# OBJECTSTREAM: LATENT OBJECTS AS MEMORY ANCHORS FOR STREAMING VIDEO UNDERSTANDING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Streaming video understanding requires models to continuously retain useful visual evidence before future questions are known. Existing approaches primarily manage the growing visual context according to token importance, temporal redundancy, or segment-level relevance, but rarely organize evidence around objects that persist and evolve over time. Thus, in this paper, we introduce **ObjectStream**, a training-free framework that treats ***latent objects as memory anchors*** for streaming video understanding. ObjectStream induces spatially coherent latent objects directly from frozen Video-LLM representations, links them across frames into persistent anchors, and maintains their histories under a bounded memory budget, without requiring external object detectors or segmentation models. Built on these anchors, ObjectStream preserves three complementary forms of evidence: *persistent object histories*, *transient object changes*, and *recent visual context*. This design enables existing Video Large Language Models (Video-LLMs) to reason over object identities, interactions, and state changes while leaving the underlying model unchanged. Extensive experiments on online streaming and offline long-video benchmarks demonstrate both effectiveness and efficiency. In online streaming evaluation, ObjectStream improves Qwen2.5-VL-7B by 10.0 points on OVO-Bench Real-Time Visual Perception, while reducing peak GPU memory and TTFT by approximately 50%. On offline long-video benchmarks, it surpasses the full-token baseline while discarding 82.5% of visual tokens. These results highlight latent objects as a practical and effective organizing principle for compact streaming video memory.

## 1 INTRODUCTION

Streaming video understanding Maaz et al. (2024); Zhang et al. (2025b); Bai et al. (2025); Wang et al. (2025) requires models to process continuously arriving visual inputs and respond to questions that may be issued at any time. This capability is essential for applications such as live monitoring Dumitru & Spînu (2026), autonomous systems Li et al. (2026a); Brödermann et al. (2024); Zhao et al. (2025), and egocentric vision and wearable assistant Fu et al. (2019); Plizzari et al. (2024); Zhang et al. (2026a); Zhu et al. (2026); Xu et al. (2025); Zhang et al. (2025a); Li et al. (2026b); Fu et al. (2025c); Pan et al. (2025), and embodied agents and robotic systems Wang et al. (2023); Zou et al. (2026); Du et al. (2026); Wang et al. (2026b;a); Lin et al. (2026b; 2025); Wang et al. (2026c); Lin et al. (2026a). Unlike offline video understanding, where the complete video can be revisited after a question is given, streaming models must decide what information to retain before future queries are known, while operating under strict memory and latency constraints.

Recent studies Wu et al. (2026); Liang et al. (2026); Zhang et al. (2026b); Guan et al. (2026) have extended Video Large Language Models (Video-LLMs) to streaming scenarios from various perspectives, including online adaptation Shen et al. (2024); Lu et al. (2026), streaming-oriented inference Zhang et al. (2024); Xu et al. (2026), and memory-based context management Xie et al. (2026); Liang et al. (2026); Zhang et al. (2026b); Ge et al. (2026); Wu et al. (2026). Among them, training-free approaches Xie et al. (2026); Chen et al. (2026); Liang et al. (2026) are particularly attractive because they can adapt existing offline Video-LLMs without additional training or modification to the underlying model. These methods typically control the growing visual context through frame selection Shen et al. (2026), visual-token pruning Xie et al. (2026), KV-cache compression Di et al. (2025); Chen et al. (2025a), and temporal or segment-level memory organization Liang et al.



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.

(2026); Zhang et al. (2026b). By reducing redundant historical information and prioritizing potentially relevant evidence over long video streams, they improve memory efficiency and response latency, facilitating the transition of Video-LLMs from offline video understanding to practical online streaming.

However, reducing the number of retained frames or tokens addresses only part of the streaming-memory problem; the retained evidence must also be organized for later retrieval and reasoning. Existing training-free methods primarily structure memory around frames, tokens, cached states, or temporal segments, rather than persistent object identities. As a result, evidence associated with the same object may remain fragmented across time, making questions about object identity, interaction, and state evolution particularly difficult. This motivates organizing streaming visual memory around persistent object-level anchors.

Based on this insight, we introduce **ObjectStream**, a training-free and plug-and-play visual memory framework that treats ***latent objects as memory anchors*** for streaming Video-LLMs. Rather than managing historical visual tokens or KV states in isolation, ObjectStream organizes the incoming stream through a *Latent Object-Anchored Memory*. Specifically, it induces spatially coherent object candidates directly from frozen Video-LLM representations, associates them across frames, and maintains their evolving histories under a bounded memory budget. Since these object-like units emerge entirely from the model's latent feature space, without requiring external object detectors or segmentation models, we refer to them as *latent objects*. The resulting cross-frame object states provide a compact semantic index over the stream, enabling the model to preserve object-related evidence beyond locally salient visual content. To complement persistent object histories, ObjectStream introduces two additional components. *Object-Conditioned Temporal Residuals* preserve short-lived but important evidence around abrupt object-level changes, while the *Recent Visual Grounding Window* retains the latest observations for timely visual grounding. Together, these components capture three complementary forms of memory: *persistent object histories*, *transient object changes*, and *recent visual context*, while leaving the underlying Video-LLM unchanged.

We evaluate ObjectStream on online streaming and offline long-video benchmarks. With Qwen2.5-VL-7B, it improves OVO-Bench real-time visual perception, OVO-Bench backward tracing, and StreamingBench by 10.0, 4.9, and 2.9 points, respectively. It also improves over the corresponding full-token backbone on the evaluated offline benchmarks while retaining only 17.5% of the visual tokens. These results show that object-centric memory can improve video understanding while substantially reducing the visual context.

Our main contributions are summarized as follows:

- We propose ObjectStream, a training-free and plug-and-play visual memory framework for efficient streaming video understanding, without architectural modifications.
- We introduce *Latent Object-Anchored Memory*, using latent objects as persistent anchors to organize long-term visual evidence without external detectors or segmenters.
- We complement persistent object memory with *Object-Conditioned Temporal Residuals* and a *Recent Visual Grounding Window*, achieving strong performance and token efficiency across online streaming and offline video understanding benchmarks.

## 2 Related Work

**Video Large Language Models.** Many Video Large Language Models (Video-LLMs) have been proposed to address video-language alignment, temporal modeling, and long-context reasoning. Representative proprietary models include Gemini 1.5 Pro Team & Vinyals (2024) and GPT-4o OpenAI: et al. (2024), while widely used open-source models include LLaVA-Video Zhang et al. (2025b), Qwen2-VL Wang et al. (2024), InternVL2 Chen et al. (2025b), and LongVU Shen et al. (2024). Despite their strong performance, these models primarily follow an offline inference paradigm, assuming that the complete video is available before question answering. This assumption limits their direct applicability to the more realistic and challenging streaming setting, where visual inputs arrive continuously and future observations are unavailable.

