong, but not lossless, long-video compression trade-off.** KeyRec obtains the best compressed result in five of six long-video settings and consistently outperforms the other compressed methods on LongVideoBench. Its 0.30–3.37-point accuracy gap below Vanilla indicates that the fixed 10% readout preserves much of the model's long-video capability but does not retain all fine-grained evidence. **Observation 3: KeyRec remains effective across substantially different VLM architectures.** KeyRec achieves the best compressed performance in every NEO-ov 2B setting across both streaming and long-video benchmarks. Visual-token compression remains underexplored for encoder- and projector-free one-vision models, whose positional mechanisms and visual interfaces differ from conventional modular VLMs. Our structure-preserving adaptation provides an initial exploration of this setting and suggests the importance of preserving model-native token structure when extending visual memory to emerging architectures.

3.3 Scaling with the Decoder-Facing Readout Budget

Figure 2 studies decoder-facing budget scaling on StreamingBench.[2] Because KeyRec selects each readout from a persistent memory approximately twice as large, the comparison also controls for

[2]The two smallest ratios stress-test the low-budget regime; at 2.5%, the readout contains only 56 tokens, fewer than one dense Gemma frame.

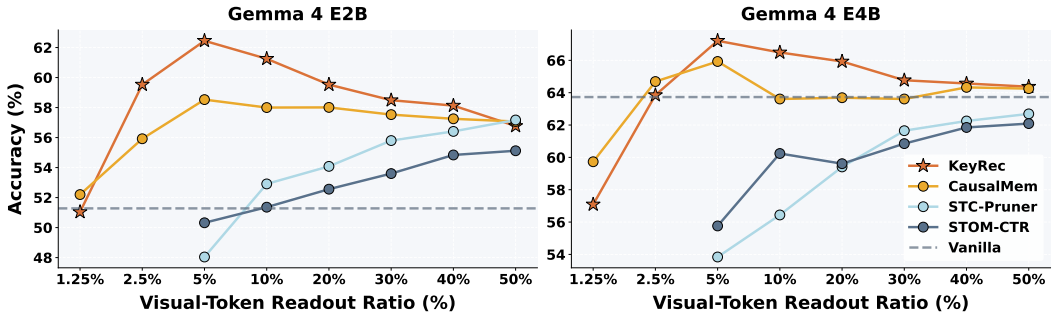


Figure 2: **Readout-budget scaling on StreamingBench.** Dashed horizontal lines denote uncompressed Vanilla performance. The horizontal axis reports the decoder-facing visual-token ratio. KeyRec selects each readout from a visual-token memory approximately twice as large, whereas each baseline decodes its complete retained token buffer (STOM-CTR: StreamingTOM-CTR).

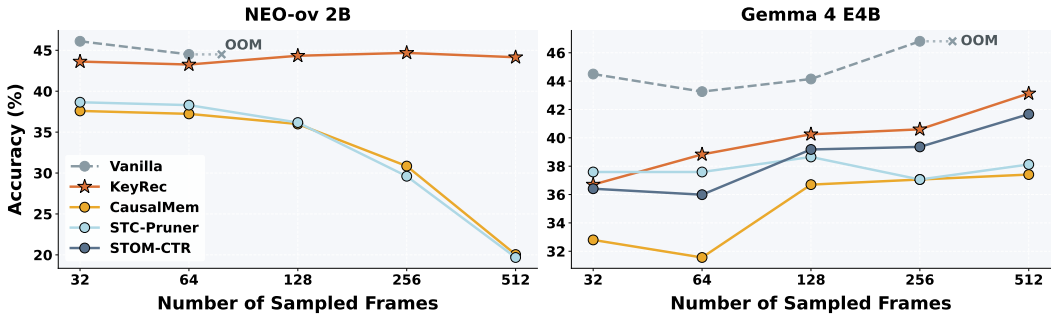


Figure 3: **Scaling with the number of sampled frames on LongVideoBench's (900s, 3600s] duration group.** All compressed methods use a 10% decoder-facing ratio. Missing Vanilla results indicate memory limitations; StreamingTOM-CTR (STOM-CTR) does not support NEO-ov.

its larger candidate pool: At approximately matched persistent storage, KeyRec achieves higher accuracy while exposing only half as many visual tokens to the decoder. Specifically, we compare KeyRec at a 5% readout with baselines at 10%, and KeyRec at 10% with baselines at 20%. This indicates that its advantage does not arise simply from storing more tokens, but from organizing a structured candidate memory and selecting a query-adaptive readout.

Across both backbones, KeyRec improves rapidly at small budgets and peaks around a 5% readout. Performance gradually declines beyond the peak, and the best configurations can exceed uncompressed Vanilla, suggesting that additional visual context is not uniformly beneficial when redundant or weakly relevant evidence is introduced.

3.4 Scaling with the Number of Sampled Frames

Figure 3 evaluates scaling on LongVideoBench videos between 900 and 3,600 seconds with a fixed 10% decoder-facing ratio. As more frames are sampled, each method observes denser temporal evidence while receiving a proportionally larger absolute readout budget. Dense Vanilla quickly becomes infeasible, with Gemma running out of memory at 512 frames and NEO-ov at 128 frames or more; where feasible, compressed methods generally remain below Vanilla, indicating that aggressive token reduction does not fully preserve dense long-video performance. The two backbones nevertheless exhibit different scaling behavior: on Gemma E4B, compressed methods generally benefit from additional frames, with KeyRec performing best from 64 frames onward, whereas on NEO-ov, STC-Pruner and CausalMem degrade substantially while KeyRec remains stable across 32–512 frames, staying above 44% accuracy at 512 frames compared with below 25% for both baselines. This divergence suggests that existing token-selection policies may not transfer directly to native one-vision models, where preserving model-native spatial and temporal structure becomes increasingly important over longer videos. KeyRec provides an initial exploration of this setting through lightweight structure-preserving adaptation without modifying the underlying memory algorithm.