**Streaming Video-LLMs and Visual Memory.** Existing streaming video understanding models explore either training-based adaptation or training-free memory management. Training-based meth-

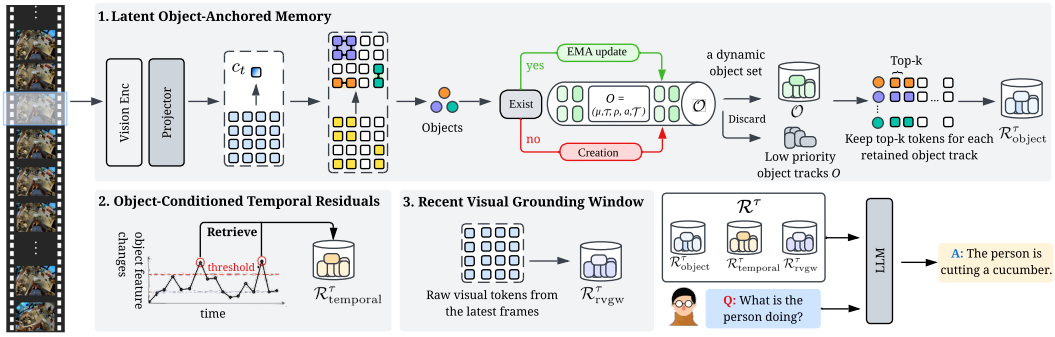



Figure 1: Overview of ObjectStream. It comprises three memory modules: *Latent Object-Anchored Memory* discovers and tracks latent object anchors and stores their persistent evidence; *Object-Conditioned Temporal Residuals* preserve transient changes; and the *Recent Visual Grounding Window* retains recent observations for immediate grounding.

ods include VideoLLM-Online Chen et al. (2024), which introduces online video instruction tuning for continuous video perception and interaction; Dispider Qian et al. (2025), which enables active real-time interaction by disentangling perception, decision, and reaction; and ThinkStream Liu et al. (2026), which adopts a Watch–Think–Speak paradigm and learns compressed semantic memory for long-horizon streaming. ViSpeak Fu et al. (2025b) and StreamForest Zeng et al. (2025) further improve online video understanding through streaming-oriented datasets, training objectives, and model adaptations. These methods generally require additional optimization or model-specific modifications.

Training-free approaches instead preserve the underlying Video-LLM and manage historical visual information directly at inference time. StreamChat Liu et al. (2025) maintains external visual memory to incorporate previous observations, while ReKV and StreamKV Di et al. (2025); Chen et al. (2025a) reduce memory overhead through KV-cache retention and compression. More recent methods explore structured memory management: FluxMem Xie et al. (2026) introduces hierarchical visual memory to balance long-term history and recent observations; QueryStream Zhang et al. (2026c) performs query-aware token pruning to retain request-relevant evidence; and OASIS Liang et al. (2026) organizes streaming history into event-level structures for on-demand retrieval. In contrast, these methods primarily organize historical information around frames, tokens, cached states, or events rather than persistent latent object anchors. ObjectStream complements these designs by explicitly associating object-like latent regions across frames and using the resulting tracks to organize retained visual evidence.

## 3 METHOD

**Overview.** Our design follows three core objectives: maintaining persistent object histories, preserving transient object-level changes, and retaining recent visual context for real-time grounding. The overall framework of our proposed **ObjectStream** is illustrated in Fig. 1 and consists of three components. First, *Latent Object-Anchored Memory* discovers spatially coherent latent object candidates from frozen Video-LLM representations, associates them across frames, and maintains their evolving histories under a bounded memory budget. Second, *Object-Conditioned Temporal Residuals* preserve short-lived evidence around abrupt object-level changes. Third, the *Recent Visual Grounding Window* retains the latest observations for timely question answering.

Formally, let $\mathcal{V}_\tau = \{I_t\}_{t=1}^{\tau}$ denote the video stream observed up to the current timestamp $\tau$. ObjectStream processes each frame incrementally upon arrival. At timestamp $t \le \tau$, the newly observed frame $I_t$ is encoded into a set of patch-level visual tokens: $\mathcal{X}_t = \{x_{t,j}\}_{j=1}^{N}$, where $N$ is the number of visual tokens extracted from frame $I_t$, and each token $x_{t,j} \in \mathbb{R}^d$ is a $d$-dimensional visual representation. After processing all observations up to timestamp $\tau$, ObjectStream maintains the retained

visual memory as:

$$\mathcal{R}^{\tau} = \mathcal{R}^{\tau}_{\text{object}} \cup \mathcal{R}^{\tau}_{\text{temporal}} \cup \mathcal{R}^{\tau}_{\text{rvgw}}, \tag{1}$$

where $\mathcal{R}^{\tau}_{\text{object}}$, $\mathcal{R}^{\tau}_{\text{temporal}}$, and $\mathcal{R}^{\tau}_{\text{rvgw}}$ are produced by the aforementioned modules, retaining features associated with persistent object histories, transient object-level changes, and recent visual context, respectively.

### 3.1 Latent Object-Anchored Memory

To use latent objects as memory anchors for object-centric evidence retention, this module comprises four stages, denoted as *S1–S4*: *S1: Query-Agnostic Token Saliency Estimation*, *S2: Spatial Clustering for Latent Object Anchor Discovery*, *S3: Cross-Frame Association and Bounded Object-Set Maintenance* and *S4: Object-Centric Evidence Retention*. We describe each stage below.

***S1: Query-Agnostic Token Saliency Estimation.*** For each frame $I_t$, we first summarize its overall visual content using a frame-level context vector $c_t$. Specifically, we $\ell_2$-normalize each visual token and aggregate the normalized tokens into a global context representation :

$$\begin{aligned} \hat{x}_{t,j} &= \frac{x_{t,j}}{\|x_{t,j}\|_2}, \\ c_t &= \frac{\sum_{j=1}^{N} \hat{x}_{t,j}}{\left\| \sum_{j=1}^{N} \hat{x}_{t,j} \right\|_2}. \end{aligned} \tag{2}$$

The context vector $c_t$ captures the dominant visual semantics of frame $I_t$. We then estimate the saliency of each token by its cosine dissimilarity from this frame-level context:

$$s_{t,j} = 1 - \cos(\hat{x}_{t,j}, c_t) = 1 - \hat{x}_{t,j}^{\top} c_t. \tag{3}$$

The score $s_{t,j}$ measures how visually distinctive token $x_{t,j}$ is within the current frame. Tokens that deviate more from the global frame representation are more likely to correspond to informative objects, persons, or regions. We then select active token candidates using a per-frame quantile threshold:

$$\mathcal{A}_t = \left\{ j \mid s_{t,j} \geq \text{Quantile}_q\left(\mathcal{S}_t\right) \right\}, \tag{4}$$

where $\mathcal{S}_t = \{s_{t,j}\}_{j=1}^{N}$ is the set of saliency scores for all visual tokens in frame $t$, and $\text{Quantile}_q(\mathcal{S}_t)$ denotes the $q$-th quantile of saliency scores within frame $t$; thus larger $q$ selects fewer, more salient tokens as active object candidates.

***S2: Spatial Clustering for Latent Object Anchor Discovery.*** As shown in Fig. **??**, the active tokens indexed by $\mathcal{A}_t$ are still patch-level candidates and may be spatially fragmented. To form coherent object-level regions, ObjectStream exploits the 2D patch-grid structure of visual tokens. Let $p_{t,j}$ denote the spatial position of token $x_{t,j}$. We build an undirected graph $\mathcal{G}_t$ over active tokens, where edges connect spatially adjacent tokens:

$$\begin{aligned} \mathcal{G}_t &= (\mathcal{A}_t, \mathcal{E}_t), \\ (j,k) \in \mathcal{E}_t &\iff j,k \in \mathcal{A}_t \wedge p_{t,k} \in \mathcal{N}(p_{t,j}), \\ \mathcal{C}_t &= \text{CC}(\mathcal{G}_t). \end{aligned} \tag{5}$$

Here, $\mathcal{E}_t$ is the edge set, $\mathcal{N}_4(p_{t,j})$ denotes the 4-connected spatial neighborhood of patch $p_{t,j}$ on the patch grid, including its upper, lower, left, and right neighbors. $\text{CC}(\cdot)$ denotes connected components, and $\mathcal{C}_t = \{C_{t,m}\}_{m=1}^{M_t}$ is the set of frame-level object candidates.

$$\mu = f(C_{t,m}) = \frac{\sum_{j \in C_{t,m}} s_{t,j} x_{t,j}}{\left\| \sum_{j \in C_{t,m}} s_{t,j} x_{t,j} \right\|_2}. \tag{6}$$

This transforms scattered salient patch tokens into coarse object-level units without requiring external detectors or segmentation annotations.

***S3: Cross-Frame Association and Bounded Object-Set Maintenance.*** After obtaining frame-level object candidates, ObjectStream associates them across frames to build persistent object tracks. We

maintain a dynamic object set $\mathcal{O} = \{O_i\}$, where each object track $O_i$ stores an object feature $\mu_i$, the last observed timestamp $\tau_i$, a temporal consistency score $\rho_i$, an object saliency score $a_i$, and a historical token set $\mathcal{T}_i$. The temporal consistency score $\rho_i$ is computed as the cosine similarity between the track feature and the candidate feature at its most recent match, indicating the confidence of cross-frame association. Upon each successful match, the temporal consistency score $\rho_i$ is updated as the cosine similarity. The object saliency score $a_i$ is inherited from the token-level saliency introduced in S2 by running averaging the saliency values of the visual tokens belonging to the corresponding object candidate.

For each frame-level candidate $C_{t,m}$, ObjectStream first finds the most similar existing object track:

$$i^\star = \arg\max_i \cos\left(f(C_{t,m}), \mu_i\right). \tag{7}$$

If $\cos(f(C_{t,m}), \mu_{i^*}^{t-1}) \geq \theta$, the candidate is assigned to $O_{i^\star}$ and the object feature is updated by Exponential Moving Average (EMA):

$$\mu_{i^*}^t = \frac{\beta\mu_{i^*}^{t-1} + (1-\beta)f(C_{t,m})}{\left\|\beta\mu_{i^*}^{t-1} + (1-\beta)f(C_{t,m})\right\|_2}. \tag{8}$$

If no existing object satisfies the matching threshold, ObjectStream initializes a new object track. For a matched object, we set $\tau_{i^*}$ to the current timestamp, update its temporal consistency $\rho_{i^*}$ and average saliency $a_{i^*}$, and insert all tokens in the matched component into $\mathcal{T}_{i^*}$.

Since streaming videos continuously introduce new objects, the object set must remain bounded. When the number of maintained tracks exceeds the budget $M$, ObjectStream evicts low-priority objects according to recency, saliency, and temporal consistency.

$$\begin{aligned} \mathrm{rank}(O_i) &= (\tau_i,\ a_i, \rho_i), \\ \mathcal{O} &= \text{Top-M}_{O_i \in \mathcal{O}}\left(\mathrm{rank}(O_i)\right). \end{aligned} \tag{9}$$

The ranking is applied in lexicographic order, prioritizing object anchors that are recently observed, visually salient, and temporally consistent.

***S4: Object-Centric Evidence Retention.*** Once persistent object tracks in the set $\mathcal{O}$ are maintained under a fixed budget, ObjectStream constructs the long-term object-centric evidence $\mathcal{R}^\tau_{\text{object}}$ by selecting representative tokens from each retained object anchor.

Object eviction and token retention operate at different granularities: the bounded object set $\mathcal{O}$ determines which object tracks are maintained up to timestamp $\tau$, while token retention is performed over the historical token pool $\mathcal{T}_i^\tau$ of each retained track $O_i \in \mathcal{O}$. For each retained object $O_i$, its historical token pool contains token indices $(t, j)$ assigned to this object up to timestamp $\tau$. Instead of globally selecting top-scoring tokens from the whole video, which may be dominated by a few salient regions or frames, ObjectStream allocates token retention capacity at the object level:

$$\begin{aligned} \mathcal{I}_i^\tau &= \text{Top-K}_{(t,j)\in\mathcal{T}_i^\tau,\ t\leq\tau}\left(s_{t,j}, k_i\right), \\ \mathcal{R}^\tau_{\text{object}} &= \bigcup_{O_i\in\mathcal{O}} \left\{x_{t,j} \mid (t,j) \in \mathcal{I}_i^\tau\right\}. \end{aligned} \tag{10}$$

The operator Top-K ranks tokens within the object-specific pool $\mathcal{T}_i^\tau$ by their saliency scores $s_{t,j}$ and returns the top-$k_i$ token indices. The retained object memory $\mathcal{R}^\tau_{\text{object}}$ then collects the corresponding visual tokens from all maintained object tracks, forming a compact yet comprehensive representation of persistent visual histories.

### 3.2 Object-Conditioned Temporal Residuals

Latent Object-Anchored memory preserves persistent visual semantics by smoothing object representations over time. However, this smoothing may suppress short-lived but important changes, such as motion, action transitions, object appearance or disappearance, state changes, and interaction changes. Thus, this module is further proposed to detect abrupt object-level feature changes and preserve additional raw tokens around the corresponding timestamps.

For each object track $O_i$, we denote its matched sequence of historical observations as $\{z_{i,l}\}_{l=1}^{L_i}$. Specifically, if the frame-level object candidate $f(C_{t,m})$ is successfully assigned to track $O_i$ at a timestamp $t$, this observation is mapped as $z_{i,l} = f(C_{t,m})$, where $l$ indexes the chronologically ordered observation sequence up to length $L_i$, where $2 \leq l \leq L_i$, and $t_{i,l} = t$ preserves its corresponding timestamp. We then compute the feature change between neighboring observations and use an object-specific adaptive threshold to detect temporal events:

$$\begin{aligned} \Delta_{i,l} &= 1 - \cos(z_{i,l}, z_{i,l-1}), \\ \gamma_{i,l} &= \text{mean}\left(\{\Delta_{i,k}\}_{k=2}^{l}\right) + \text{std}\left(\{\Delta_{i,k}\}_{k=2}^{l}\right), \\ \mathcal{B}^{\tau} &= \{(i, t_{i,l}) \mid \Delta_{i,l} > \gamma_i,\ t_{i,l} \leq \tau\}. \end{aligned} \tag{11}$$

Here, $\Delta_{i,l}$ measures the temporal change of object $O_i$ between two consecutive matched observations, $\gamma_{i,l}$ is its adaptive temporal threshold, and $\mathcal{B}^{\tau}$ denotes the temporal events detected up to timestamp $\tau$.

For each temporal event $(i,t) \in \mathcal{B}^{\tau}$, ObjectStream retrieves the original visual tokens assigned to the same object within a local temporal window $\mathcal{W}(t,r) = \{u \mid t - r \leq u \leq t + r,\ u \leq \tau\}$, where $r$ denotes the temporal window frame offset that defines the local neighborhood around event timestamp $t$. Within this object-specific temporal neighborhood, we select token indices according to their saliency scores:

$$\begin{aligned} \mathcal{I}_{i,t}^{\tau} &= \text{TopK}_{(u,j)\in\mathcal{T}_i^{\tau},\ u\in\mathcal{W}(t,r)}\left(s_{u,j}, k_b\right), \\ \mathcal{R}_{\text{temporal}}^{\tau} &= \bigcup_{(i,t)\in\mathcal{B}^{\tau}} \left\{x_{u,j} \mid (u,j) \in \mathcal{I}_{i,t}^{\tau}\right\}. \end{aligned} \tag{12}$$

In Eq.12 Top-K returns the indices of the top-$k_b$ tokens ranked by saliency within the local temporal neighborhood, and $\mathcal{R}_{\text{temporal}}^{\tau}$ stores the corresponding raw visual tokens.