3.5 Effect of Query-Adaptive Memory Allocation

Figure 4 compares the query router with three fixed readout policies on Gemma E4B, while keeping the constructed memory and total decoder-facing budget unchanged. The optimal fixed allocation varies substantially across evaluation settings. OVO-Bench Backward benefits most from event memory, whereas OVO-Bench Realtime strongly favors detailed recent evidence and degrades sharply under an event-only readout. LongVideoBench instead performs best with a mixture of recent and historical evidence. Consequently, no single fixed allocation performs consistently well across all three settings. The router adapts the recent–event allocation to each question, matching the event-only policy on backward reasoning while outperforming the fixed policies on real-time and long-video understanding. It therefore achieves the best or tied performance in every setting, demonstrating the value of query-adaptive readout over a universal memory allocation.

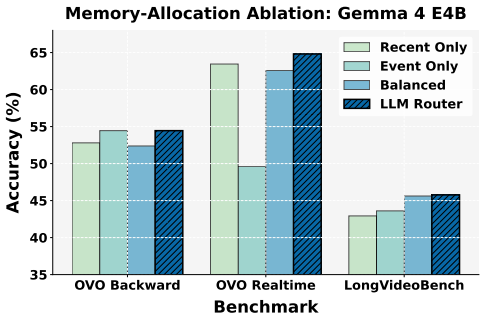


Figure 4: **Effect of query-adaptive memory allocation on Gemma 4 E4B.** All policies use the same constructed memory and decoder-facing budget; only the allocation between recent and event memory differs.

3.6 Latency and Memory Analysis

Table 2 reports E2E latency and peak GPU memory for Gemma E4B on LongVideoBench, comparing KeyRec with dense Vanilla and StreamingTOM-CTR (STOM-CTR), the strongest compressed competitor at large frame counts. E2E latency is measured from ingestion of the first predecoded frame to the first generated token and includes vision encoding, memory updates, and query processing; peak memory covers both ingestion and query processing, including model weights.

Table 2: **Latency and memory scaling on LongVideoBench with Gemma E4B.** Compressed methods use a 10% decoder-facing ratio.

Frames	Method	E2E (s)	Peak Mem (GiB)
64	Vanilla	5.73	17.09
	STOM-CTR	5.92	15.05
	KeyRec	7.08	15.06
256	Vanilla	26.55	39.23
	STOM-CTR	20.04	15.47
	KeyRec	22.81	16.17
512	Vanilla	OOM	OOM
	STOM-CTR	39.12	16.57
	KeyRec	43.08	18.93

KeyRec incurs a largely frame-independent overhead from text-only routing, which is progressively amortized as the video grows: relative to STOM-CTR, its E2E overhead decreases from approximately 19% at 64 frames to 10% at 512 frames. KeyRec remains feasible throughout, including at 512 frames, where dense Vanilla runs out of memory; Vanilla already reaches 39.23 GiB at 256 frames. KeyRec uses moderately more memory than STOM-CTR due to its larger candidate memory. Together with the stronger compressed accuracy in Section 3.4, these results show that KeyRec trades modest overhead for higher video-understanding performance while scaling substantially better than dense inference.

3.7 Fixed-Budget 1-FPS Streaming Evaluation

Our main experiments cap each streaming prefix at 32 sampled frames to enable controlled comparisons with methods whose retained tokens grow with the number of frames. We further evaluate KeyRec under a full-prefix 1-FPS protocol, where all frames available before each query are processed while the visual-memory and decoder-facing budgets remain fixed. We compare KeyRec with CausalMem, which also maintains a globally bounded memory, and omit baselines that retain a fixed token ratio per frame and therefore grow linearly with the observed stream.

Table 3: **Fixed-budget 1-FPS evaluation.**

Subset	Method	E2B	E4B
OVO-Bench			
Backward	CausalMem	46.70	**56.58**
	KeyRec	**48.61**	56.15
Realtime	CausalMem	57.12	62.96
	KeyRec	**57.49**	**65.71**
StreamingBench			
Realtime	CausalMem	61.20	67.77
	KeyRec	**63.21**	**68.45**

The mean video durations of OVO-Bench and StreamingBench are approximately 235 and 264 seconds, respectively. We use 256 frames as a representative reference and set the decoder-facing budget to $B = 256 \times 70 \times 0.1 = 1{,}792$ visual tokens for both methods, independent of the actual prefix length. As shown in Table 3, KeyRec achieves higher overall performance on both OVO-Bench and StreamingBench across the two backbones, with particularly clear gains on real-time understanding. On OVO-Bench Backward, KeyRec improves E2B and remains close to CausalMem on E4B. Overall, KeyRec maintains its advantage when the 32-frame cap is removed and substantially longer sequences are processed, while keeping persistent storage and per-query decoding cost bounded independently of stream duration.

4 Related Work

Efficient Video Understanding Efficient video understanding initially focused on compressing a fixed video input by exploiting spatial and temporal redundancy (Shao et al., 2026; Ren et al., 2023; Bolya et al., 2022; Huang et al., 2025; Xing et al., 2024; Yang et al., 2025a; Tao et al., 2025). Global methods select representative frames, partition videos into coherent segments, or retain a diverse subset of visual tokens (Tang et al., 2025; Shen et al., 2026; Alvar et al., 2025). More recent approaches make this allocation query-aware, prioritizing frames or regions according to the question (Lin et al., 2026b; Liu et al., 2025; Luo et al., 2026; Zhang et al., 2026b), while trained selectors such as AutoGaze remove redundant patches before ViT processing to reduce both vision-encoder and language-model computation (Shi et al., 2026). These approaches primarily address which visual evidence should be retained from a fixed or query-conditioned video. KeyRec instead treats the retained evidence as a persistent bounded memory: construction remains query-agnostic, while query conditioning is deferred to a fixed-budget readout.