Unlike object-level representative tokens, temporal residual tokens preserve raw visual evidence around rapid object-level changes. Therefore, they complement the smooth long-term object memory and improve the ability to reason about short-term temporal events.

### 3.3 Recent Visual Grounding Window

Although object memory and temporal residuals preserve long-term semantics and short-term dynamics, streaming questions may also require the latest visual state. For example, questions about what is currently happening or which object is now visible depend strongly on the most recent frames. To support such real-time questions, ObjectStream maintains a lightweight *Recent Visual Grounding Window* (rvgw), which directly preserves raw visual tokens from the latest $L$ frames:

$$\mathcal{R}_{\text{rvgw}}^{\tau} = \bigcup_{t=\max(1,\tau-L+1)}^{\tau} \mathcal{X}_t. \tag{13}$$

The rvgw complements the compressed object memory by providing high-fidelity local evidence for the current visual state, while the object and temporal residual memories preserve compact historical context.

***Inference.*** Given a textual question $Q$, the language model generates the answer by conditioning on the retained visual memory:

$$\hat{Y} = \text{LLM}(Q, \mathcal{R}^{\tau}). \tag{14}$$

By integrating persistent object histories, transient object-level changes, and recent visual context, ObjectStream provides the underlying Video-LLM with a compact yet expressive representation of the observed stream. This unified memory supports both historical reasoning and grounding in the current visual state under a bounded visual-token budget.

## 4 Experiments

**Implementation Details.** For all experiments, we build our method on Qwen2.5-VL-3B and 7B Bai et al. (2025). For online benchmarks, we sample videos at 1 fps and set the maximum video length to

| Method | Frames | OVO-Bench Real-Time | | | | | | | BT | StreamingBench | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | OCR | ACR | ATR | STU | FPD | OJR | Avg. | Avg. | OP | CR | CS | ATP | EU | TR | PR | SU | ACP | CT | Avg. |
| *Proprietary Models* | | | | | | | | | | | | | | | | | | | | |
| Gemini 1.5 Pro | 1 fps | 85.9 | 67.0 | 79.3 | 58.4 | 63.4 | 62.0 | 69.3 | 62.5 | 79.0 | 80.5 | 83.5 | 79.7 | 80.0 | 84.7 | 77.8 | 64.2 | 72.0 | 48.7 | 75.7 |
| GPT-4o | 64 | 69.8 | 64.2 | 71.6 | 51.1 | 70.3 | 59.8 | 64.5 | 60.8 | 77.1 | 80.5 | 83.9 | 76.5 | 70.2 | 83.8 | 66.7 | 62.2 | 69.1 | 49.2 | 73.3 |
| *Open-source Offline MLLMs* | | | | | | | | | | | | | | | | | | | | |
| LLaVA-Video | 64 | 69.8 | 59.6 | 66.4 | 50.6 | 72.3 | 61.4 | 63.3 | 41.7 | – | – | – | – | – | – | – | – | – | – | – |
| Qwen2-VL | 64 | 69.1 | 53.2 | 63.8 | 50.6 | 66.3 | 60.9 | 60.7 | 48.6 | – | – | – | – | – | – | – | – | – | – | – |
| InternVL2 | 64 | 68.5 | 58.7 | 69.0 | 44.9 | 67.3 | 56.0 | 60.7 | 44.0 | 68.1 | 60.9 | 69.4 | 77.1 | 67.7 | 62.9 | 59.3 | 53.3 | 55.0 | 56.5 | 63.7 |
| LongVU | 1 fps | 55.7 | 49.5 | 59.5 | 48.3 | 68.3 | 63.0 | 57.4 | 39.5 | – | – | – | – | – | – | – | – | – | – | – |
| *Open-source Online MLLMs (Training-Based)* | | | | | | | | | | | | | | | | | | | | |
| VideoLLM-Online-8B | 2 fps | 8.1 | 23.9 | 12.1 | 14.0 | 45.5 | 21.2 | 20.8 | 17.7 | 39.1 | 40.1 | 34.5 | 31.1 | 46.0 | 32.4 | 31.5 | 34.2 | 42.5 | 27.9 | 36.0 |
| Dispider-7B | 1 fps | 57.7 | 49.5 | 62.1 | 44.9 | 61.4 | 51.6 | 54.6 | 36.1 | 74.9 | 75.5 | 74.1 | 73.1 | 74.4 | 59.9 | 76.1 | 62.9 | 62.2 | 45.8 | 67.6 |
| Flash-VStream-7B | 1 fps | 25.5 | 32.1 | 29.3 | 33.7 | 29.7 | 28.8 | 29.9 | 25.4 | 25.9 | 43.6 | 24.9 | 23.9 | 27.3 | 13.1 | 18.5 | 25.2 | 23.9 | 48.7 | 23.2 |
| ViSpeak | 1 fps | 75.2 | 58.7 | 71.6 | 51.1 | 74.3 | 66.9 | 66.3 | **57.5** | 79.8 | **88.3** | 83.3 | 81.1 | 76.4 | 75.1 | 70.4 | 65.9 | **77.3** | 34.2 | 74.4 |
| ThinkStream | 1 fps | 85.2 | 64.2 | 69.8 | 49.4 | 69.3 | 64.1 | 67.0 | 52.3 | – | – | – | – | – | – | – | – | – | – | – |
| TimeChat-Online-7B | 1 fps | 75.2 | 46.8 | 70.7 | 47.8 | 69.3 | 61.4 | 61.9 | 41.7 | 80.8 | 79.7 | 80.8 | 83.3 | 74.8 | 78.8 | 78.7 | 64.2 | 68.8 | **58.0** | 75.3 |
| StreamForest-7B | 1 fps | 68.5 | 53.2 | 71.6 | 47.8 | 65.4 | 60.9 | 61.2 | 52.0 | **83.1** | 82.8 | 82.7 | 84.3 | 77.5 | 78.2 | 76.9 | 69.1 | 75.6 | 54.4 | **77.3** |
| *Open-source Online MLLMs (Training-Free)* | | | | | | | | | | | | | | | | | | | | |
| Qwen2.5-VL-3B† | 1 fps | 76.5 | 44.0 | 67.2 | 42.1 | 68.3 | 62.0 | 60.0 | 42.0 | 76.2 | 68.8 | 75.4 | 79.2 | 73.0 | 72.3 | 71.3 | 61.4 | 71.6 | 26.1 | 68.0 |
| + FluxMem | 1 fps | 83.2 | 56.9 | 67.2 | 47.8 | 68.3 | 63.6 | 64.5 | 42.1 | 72.5 | 73.8 | 73.8 | 79.4 | 72.6 | 79.7 | 76.9 | 64.2 | 65.1 | 42.6 | 70.9 |
| + **ObjectStream (Ours)** | 1 fps | 87.3 | 68.8 | 73.3 | 54.5 | 66.3 | 65.2 | 68.6 | 42.5 | 72.7 | 69.8 | 79.8 | 80.4 | 73.9 | 82.8 | 65.7 | 71.1 | 72.0 | 34.6 | 72.4 |
| Qwen2.5-VL-7B† | 1 fps | 79.2 | 53.2 | 67.2 | 51.7 | 71.3 | 57.1 | 63.3 | 44.6 | 78.3 | 80.5 | 79.8 | 82.4 | 75.5 | 80.4 | 74.1 | 62.6 | 67.6 | 51.1 | 73.9 |
| + FluxMem | 1 fps | 81.2 | 59.6 | 70.7 | 53.4 | 75.2 | 63.0 | 67.2 | 46.8 | 80.2 | 81.1 | 81.4 | **85.3** | **78.0** | 83.8 | 80.6 | 65.9 | 69.6 | 52.1 | 76.4 |
| + QueryStream | 1 fps | 75.2 | 49.5 | 69.8 | 50.0 | 71.3 | 62.5 | 63.1 | 44.9 | 82.4 | 84.4 | 79.2 | 82.4 | **78.0** | 81.3 | 78.7 | 65.0 | 69.3 | 47.3 | 75.3 |
| + OASIS | – | 85.2 | **72.5** | 66.4 | 52.3 | 67.3 | 64.7 | 64.7 | 52.6 | – | – | – | – | – | – | – | – | – | – | 70.6 |
| + **ObjectStream (Ours)** | 1 fps | **89.9** | 70.6 | **74.1** | **59.0** | **75.3** | **70.7** | **73.3** | 49.5 | 74.1 | 80.2 | **88.6** | 82.6 | 77.1 | **85.0** | **82.4** | **73.6** | 73.2 | 44.2 | 76.8 |