Streaming Video Understanding Streaming video requires causal processing over an observation horizon that may be unknown (Chen et al., 2024; 2025a; Qian et al., 2024; Wang et al., 2026a; Qian et al., 2025; Yang et al., 2026). Training-based systems achieve strong performance through learned response timing, memory organization, and perception–decision coordination, but require model-specific adaptation (Zeng et al., 2026; Ding et al., 2025; Xu et al., 2026; Zhang et al., 2025b). Training-free methods instead operate directly on frozen VLMs and have evolved from reducing redundancy in incoming frames to explicitly managing accumulated history (Guan et al., 2026; Xiong et al., 2025; Xie et al., 2026). At the visual-token level, STC combines reusable vision features with hierarchical token pruning (Wang et al., 2026b), while StreamingTOM selects tokens using adjacent-frame changes and vision-encoder attention (Chen et al., 2026). More memory-oriented approaches explicitly maintain compressed history: FluxMem organizes observations into hierarchical short-, middle-, and long-term memory (Xie et al., 2026), while CausalMem maintains a globally bounded semantic basis through online low-rank updates (Song et al., 2026). KeyRec follows this causal, training-free trajectory but organizes bounded memory differently: query-agnostic writing separates recent evidence from temporally localized events, which are maintained through semantic-temporal add–merge–evict updates. At query time, a fixed readout budget is adaptively allocated between the two. Complementary work instead manages accumulated language-model KV states through query-agnostic or hierarchical memory (Kim et al., 2026b; Di et al., 2025; Yang et al., 2025c; Zhang et al., 2026a), whereas KeyRec operates on model-facing visual embeddings before decoding.

5 Conclusion

KeyRec constructs bounded visual memory through structured historical events, detailed recent evidence, and query-adaptive readout within a training-free framework. Experiments across streaming and long-video benchmarks show that organizing and presenting visual evidence is as important as selecting individually informative tokens, particularly for questions requiring fine-grained recent context. Our evaluation also reveals an underexplored challenge in adapting visual-token compression to emerging encoder-free VLMs. Although several existing methods can be transferred to these models, their effectiveness may change substantially because architecture-specific signals, positional mechanisms, and visual-token interfaces differ from those of conventional encoder–projector VLMs. As encoder-free architectures such as NEO-ov and Gemma 4 12B continue to emerge, understanding how visual memory should adapt to their native representations constitutes a promising and consequential direction for efficient video understanding.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing of this manuscript, including improving clarity, grammar, and presentation. We did not use generative AI to generate datasets or to formulate or prove mathematical claims. All technical content, research claims, experimental results, and conclusions were developed and verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses existing video-understanding benchmarks and pretrained vision-language models and does not involve human-subject studies or new data collection. We do not introduce or release new personal or sensitive data. As with other vision-language systems, downstream applications of the proposed method may inherit biases and limitations from the underlying models and evaluation datasets.

REPRODUCIBILITY STATEMENT

We provide the algorithmic details of KeyRec in Section 2 and the complete experimental setup in Section 3.1. Additional implementation details, including memory configurations, hyperparameters, router allocation profiles, and the full router prompts, are provided in Appendix B.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401. IEEE, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18407–18418. IEEE, 2024.

Joya Chen, Ziyun Zeng, Yiqi Lin, Wei Li, Zejun Ma, and Mike Zheng Shou. Live: Learning video llm with streaming speech transcription at scale. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29083–29095. IEEE, 2025a.

Xueyi Chen, Keda Tao, Kele Shao, and Huan Wang. Streamingtom: Streaming token compression for efficient video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24675–24685, 2026.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. In *International Conference on Learning Representations*, volume 2025, pp. 18227–18246, 2025b.

Seokju Cho, Ryo Hachiuma, Abhishek Badki, Hang Su, Byung-Kwan Lee, Chan Hee Song, Sifei Liu, Subhashree Radhakrishnan, Seungryong Kim, Yu-Chiang Frank Wang, et al. Spatialclaw: Rethinking action interface for agentic spatial reasoning. *arXiv preprint arXiv:2606.13673*, 2026.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video kv-cache retrieval. In *International Conference on Learning Representations*, volume 2025, pp. 42115–42127, 2025.

Haiwen Diao, Jiahao Wang, Penghao Wu, Yuhao Dong, Yuwei Niu, Yue Zhu, Zhongang Cai, Weichen Fan, Linjun Dai, Silei Wu, et al. From pixels to words–towards native one-vision models at scale. *arXiv preprint arXiv:2605.28820*, 2026.

Xin Ding, Hao Wu, Yifan Yang, Shiqi Jiang, Qianxi Zhang, Donglin Bai, Zhibo Chen, and Ting Cao. Streammind: Unlocking full frame rate streaming video dialogue through event-gated cognition. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13448–13459. IEEE, 2025.

Chaoyou Fu, Haozhi Yuan, Yuhao Dong, Yi-Fan Zhang, Yunhang Shen, Xiaoxing Hu, Xueying Li, Jinsen Su, Chengwu Long, Xiaoyao Xie, et al. Video-mme-v2: Towards the next stage in benchmarks for comprehensive video understanding. *arXiv preprint arXiv:2604.05015*, 2026.

Gemma Team. Gemma 4 technical report, 2026. URL `https://arxiv.org/abs/2607.02770`.

Yiran Guan, Liang Yin, Dingkang Liang, Jianzhong Ju, Zhenbo Luo, Jian Luan, Yuliang Liu, and Xiang Bai. Video streaming thinking: Videollms can watch and think simultaneously. *arXiv preprint arXiv:2603.12262*, 2026.

Xiaohu Huang, Hao Zhou, and Kai Han. Prunevid: Visual token pruning for efficient video large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19959–19973, 2025.

Yizhang Jin, Jian Li, Tianjun Gu, Yexin Liu, Bo Zhao, Jinxiang Lai, Zhenye Gan, Yabiao Wang, Chengjie Wang, Xin Tan, et al. Efficient multimodal large language models: A survey. *Visual Intelligence*, 3(1):27, 2025.

Jihwan Kim, Nikhil Parthasarathy, Danfeng Qin, Junhwa Hur, Deqing Sun, Bohyung Han, Ming-Hsuan Yang, and Boqing Gong. Liteframe: Efficient vision encoders unlock frame scaling in video llms. *arXiv preprint arXiv:2605.17260*, 2026a.

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

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 (ICASSP)*, pp. 12147–12151. IEEE, 2026a.

Kuanwei Lin, Wenhao Zhang, and Ge Li. Videorouter: Query-adaptive dual routing for efficient long-video understanding. *arXiv preprint arXiv:2605.05848*, 2026b.

Yudong Liu, Jingwei Sun, Yueqian Lin, Jianyi Zhang, Jingyang Zhang, Ming Yin, Qinsi Wang, Hai Li, and Yiran Chen. Keyframe-oriented vision token pruning: Enhancing efficiency of large vision language models on long-form video processing. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20802–20811. IEEE, 2025.

Yongdong Luo, Wang Chen, Weizhong Huang, Shukang Yin, Haojia Lin, Jinfa Huang, Chaoyou Fu, Jiayi Ji, Xiawu Zheng, and Jiebo Luo. Quota: Query-oriented token assignment via cot query decouple for long video comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24160–24168, 2026.

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. *arXiv preprint arXiv:2505.15269*, 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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18902–18913. IEEE, 2025.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37:119336–119360, 2024.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24045–24055. IEEE, 2025.

Shuhuai Ren, Sishuo Chen, Shicheng Li, Xu Sun, and Lu Hou. Testa: Temporal-spatial token aggregation for long-form video-language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 932–947, 2023.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Holitom: Holistic token merging for fast video large language models. *Advances in Neural Information Processing Systems*, 38:135547–135570, 2026.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, et al. Fastvid: Dynamic density pruning for fast video large language models. *Advances in Neural Information Processing Systems*, 38:123553–123581, 2026.

Baifeng Shi, Stephanie Fu, Long Lian, Hanrong Ye, David Eigen, Aaron Reite, Jan Kautz, Boyi Li, David M Chan, Trevor Darrell, et al. Attend before attention: Efficient and scalable video understanding via autoregressive gazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17022–17034, 2026.

Baiyang Song, Yuli Lin, Qiong Wu, Tao Chen, Jun Peng, Xiao Chen, Yiyi Zhou, and Rongrong Ji. Towards a dynamic and fixed-budget memory bank for efficient streaming video understanding. *arXiv preprint arXiv:2606.25658*, 2026.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29118–29128. IEEE, 2025.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Dycoke: Dynamic compression of tokens for fast video large language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18992–19001. IEEE, 2025.

Haibo Wang, Bo Feng, Zhengfeng Lai, Mingze Xu, Shiyu Li, Weifeng Ge, Afshin Dehghan, Meng Cao, and Ping Huang. Streambridge: Turning your offline video large language model into a proactive streaming assistant. *Advances in Neural Information Processing Systems*, 38:132332–132359, 2026a.

Yiyu Wang, Xuyang Liu, Xiyan Gui, Xinying Lin, Boxue Yang, Chenfei Liao, Tailai Chen, and Linfeng Zhang. Accelerating streaming video large language models via hierarchical token compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18523–18533, 2026b.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 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.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.

Haomiao Xiong, Zongxin Yang, Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Jiawen Zhu, and Huchuan Lu. Streaming video understanding and multi-round interaction with memory-enhanced knowledge. In *International Conference on Learning Representations*, volume 2025, pp. 69332–69351, 2025.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. Streamingvlm: Real-time understanding for infinite video streams. In *International Conference on Learning Representations*, volume 2026, pp. 61463–61475, 2026.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Chendi Li, Jinghua Yan, Yu Bai, Ponnuswamy Sadayappan, Xia Hu, et al. Topv: Compatible token pruning with inference time optimization for fast and low-memory multimodal vision language model. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19803–19813. IEEE, 2025a.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802. IEEE, 2025b.

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*, 2025c.

Zhenyu Yang, Kairui Zhang, Qi Liu, Tiancheng Liu, Long Ying, Dizhan Xue, Qibin Hou, Shengsheng Qian, and Changsheng Xu. Towards online interactors: A comprehensive survey on streaming video understanding. *Preprints*, June 2026.

Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, et al. Streamforest: Efficient online video understanding with persistent event memory. *Advances in Neural Information Processing Systems*, 38:75804–75835, 2026.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025a.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-vstream: Efficient real-time understanding for long video streams. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21059–21069. IEEE, 2025b.

Haowei Zhang, Shudong Yang, Jinlan Fu, See Kiong Ng, and Xipeng Qiu. Hermes: Kv cache as hierarchical memory for efficient streaming video understanding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8411–8430, 2026a.

Kairui Zhang, Zhenyu Yang, Bing Wang, Shengsheng Qian, and Changsheng Xu. Querystream: Advancing streaming video understanding with query-aware pruning and proactive response. In *The Fourteenth International Conference on Learning Representations*, 2026b.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

Default

Selected visual memory segments:
Memory segment 1 (n_1 tokens):
⟨visual token embeddings$_1$⟩
Memory segment 2 (n_2 tokens):
⟨visual token embeddings$_2$⟩
...
Question: ⟨Question q⟩

Naive

<|video|><|video|> ··· <|video|>
Question: ⟨Question q⟩

Detailed

Selected visual memory segments.
Retrieved key event around $\langle \min(\boldsymbol{\tau}_1), \max(\boldsymbol{\tau}_1) \rangle$:
<|video|> ··· <|video|>
Recent context around $\langle T - n, T \rangle$:
<|video|> ··· <|video|>
Question: ⟨Question q⟩

Figure 5: **Memory-serialization templates.** `Default` preserves memory boundaries using neutral segment labels; `Naive` removes these boundaries; and `Detailed` additionally exposes memory types and temporal ranges.