Table 1: Performance comparison on OVO-Bench and StreamingBench. For OVO-Bench, we report real-time visual perception scores and the average backward tracing score. For StreamingBench, we report real-time category scores and the average score. **Bold** indicates the best result, and underlining indicates the second-best result. † indicates the backbone model.

| Model | #Frames | VideoMME-L | EgoSchema |
|---|---|---|---|
| Gemini 1.5 Pro (P) | 1 fps | 62.5 | 69.3 |
| GPT-4o (P) | 64 | 60.7 | 64.5 |
| LLaVA-Video-7B (O) | 32 | – | 57.3 |
| Qwen2-VL-7B (O) | 64 | – | **66.7** |
| LongVU-7B (O) | 1 fps | **60.6** | 58.2 |
| TimeChat-Online-7B (On) | 1 fps | 48.4 | 61.9 |
| Dispider-7B (On) | 1 fps | 49.7 | 55.6 |
| Vista (On) | 1 fps | – | 58.7 |
| Qwen2.5-VL-3B† | 1 fps | 49.3 | 54.9 |
| + FluxMem | 1 fps | 51.2 | 57.8 |
| + **ObjectStream** | 1 fps | 52.9 | 57.7 |
| Qwen2.5-VL-7B† | 1 fps | 51.3 | 58.5 |
| + FluxMem | 1 fps | 53.9 | 60.1 |
| + QueryStream | 1 fps | 52.9 | – |
| + **ObjectStream** | 1 fps | 54.0 | 60.8 |

Table 2: Performance comparison (%) on offline benchmarks. P: proprietary; O: open-source offline; On: open-source online. † denotes the backbone model.

256 frames. We use 8 frames as *the Recent Visual Grounding Window* (rvgw) for real-time question answering and compress older frames into object-centric memory. We keep up to 64 objects with 8 tokens per object, and further enable temporal residuals with at most 32 events, radius 1, and extra 2 tokens per selected frame. For offline benchmarks, we limit the maximum sequence length to 1024 frames. We use a larger object memory budget to preserve longer-range temporal context, keeping up to 512 objects with 80 tokens per object. We preserve the most recent 80 frames as the rvgw, while compressing older visual content into object-centric long-term memory. Details are in Appendix. 7.4

**Benchmarks.** We evaluate our method on both online and offline benchmarks. For online evaluation, we use StreamingBench Lin et al. (2024) and OVO-Bench Li et al. (2025). StreamingBench evaluates streaming video question answering under causal access, requiring the model to answer questions incrementally as video frames arrive over time. OVO-Bench focuses on online video understanding with an emphasis on real-time reasoning, covering abilities such as object perception, temporal reasoning, causal understanding, and event-based decision making under constrained memory. For offline evaluation, we use EgoSchema Mangalam et al. (2023) and VideoMME-Long Fu et al. (2025a). These benchmarks evaluate holistic video understanding when the entire video is available, with a particular focus on long-range temporal reasoning, egocentric understanding, and comprehensive multi-step video QA. More details are in Appendix. 7.3

**Results on Online Benchmark.** We compare ObjectStream with strong proprietary models, open-source offline Video-LLMs, training-based online Video-LLMs, and training-free online methods, including Gemini 1.5 Pro Team & Vinyals (2024), GPT-4o OpenAI: et al. (2024), LLaVA-Video Zhang et al. (2025b), Qwen2-VL Wang et al. (2024), InternVL2 Chen et al. (2025b), LongVU Shen et al. (2024), VideoLLM-Online Chen et al. (2024), Dispider Qian et al. (2025), Flash-VStream Zhang et al. (2024), ViSpeak Fu et al. (2025b), ThinkStream Liu et al. (2026), TimeChat-Online Yao et al. (2025), StreamForest Zeng et al. (2025), QueryStream Zhang et al. (2026c), OASIS Liang et al. (2026) and FluxMem Xie et al. (2026). Details are in Appendix. 7.2

**Main Results on Streaming Benchmarks.** As shown in Tab. 1, ObjectStream consistently improves the unmodified Qwen2.5-VL backbones across the evaluated streaming benchmarks without additional training or architectural modification. With Qwen2.5-VL-7B, it increases the OVO-Bench real-time visual perception score from 63.3 to 73.3 and the backward tracing score from 44.6 to 49.5, yielding gains of 10.0 and 4.9 points, respectively. ObjectStream also outperforms FluxMem on the same backbone, improving real-time visual perception from 67.2 to 73.3 and backward tracing from 46.8 to 49.5. Similar gains are observed with Qwen2.5-VL-3B, where ObjectStream improves the corresponding scores over FluxMem from 64.5 to 68.6 and from 42.1 to 42.5. These results show that object-centric memory organization provides consistent benefits across both backbone scales, particularly for real-time perception and retrieval of historical visual evidence. On StreamingBench, ObjectStream improves the average score from 73.9 to 76.8 with the 7B backbone and from 68.0 to 72.4 with the 3B backbone. These results demonstrate that latent object memory construction provides a general and effective solution for online perception and temporal reasoning.

**Extension to Offline Long-video Benchmarks.** We also evaluate ObjectStream on offline long-video benchmarks to verify whether the proposed memory design remains effective in the offline scenario. As shown in Tab. 2, ObjectStream consistently improves the original Qwen2.5-VL baselines on both VideoMME-Long and EgoSchema. With Qwen2.5-VL-7B, our method improves VideoMME-Long from 51.3 to 54.0 and EgoSchema from 58.5 to 60.8, achieving gains of 2.7 and 2.3 points, respectively. With Qwen2.5-VL-3B, ObjectStream also improves VideoMME-Long from 49.3 to 52.9 and EgoSchema from 54.9 to 57.7, showing that the benefit is consistent across model scales. These results demonstrate that ObjectStream is not limited to online streaming inference.