A Additional Experimental Results

A.1 Effect of Memory Serialization

We compare three ways of presenting the same selected visual evidence to the VLM. The `Default` template preserves memory-block boundaries using neutral numbered segments and token counts. The `Naive` template instead concatenates all selected visual tokens into a single sequence, whereas the `Detailed` template exposes both the memory type and its temporal range. All variants use identical selected embeddings, segment order, and readout budget; they differ only in the textual scaffold. Figure 5 shows the three templates.

Table 4 shows that the default template generally achieves the best performance across the evaluated settings. It supports the importance of preserving memory organization during presentation to the language model. The `Naive` variant follows the flat presentation commonly adopted by existing token-selection methods, which directly concatenate all retained visual embeddings before language-model decoding. This removes the boundaries between different events and the recent cache, leaving the language model with an unstructured token sequence and making it difficult to distinguish evidence originating from different temporal contexts. The `Detailed` variant preserves these boundaries but additionally assigns descriptions such as "key event" and "recent context." Such labels introduce semantic priors that may bias how the language model weighs the evidence; for example, the word "key" may cause a retrieved segment to be overemphasized regardless of its relevance to the current question. In contrast, the `Default` template preserves each memory block as a separate segment while using only neutral labels and token counts. These results suggest that the benefit comes from preserving the structural organization of memory while avoiding additional semantic cues that may bias how the language model interprets the retained evidence.

A.2 Sensitivity to Event Granularity

We study how the granularity of each event affects memory quality under a fixed storage budget. Specifically, we vary the per-event visual token size m and adjust the event-bank capacity K in-

Table 4: **Ablation of memory serialization.** All variants use identical selected visual evidence and differ only in their textual scaffold. Bold indicates the best result for each model and benchmark.

Model	Template	OVO Backward	OVO Realtime	LongVideoBench
Gemma 4 E4B	`Naive`	**56.40**	63.72	41.21
	`Detailed`	53.10	63.78	44.43
	`Default`	54.45	**64.81**	**45.77**
NEO-ov 2B	`Naive`	38.78	62.01	48.84
	`Detailed`	39.86	61.90	48.99
	`Default`	**41.90**	**62.78**	**49.29**

Table 5: **Event granularity under fixed storage with Gemma 4 E4B.** We keep the total event capacity Km and the recent/event readout allocation fixed while varying the number and visual token size of stored events.

m	K_{OVO}	OVO Backward	K_{LVB}	LongVideoBench
8	32	53.31	64	44.88
16	16	**54.45**	32	**45.77**
32	8	50.43	16	44.88

versely, keeping $Km = 256$ on OVO-Bench and $Km = 512$ on LongVideoBench. The recent-cache capacity and total readout budget remain unchanged. We also adjust the number of retrieved events so that every variant receives the same recent/event token allocation for each router state. The comparison therefore isolates how the fixed event-memory budget is organized, rather than how many tokens are stored or presented to the language model.

Table 5 shows that the intermediate visual token size $m = 16$ performs best on both benchmarks. With $m = 8$, the memory can retain more events, but each event contains less visual evidence and may provide an incomplete representation of its underlying observation. Increasing the visual token budget to $m = 32$ preserves more detail within each event but substantially reduces the number of distinct events that fit within the same storage budget, particularly harming backward reasoning on OVO-Bench. The default setting therefore provides a favorable balance between event-level evidence completeness and temporal coverage. More broadly, the result shows that KeyRec benefits from organizing its bounded storage into moderately sized evidence units rather than maximizing either the number or the individual resolution of stored events.

A.3 Robustness to Consolidation Thresholds

We evaluate the sensitivity of event consolidation to its semantic and temporal thresholds on OVO-Bench Backward. We vary the route-similarity threshold over $\gamma \in \{0.65, 0.75, 0.85\}$ and the maximum temporal gap over $\delta \in \{1, 2, 4\}$ seconds, while keeping all other settings fixed.

Table 6: **Hyperparameter ablation on OVO-Bench Backward with Gemma 4 E4B.** γ controls route similarity and δ controls temporal locality. † denotes the default configuration used in the main experiments.

$\gamma \backslash \delta$	1 s	2 s	4 s
0.65	54.48	**54.67**	54.27
0.75	54.48	54.45†	53.64
0.85	54.36	53.48	52.47

As shown in Table 6, KeyRec is stable across a broad neighborhood around the default configuration. The default setting is within 0.22 points of the best result, and several nearby threshold combinations obtain comparable accuracy without benchmark-specific tuning. Performance decreases more noticeably when a strict similarity criterion is combined with a wide temporal window. This configuration suppresses the consolidation of moderately changing local observations while still allowing temporally distant near-duplicates to be merged, potentially obscuring the distinction between recurring events. Overall, the results indicate that KeyRec does not depend on a narrowly tuned threshold pair, while also supporting the use of both semantic similarity and temporal locality.

A.4 Router Behavior Across Backbones

Table 7: **Per-task accuracy on the nine OVO-Bench tasks**. Bold and underline indicate the best and second-best compressed methods, respectively; Vanilla is shown as the uncompressed reference. Backward contains EPM, ASI, and HLD; Realtime contains STU, OJR, ATR, ACR, OCR, and FPD.