### 4.1 Ablation Studies

Table 3: Component ablation of ObjectStream.

| Component | | | VME | Ego | OVO | Streaming |
|---|---|---|---|---|---|---|
| RVGW | LOAM | OCTR | Long | Schema | Bench | Bench |
| ✓ | | | 52.3 | 58.6 | 62.3 | 73.9 |
| | ✓ | | 51.1 | 58.3 | 62.0 | 74.8 |
| ✓ | ✓ | | 53.4 | 59.4 | 60.6 | 76.5 |
| | ✓ | ✓ | 53.8 | 60.4 | 62.1 | 74.9 |
| ✓ | ✓ | ✓ | **54.0** | **60.8** | **65.4** | **76.8** |

**Effect of Memory Components.** To evaluate the contributions of Recent Visual Grounding Window (RVGW), Latent Object-Anchored Memory (LOAM), and Object-Conditioned Temporal Residuals (OCTR), we progressively enable each module. As shown in Tab. 3, RVGW mainly improves StreamingBench by preserving recent visual observations, while LOAM enhances OVO-Bench through structured object memory. Combining the two consistently improves performance

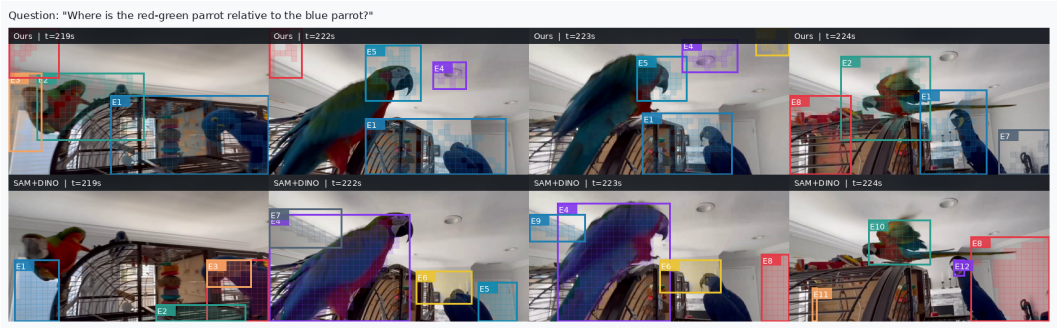



Figure 2: Qualitative comparison of latent object construction. SAM+DINOv2 produces finer segmentation masks, whereas ObjectStream constructs coarser object-like regions directly from frozen backbone features without an external segmentation model.

across both streaming and offline settings. Adding OCTR further preserves abrupt object changes, resulting in the best overall performance.

### 4.2 Object Memory Construction and Efficiency Analysis.

We compare different object-memory construction strategies using the same Qwen2.5-VL-7B backbone on OVO-Bench. As shown in Table 5, Patch Top-K is memory-efficient but incurs high TTFT, while DINOv2- and SAM-based alternatives introduce additional feature extraction or segmentation overhead. ObjectStream instead constructs latent objects directly from backbone features, achieving 73.3 performance with 19.89 GB peak memory and 6.697 s TTFT. These results demonstrate a favorable trade-off among performance, memory usage, and latency.Fig. 2 further illustrates this difference qualitatively. Although SAM+DINOv2 produces finer object masks, its outputs are optimized for pixel-level segmentation rather than temporally reusable memory for VLM reasoning. ObjectStream produces coarser but sufficiently localized regions and maintains task-relevant objects across consecutive observations despite changes in scale, pose, and viewpoint. These results suggest that ObjectStream provides a lightweight and temporally grounded object construction mechanism better suited to the low-latency and memory-bounded requirements of streaming video understanding. Detailed quantitative and qualitative comparisons are provided in Appendix 7.5.

## 5 Conclusion

To conclude, in this paper, we propose ObjectStream, a training-free object-centric memory mechanism for efficient streaming video understanding. By discovering object-latent anchors and linking them into persistent tracks, ObjectStream builds a compact semantic object related memory over incoming video streams. This design preserves persistent visual evidence while remaining sensitive to dynamic changes and the latest visual state, enabling effective reasoning under memory and latency constraints. Experiments on both streaming and offline long-video benchmarks show that ObjectStream substantially improves online and offline video understanding while maintaining strong robustness under limited token budgets.

## 6 Limitations

Our current evaluation focuses primarily on downstream video understanding performance and does not directly measure the long-term accuracy of the latent object tracks. Consequently, tracking errors such as object fragmentation, identity switches, and incorrect merging are not independently quantified. Moreover, our experiments are limited to the evaluated VLM backbone, and the generalization of ObjectStream to newer or architecturally distinct VLMs remains to be established. Future work will develop dedicated long-term tracking diagnostics and evaluate ObjectStream across a broader range of VLM architectures.

### AI USE STATEMENT

In this work, we used generative AI tools for language polishing and formatting assistance. We have not used generative AI tools for generating research ideas, conducting experiments, analyzing results, or drawing scientific conclusions; other disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work and manually verified the revised text and formatting to ensure accuracy and consistency with the authors' intended meaning. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

We provide sufficient details to facilitate reproducibility of our results. The main paper describes the overall architecture and key components of ObjectStream, while the appendix provides additional implementation details, hyperparameter settings, benchmark configurations, and evaluation protocols. We also document the preprocessing and inference procedures used for all evaluated benchmarks, together with the configurations of the compared baselines. Additional qualitative results and ablation studies are included in the supplementary material.

## 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, 2025. URL https://arxiv.org/abs/2502.13923.

Tim Brödermann, Christos Sakaridis, Yuqian Fu, and Luc Van Gool. Cafuser: Condition-aware multimodal fusion for robust semantic perception of driving scenes. *arXiv preprint arXiv:2410.10791*, 2024.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large language model for streaming video, 2024. URL https://arxiv.org/abs/2406.11816.

Xueyi Chen, Keda Tao, Kele Shao, and Huan Wang. Streamingtom: Streaming token compression for efficient video understanding, 2026. URL https://arxiv.org/abs/2510.18269.

Yilong Chen, Xiang Bai, Zhibin Wang, Chengyu Bai, Yuhan Dai, Ming Lu, and Shanghang Zhang. Streamkv: Streaming video question-answering with segment-based kv cache retrieval and compression, 2025a. URL https://arxiv.org/abs/2511.07278.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yiming Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling, 2025b. URL https://arxiv.org/abs/2412.05271.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Tao Zhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video kv-cache retrieval, 2025. URL https://arxiv.org/abs/2503.00540.

Yuxin Du, Tao Lin, Zile Zhong, Runting Li, Xiyao Chen, Jiting Liu, Chenglin Liu, Ying-Cong Chen, Yuqian Fu, and Bo Zhao. Focusable monocular depth estimation, 2026. URL https://arxiv.org/abs/2605.11756.

Estera Dumitru and Stelian Spînu. A multi-task deep learning framework for real-time intelligent video surveillance with temporal event validation, 2026. URL https://arxiv.org/abs/2607.03131.

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, 2025a. URL https://arxiv.org/abs/2405.21075.

Shenghao Fu, Qize Yang, Yuan-Ming Li, Yi-Xing Peng, Kun-Yu Lin, Xihan Wei, Jian-Fang Hu, Xiaohua Xie, and Wei-Shi Zheng. Vispeak: Visual instruction feedback in streaming videos, 2025b. URL https://arxiv.org/abs/2503.12769.

Yuqian Fu, Chengrong Wang, Yanwei Fu, Yu-Xiong Wang, Cong Bai, Xiangyang Xue, and Yu-Gang Jiang. Embodied one-shot video recognition: Learning from actions of a virtual embodied agent. In *Proceedings of the 27th ACM international conference on multimedia*, 2019.

Yuqian Fu, Runze Wang, Bin Ren, Guolei Sun, Biao Gong, Yanwei Fu, Danda Pani Paudel, Xuanjing Huang, and Luc Van Gool. Objectrelator: Enabling cross-view object relation understanding across ego-centric and exo-centric perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025c.

Haonan Ge, Yiwei Wang, Hang Wu, and Yujun Cai. What should a streaming video model remember?, 2026. URL https://arxiv.org/abs/2606.16353.

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, 2026. URL https://arxiv.org/abs/2603.12262.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. URL https://arxiv.org/abs/2304.02643.

Feng Li, Chaokun Zhang, and Gong Chen. Sparse-aware vector quantization for bandwidth-efficient collaborative 3d semantic occupancy prediction, 2026a. URL https://arxiv.org/abs/2607.01928.

Yanjun Li, Yuqian Fu, Tianwen Qian, Qi'ao Xu, Silong Dai, Danda Pani Paudel, Luc Van Gool, and Xiaoling Wang. Egocross: Benchmarking multimodal large language models for cross-domain egocentric video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026b.

Yifei Li, Junbo Niu, 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?, 2025. URL https://arxiv.org/abs/2501.05510.

Zhijia Liang, Jiaming Li, Weikai Chen, Yanhao Zhang, Haonan Lu, and Guanbin Li. Oasis: On-demand hierarchical event memory for streaming video reasoning, 2026. URL https://arxiv.org/abs/2604.17052.

Junming Lin, Zheng Fang, Chi Chen, Zihao Wan, Fuwen Luo, Peng Li, Yang Liu, and Maosong Sun. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding, 2024. URL https://arxiv.org/abs/2411.03628.

Tao Lin, Yilei Zhong, Yuxin Du, Jingjing Zhang, Jiting Liu, Yinxinyu Chen, Encheng Gu, Ziyan Liu, Hongyi Cai, Yanwen Zou, et al. Evo-1: Lightweight vision-language-action model with preserved semantic alignment. *arXiv preprint arXiv:2511.04555*, 2025.

Tao Lin, Yuxin Du, Jiting Liu, Nuobei Zhu, Yunhe Li, Yuqian Fu, Yinxinyu Chen, Hongyi Cai, Zewei Ye, Bing Cheng, et al. Evo-depth: A lightweight depth-enhanced vision-language-action model. *arXiv preprint arXiv:2605.14950*, 2026a.

Tao Lin, Yuxin Du, Yiran Mao, Zewei Ye, Yilei Zhong, Bing Cheng, Yiming Wang, Jiting Liu, Yang Tian, Junchi Yan, et al. La4vla: Learning to act without seeing via language-action pretraining. *arXiv preprint arXiv:2606.27295*, 2026b.

Jihao Liu, Zhiding Yu, Shiyi Lan, Shihao Wang, Rongyao Fang, Jan Kautz, Hongsheng Li, and Jose M. Alvare. Streamchat: Chatting with streaming video, 2025. URL https://arxiv.org/abs/2412.08646.

Zikang Liu, Longteng Guo, Handong Li, Ru Zhen, Xingjian He, Ruyi Ji, Xiaoming Ren, Yanhao Zhang, Haonan Lu, and Jing Liu. Thinking in streaming video, 2026. URL https://arxiv.org/abs/2603.12938.

Xudong Lu, Yang Bo, Jinpeng Chen, Shuhan Li, Xintong Guo, Huankang Guan, Fang Liu, Dunyuan Xu, Peiwen Sun, Heyang Sun, Rui Liu, and Hongsheng Li. Aura: Always-on understanding and real-time assistance via video streams, 2026. URL https://arxiv.org/abs/2604.04184.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models, 2024. URL https://arxiv.org/abs/2306.05424.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding, 2023. URL https://arxiv.org/abs/2308.09126.

OpenAI:, Yunxing Dai, and Yury Malkov. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

Jiancheng Pan, Runze Wang, Tianwen Qian, Mohammad Mahdi, Yanwei Fu, Xiangyang Xue, Xiaomeng Huang, Luc Van Gool, Danda Pani Paudel, and Yuqian Fu. V2-sam: Marrying sam2 with multi-prompt experts for cross-view object correspondence. *arXiv preprint arXiv:2511.20886*, 2025.

Chiara Plizzari, Gabriele Goletto, Antonino Furnari, Siddhant Bansal, Francesco Ragusa, Giovanni Maria Farinella, Dima Damen, and Tatiana Tommasi. An outlook into the future of egocentric vision, 2024. URL https://arxiv.org/abs/2308.07123.

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, 2025. URL https://arxiv.org/abs/2501.03218.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. Longvu: Spatiotemporal adaptive compression for long video-language understanding, 2024. URL https://arxiv.org/abs/2410.17434.

Yujiao Shen, Shulin Tian, Jingkang Yang, and Ziwei Liu. A simple baseline for streaming video understanding, 2026. URL https://arxiv.org/abs/2604.02317.

Gemini Team and Petko Georgiev Oriol Vinyals. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL https://arxiv.org/abs/2403.05530.

Kuanning Wang, Ke Fan, Yuqian Fu, Siyu Lin, Hu Luo, Daniel Seita, Yanwei Fu, Yu-Gang Jiang, and Xiangyang Xue. Ocra: Object-centric learning with 3d and tactile priors for human-to-robot action transfer. *arXiv preprint arXiv:2603.14401*, 2026a.

Kuanning Wang, Ke Fan, Chenhao Qiu, Zeyu Shangguan, Yuqian Fu, Yanwei Fu, Daniel Seita, and Xiangyang Xue. Oflow: Injecting object-aware temporal flow matching for robust robotic manipulation. *arXiv preprint arXiv:2604.17876*, 2026b.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024. URL https://arxiv.org/abs/2409.12191.

Runze Wang, Yuqian Fu, Yu Li, Tao Lin, Tianwen Qian, Mohamed Elhoseiny, Bo Zhao, Yanwei Fu, Yu-Gang Jiang, and Xiangyang Xue. Afford-vla: Action-aligned visual planning via internalized affordance. *arXiv preprint arXiv:2605.24203*, 2026c.

Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, Neel Joshi, and Marc Pollefeys. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world, 2023. URL https://arxiv.org/abs/2309.17024.

Yifei Wang, Zhenkai Li, Tianwen Qian, Huanran Zheng, Zheng Wang, Yuqian Fu, and Xiaoling Wang. Streameqa: Towards streaming video understanding for embodied scenarios. *arXiv preprint arXiv:2512.04451*, 2025.

Hang Wu, Sherin Mary Mathews, Yujun Cai, Ming-Hsuan Yang, and Yiwei Wang. Semantic-aware adaptive visual memory for streaming video understanding, 2026. URL https://arxiv.org/abs/2605.07897.

Yiweng Xie, Bo He, Junke Wang, Xiangyu Zheng, Ziyi Ye, and Zuxuan Wu. Fluxmem: Adaptive hierarchical memory for streaming video understanding, 2026. URL https://arxiv.org/abs/2603.02096.

Qi'ao Xu, Tianwen Qian, Yuqian Fu, Kailing Li, Yang Jiao, Jiacheng Zhang, Xiaoling Wang, and Liang He. Tog-bench: Task-oriented spatio-temporal grounding in egocentric videos. *arXiv preprint arXiv:2512.03666*, 2025.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Yao Lu, and Song Han. Streamingvlm: Real-time understanding for infinite video streams, 2026. URL https://arxiv.org/abs/2510.09608.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, Lingpeng Kong, Qi Liu, Yuanxing Zhang, and Xu Sun. Timechat-online: 80URL https://arxiv.org/abs/2504.17343.

Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, Yi Wang, and Limin Wang. Streamforest: Efficient online video understanding with persistent event memory, 2025. URL https://arxiv.org/abs/2509.24871.