Model	Method	Backward				Realtime						
		EPM	ASI	HLD	Avg.	STU	OJR	ATR	ACR	OCR	FPD	Avg.
Gemma 4 E2B	Vanilla	41.41	39.86	48.39	43.22	36.52	44.02	58.62	44.95	63.76	56.44	50.72
	STC-Pruner	36.70	35.81	51.08	41.20	41.57	42.93	56.03	35.78	56.38	55.45	48.02
	StreamingTOM-CTR	36.36	40.54	50.00	42.30	35.96	42.39	47.41	43.12	57.05	57.43	47.23
	CausalMem	**40.40**	**41.22**	**54.30**	**45.31**	45.51	52.72	**62.93**	42.20	**71.14**	58.42	55.49
	KeyRec	37.71	40.54	53.23	43.83	**46.63**	**54.89**	62.07	**50.46**	67.79	**64.36**	**57.70**
Gemma 4 E4B	Vanilla	45.12	59.46	56.99	53.86	52.25	52.72	66.38	54.13	71.14	62.38	59.83
	STC-Pruner	37.37	50.68	**70.97**	53.01	43.26	46.20	55.17	48.62	56.38	60.40	51.67
	StreamingTOM-CTR	**41.41**	46.62	63.44	50.49	47.19	49.46	54.31	46.79	62.42	59.41	53.26
	CausalMem	40.74	52.70	69.89	**54.45**	52.81	56.52	**69.83**	58.72	73.83	63.37	62.51
	KeyRec	**41.41**	**54.73**	67.20	**54.45**	**56.74**	**59.78**	67.24	**64.22**	**76.51**	**64.36**	**64.81**
NEO-ov 2B	Vanilla	48.48	54.05	13.44	38.66	44.38	54.35	69.83	51.38	67.11	61.39	58.07
	STC-Pruner	39.73	35.81	**30.11**	35.22	31.46	34.24	41.38	31.19	34.23	48.51	36.84
	StreamingTOM-CTR	–	–	–	–	–	–	–	–	–	–	–
	CausalMem	41.75	45.95	29.57	39.09	34.83	39.13	50.00	36.70	50.34	55.45	44.41
	KeyRec	**48.15**	**50.68**	26.88	**41.90**	**52.25**	**62.50**	**73.28**	**54.13**	**71.14**	**63.37**	**62.78**

We examine how the text-only router allocates memory across backbones. Figure 6 reports the five routing-state distributions on the OVO-Bench Backward and Realtime subsets. The fine-grained distributions are model-dependent: Gemma 4 E2B favors ALLRECENT and BALANCED, Gemma 4 E4B more frequently selects HEAVYRECENT, and NEO-ov 2B assigns more backward questions to HEAVYEVENT.

Despite these differences, a consistent task-dependent shift emerges at the level of memory orientation. For every backbone, realtime questions are routed predominantly toward ALLRECENT or HEAVYRECENT. Backward questions shift away from these recency-oriented states toward balanced or event-oriented allocations. Thus, although the models differ in their preferred allocation granularity, they consistently assign more recent evidence to realtime understanding and more historical evidence to backward reasoning. This behavior supports query-adaptive allocation rather than applying a single recent–event split to all questions.

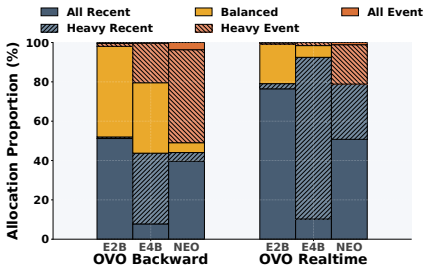


Figure 6: **Router-state distributions across backbones on OVO-Bench.** Each stacked bar shows the percentage of questions assigned to each state. Across backbones, realtime questions favor recent memory, while backward questions shift toward balanced or event-oriented ones.

A.5 PER-TASK AND CATEGORY-LEVEL RESULTS

Tables 7–9 provide finer-grained results underlying the aggregate scores reported in Table 1. On OVO-Bench, KeyRec's gains are most consistent on the real-time tasks, while backward-tracing performance is more mixed across backbones. On StreamingBench, KeyRec achieves the best overall accuracy on all three backbones and performs strongly across most task categories, although some categories such as Cnt remain challenging. On Video-MME-v2, KeyRec is the strongest compressed method across all reported rating dimensions for Gemma 4 E4B and NEO-ov 2B, while the Gemma 4 E2B results are more competitive across methods.

Table 8: **Per-task accuracy on the ten StreamingBench Real-Time tasks**. Bold and underline indicate the best and second-best compressed methods, respectively; Vanilla is shown as the uncompressed reference.

Backbone	Method	CS	OR	AtR	PR	AcR	SU	EU	Cnt	TRU	CR	Overall
Gemma 4 E2B	Vanilla	59.31	54.77	60.00	60.19	41.64	45.53	43.75	43.01	57.32	37.50	51.28
	STC-Pruner	63.41	56.87	56.39	50.93	48.73	42.04	53.75	**41.97**	53.75	54.69	52.91
	StreamingTOM-CTR	62.46	55.31	56.07	40.74	47.88	46.34	51.88	32.12	53.27	53.12	51.36
	CausalMem	67.82	62.67	68.52	53.70	53.82	56.10	57.50	31.09	58.88	53.12	58.01
	KeyRec	**72.24**	**65.94**	**74.43**	**65.74**	**56.66**	**57.72**	**64.38**	17.62	**64.80**	**58.59**	**61.29**
Gemma 4 E4B	Vanilla	71.92	65.67	68.20	67.59	59.49	54.47	60.62	54.92	71.03	52.34	63.73
	STC-Pruner	65.30	57.69	57.70	**71.30**	51.84	46.12	61.25	41.45	58.75	58.59	56.44
	StreamingTOM-CTR	70.03	58.31	64.59	60.19	54.67	55.69	65.62	**51.30**	60.75	60.94	60.25
	CausalMem	71.61	66.49	72.46	70.37	58.92	61.38	65.62	37.31	65.11	59.38	63.61
	KeyRec	**76.66**	**68.94**	**78.36**	68.52	**60.91**	**66.26**	**66.25**	40.41	**65.73**	**61.72**	**66.49**
NEO-ov 2B	Vanilla	70.03	76.29	77.70	74.07	66.01	59.35	68.12	52.85	67.60	77.34	69.06
	STC-Pruner	52.05	49.32	53.44	59.26	52.12	46.75	52.50	**32.64**	52.65	60.94	50.68
	StreamingTOM-CTR	–	–	–	–	–	–	–	–	–	–	–
	CausalMem	59.62	56.68	60.33	57.41	54.67	52.85	65.00	29.53	57.32	60.16	55.56
	KeyRec	**74.76**	**75.20**	**81.97**	**64.81**	**68.56**	**68.29**	**70.00**	31.61	**68.54**	**70.31**	**69.10**

Table 9: **Video-MME-v2 official ratings and unnormalized accuracy**. Bold and underline indicate the best and second-best compressed methods, respectively; Vanilla is shown as the uncompressed reference.

Backbone	Method	Unnorm. Acc.	Level 1	Level 2	Level 3	Relevance	Rel.-Linear	Logic	Total
Gemma 4 E2B	Vanilla	18.47	10.19	6.01	4.84	7.36	17.29	5.09	6.56
	STC-Pruner	14.16	4.69	4.03	3.54	4.35	12.96	3.32	3.99
	StreamingTOM-CTR	**15.69**	**6.73**	**5.77**	3.93	**5.92**	**14.93**	3.89	**5.21**
	CausalMem	15.31	5.37	4.51	**4.49**	5.24	13.82	3.76	4.72
	KeyRec	15.41	5.69	4.82	4.35	5.24	13.73	**4.10**	4.84
Gemma 4 E4B	Vanilla	20.09	10.44	7.56	6.03	8.25	18.55	6.47	7.62
	STC-Pruner	14.34	5.29	5.47	3.79	5.31	13.15	3.55	4.69
	StreamingTOM-CTR	15.91	6.75	5.93	4.56	6.05	14.55	4.61	5.54
	CausalMem	14.84	5.02	5.53	3.91	5.13	13.01	3.89	4.70
	KeyRec	**17.63**	**6.97**	**6.05**	**5.19**	**6.19**	**15.41**	**5.39**	**5.91**
NEO-ov 2B	Vanilla	22.06	13.18	8.18	6.55	9.85	21.00	6.68	8.74
	STC-Pruner	17.56	7.57	5.98	4.59	7.50	17.97	2.59	5.78
	StreamingTOM-CTR	–	–	–	–	–	–	–	–
	CausalMem	17.47	6.52	5.79	4.63	6.54	16.71	3.49	5.47
	KeyRec	**20.03**	**10.30**	**6.68**	**4.83**	**7.86**	**19.22**	**4.81**	**6.79**

B Additional Implementation Details

We use two fixed operating profiles: OVO-Bench and StreamingBench share the *streaming* profile, while LongVideoBench and Video-MME-v2 share the *long-video* profile. No configuration is tuned separately for individual benchmarks.

Memory configuration. The streaming and long-video profiles process at most $F = 32$ and $F = 64$ frames, respectively. Given n model-facing visual tokens per frame, we set the decoder-facing budget to $B = 0.1Fn$ and the recent-cache capacity to $N_C = B$. Gemma produces $n = 70$ tokens per frame and uses $m = 16$ tokens per event, whereas NEO-ov produces approximately $n = 210$ tokens per frame and uses $m = 48$. The event-bank capacity is $K = 16$ for streaming and $K = 32$ for long-video evaluation. Thus, for example, the Gemma streaming profile stores at most 224 recent tokens and 256 event tokens, totaling $480/2240 \approx 21\%$ of the dense visual input, while only the 10% readout is exposed to the VLM for each query.

Table 10 reports the remaining event-memory parameters. We use temporal-coverage weighting for streaming video, where the observation horizon is unknown, and disable it for offline long-video evaluation, where frames are sampled across the complete known horizon. This regime-specific temporal weighting follows the same principle as CausalMem (Song et al., 2026). Each profile is shared across all benchmarks and backbones.

Table 10: **Event-memory hyperparameters.** The parameters are shared across settings and backbones. The two values of λ correspond to the streaming and long-video profiles, respectively.

Parameter	Value	Role
γ	0.75	Route-similarity threshold
δ	2.0 s	Maximum temporal merge gap
β	0.2	Route update weight
λ	0.2/0.0	Temporal-coverage weight

Router allocation profiles. Table 11 reports the effective event-memory allocations induced by Eq. 11. Because complete events are retrieved, the allocation is discretized into $B/m = 14$ event blocks for the streaming profile and $B/m = 28$ for the long-video profile. The same codebook is used across backbones within each profile. Figure 7 provides the exact text-only router prompts.

Baseline configurations. All baselines use the same frozen backbone, sampled frames, benchmark prompts, and answer-extraction procedure as KeyRec. STC-Pruner uses its Gaussian token-scoring variant. StreamingTOM-CTR uses a temporal-similarity threshold of 0.9, DPC neighborhood size 7, and merge weight 0.6; we evaluate only its causal temporal-reduction component and exclude post-VLM KV-cache quantization. Because it requires attention signals from a dedicated vision encoder, StreamingTOM-CTR is evaluated only on Gemma. CausalMem uses at most 64 basis vectors, an activity-decay factor of 0.9, and at most eight new basis vectors per frame; following its original protocol, its temporal weight is 0.8 for streaming evaluation and disabled for long-video evaluation. All compressed baselines use a 10% decoder-facing retention ratio.

Table 11: **Router allocation codebooks.** Each entry reports the effective fraction of the readout assigned to event memory and the corresponding number of complete events k_q in parentheses. The same profile is used across backbones.

Router state	Streaming Fraction (k_q)	Long video Fraction (k_q)
AllRecent	0 (0)	0 (0)
HeavyRecent	1/7 (2)	2/7 (8)
Balanced	2/7 (4)	1/2 (14)
HeavyEvent	6/7 (12)	6/7 (24)
AllEvent	1 (14)	1 (28)

Streaming Video Memory Router Prompt

```
You are a memory-allocation router for a streaming video
understanding system.
The model can read two kinds of visual memory:
- recent: the latest visual tokens before the question time, useful
for current-state and fine visual detail.
- key_event: selected past visual event tokens, useful for
long-range evidence and temporal coverage.
Choose exactly one category for the question:
- all_recent: the answer should mostly depend on the latest/current
visual details.
- heavy_recent: mostly current-state or fine-detail question, but a
little past context may help.
- balanced: both recent context and selected past events are
useful.
- heavy_key_event: the question likely requires searching across
earlier events or broad temporal coverage.
- all_key_event: the question is mainly about historical events and
recent visual context is unlikely to help.
Return only JSON in this format:
{"category": "<one category>", "reason": "<short reason>"}
Question: ⟨Question q⟩
```

Long Video Memory Router Prompt

```
You are a memory-allocation router for offline whole-video question
answering.
The complete video has already been processed before the question is
asked. The model can read two kinds of compressed visual memory:
- recent: visual tokens from only the ending portion of the video.
They do not represent the current moment at query time.
- key_event: selected visual events distributed across the full
video timeline.
Choose exactly one category according to where and how broadly the
answer evidence is likely to occur:
- all_recent: use only when the question explicitly asks about the
final scene, ending, or final state.
- heavy_recent: the evidence is likely concentrated near the end,
but some earlier context may help.
- balanced: the evidence is localized but its temporal position is
uncertain, or both ending and earlier context may matter.
- heavy_key_event: the question requires searching, comparing,
ordering, counting, or summarizing events across the video.
- all_key_event: broad historical coverage is essential and fine
detail from the ending is unlikely to help.
Return only JSON in this format:
{"category": "<one category>", "reason": "<short reason>"}
Question: ⟨Question q⟩
```

Figure 7: **Text-only router prompts.** The streaming router memory prompt interprets recent memory as evidence immediately preceding the query, whereas the offline long video memory router prompt treats it as evidence from the ending portion of the complete video.

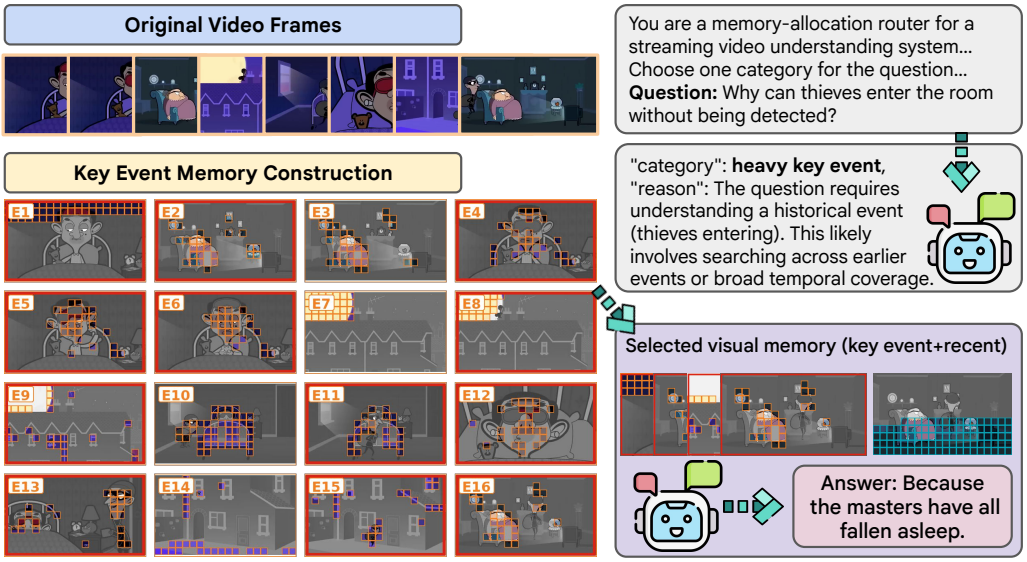


Figure 8: **KeyRec case study on StreamingBench with NEO-ov 2B.** KeyRec constructs query-agnostic key events from selected visual patches, routes the history-dependent question to coverage-oriented retrieval, and produces a fixed-budget readout that supports the correct answer. The red border indicates the selected events after determining the budget allocation by the router.

C Qualitative Case Study

Figure 8 traces how KeyRec processes an example with NEO-ov 2B. During query-agnostic memory construction, KeyRec retains the visually informative patches highlighted by the orange boxes and organizes them into key events. Because NEO-ov has no separate ViT encoder, these selected patch embeddings are *exactly* the visual evidence available to its unified backbone at query time, rather than an attribution visualization over additional hidden image features. The resulting events preserve sufficient information about the characters and their states before the question is observed.

The question, *"Why can thieves enter the room without being detected?"*, appears in a streaming setting but requires evidence distributed across the preceding history rather than only the latest frame. The router accordingly assigns it to the coverage-oriented heavy-event state. KeyRec then orders the event bank by event-center time and selects events uniformly over the observed history. The bank contains visually similar pairs, such as E2–E3 and E7–E8, which remain separate because their temporal gaps exceed the merge threshold δ. Nevertheless, the fixed-budget readout includes E2 and E8 while excluding E3 and E7, thereby covering both portions of the history without presenting both members of either redundant pair. Together with the recent contribution, this evidence enables NEO-ov to correctly infer that the thieves remain undetected because the masters have fallen asleep.

D Limitations

KeyRec is training-free and relies on fixed novelty, consolidation, and retention rules; learning these policies from token-level supervision or downstream QA signals may further improve memory construction and routing. In our Gemma implementation, KeyRec operates after the vision encoder, reducing visual-memory and language-model costs but not ViT encoding latency Kim et al. (2026a). Preliminary intermediate-token reduction did not improve wall-clock encoding time because it disrupted the optimized encoder path, leaving efficient integration of encoder-side pruning with post-encoder memory as an open systems challenge. Finally, the event memory stores selected visual patches without explicitly modeling object identities, trajectories, or state transitions, limiting precise reasoning about duration, order, and cross-frame evolution. Object-centric or hierarchical event representations may better preserve such temporal dynamics under a bounded memory budget.