Deheng Zhang, Yuqian Fu, Runyi Yang, Yang Miao, Tianwen Qian, Xu Zheng, Guolei Sun, Ajad Chhatkuli, Xuanjing Huang, Yu-Gang Jiang, et al. Egonight: Towards egocentric vision understanding at night with a challenging benchmark. *arXiv preprint arXiv:2510.06218*, 2025a.

Ganlin Zhang, Deheng Zhang, Longteng Duan, Guo Han, Yuqian Fu, Danda Pani Paudel, Luc Van Gool, and Eric Vollenweider. Egospot:egocentric multimodal control for hands-free mobile manipulation, 2026a. URL https://arxiv.org/abs/2306.02393.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-vstream: Memory-based real-time understanding for long video streams, 2024. URL https://arxiv.org/abs/2406.08085.

Haowei Zhang, Shudong Yang, Jinlan Fu, See-Kiong Ng, and Xipeng Qiu. Hermes: Kv cache as hierarchical memory for efficient streaming video understanding, 2026b. URL https://arxiv.org/abs/2601.14724.

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*, 2026c. URL https://openreview.net/forum?id=738HjJEbml.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data, 2025b. URL https://arxiv.org/abs/2410.02713.

Jingyuan Zhao, Yuyan Wu, Rui Deng, Susu Xu, Jinpeng Gao, and Andrew Burke. A survey of autonomous driving from a deep learning perspective. *ACM Computing Surveys*, 2025.

Bingwen Zhu, Yuqian Fu, Qiaole Dong, Guolei Sun, Tianwen Qian, Yuzheng Wu, Danda Pani Paudel, Xiangyang Xue, and Yanwei Fu. Egosound: Benchmarking sound understanding in egocentric videos. *arXiv preprint arXiv:2602.14122*, 2026.

Zhishan Zou, Guoyan Sun, Zhiwei Wei, Jiancheng Pan, Yujie Li, Mugen Peng, and Wenjia Xu. Self in space: Benchmarking self-awareness and spatial cognition in uav embodied intelligence, 2026. URL https://arxiv.org/abs/2607.12477.

# 7 Appendix

## 7.1 Backbone Models

ObjectStream is built on top of Qwen2.5-VL, a widely used open-source Video-LLM, evaluated at two parameter scales: Qwen2.5-VL-3B and Qwen2.5-VL-7B (Bai et al., 2025; Wang et al., 2024). Both are used as frozen backbones. ObjectStream does not modify their weights or architecture in any way; it only manages which visual tokens are exposed to the language model at inference time. This training-free design allows the same memory mechanism to be applied to either model scale without retraining.

## 7.2 Baseline Descriptions

We compare ObjectStream with representative video-language models and streaming-memory methods. The baselines are grouped into four categories: proprietary models, open-source offline Video-LLMs, training-based online Video-LLMs, and training-free visual-memory methods.

**Proprietary and offline Video-LLMs.** Gemini 1.5 Pro (Team & Vinyals, 2024) and GPT-4o (OpenAI: et al., 2024) are included as proprietary reference models. For open-source offline Video-LLMs, we consider LLaVA-Video (Zhang et al., 2025b), Qwen2-VL (Wang et al., 2024), InternVL2 (Chen et al., 2025b), and LongVU (Shen et al., 2024). These models process a completed video clip and are not specifically trained for causal streaming inference. Their results provide a reference for the performance gap between offline video understanding and online streaming video understanding.

**Training-based online Video-LLMs.** We compare with VideoLLM-Online (Chen et al., 2024), Dispider (Qian et al., 2025), Flash-VStream (Zhang et al., 2024), ViSpeak (Fu et al., 2025b), ThinkStream (Liu et al., 2026), TimeChat-Online (Yao et al., 2025), and StreamForest (Zeng et al., 2025). These methods are designed for streaming video understanding through online instruction tuning, streaming-specific objectives, persistent event memory, or active perception. In contrast, ObjectStream does not require additional training or architectural modification.

**Training-free visual-memory methods.** Qwen2.5-VL-3B and Qwen2.5-VL-7B are used as backbone-only baselines without any additional visual-memory module. FluxMem (Xie et al., 2026) is a training-free hierarchical visual-memory method that separates recent, intermediate, and long-term visual information. We also include QueryStream (Zhang et al., 2026c) and OASIS (Liang et al., 2026), which perform query-aware token selection and event-level memory organization, respectively. These methods are compared under their corresponding published settings. The dagger symbol denotes the unmodified Qwen2.5-VL backbone.

**Object-construction baselines.** To isolate the effect of our latent object construction strategy, we further compare three alternative strategies under the same Qwen2.5-VL-7B backbone and approximately the same visual-token budget. Patch Top-K directly retains the most salient patch tokens

without constructing object tracks. DINOv2 (Oquab et al., 2024) Clustering uses external DINOv2 features to form spatial clusters, whose locations are then mapped back to the visual tokens of the backbone. SAM+DINOv2 (Kirillov et al., 2023) first obtains region masks with SAM and then uses DINOv2 features for region representation and cross-frame association. These external frontends introduce additional computation and memory overhead. In contrast, ObjectStream constructs latent object candidates directly in the frozen backbone feature space without external detectors or segmentation models.

### 7.3 BENCHMARKS

**Online streaming benchmarks** (used to evaluate real-time streaming video understanding under causal access, i.e., the model can only use information observed up to the current timestamp):

OVO-Bench (Li et al., 2025): focuses on online video understanding, evaluating abilities such as object perception, temporal reasoning, causal understanding, and event-based decision making under constrained memory. The paper reports Real-Time Visual Perception scores (broken down into sub-categories: OCR, ACR, ATR, STU, FPD, OJR) and Backward Tracing (BT) scores.

StreamingBench (Lin et al., 2024): evaluates streaming video question answering under causal access, requiring the model to answer questions incrementally as frames arrive. It covers multiple real-time categories (OP, CR, CS, ATP, EU, TR, PR, SU, ACP, CT).

**Offline long-video benchmarks** (used to evaluate whether the memory design remains effective when the entire video is available before the question is asked, and to test token efficiency on long videos):

VideoMME-Long (Fu et al., 2025a): the long-video subset (¿30 min) of Video-MME, a comprehensive multi-modal LLM video evaluation benchmark.

EgoSchema (Mangalam et al., 2023): a diagnostic benchmark for very long-form egocentric video-language understanding, focused on egocentric reasoning over multi-minute clips.

### 7.4 FULL HYPERPARAMETER SETTINGS

Table 4 lists the complete set of hyperparameters used across the online and offline evaluation settings. Unless otherwise noted, all values were selected via a light grid search on a held-out validation split of StreamingBench and kept fixed across all other benchmarks to demonstrate that ObjectStream does not require per-benchmark tuning.

Table 4: Hyperparameter settings for online and offline evaluation.

| Symbol | Description | Online | Offline |
|---|---|---|---|
| – | Frame sampling rate | 1 fps | 1 fps |
| – | Max stream length | 256 frames | 1024 frames |
| $L$ | RVGW window size | 8 frames | 80 frames |
| $M$ | Max object tracks | 64 | 512 |
| $k_i$ | Tokens retained / object | 8 | 80 |
| $q$ | Saliency quantile (Eq. 4) | 0.6 | 0.6 |
| $\theta$ | Association threshold (Eq. 7–8) | 0.75 | 0.75 |
| $\beta$ | EMA decay (Eq. 8) | 0.9 | 0.9 |
| – | Max temporal residual events | 32 | 64 |
| $r$ | Temporal window radius | 1 frame | 1 frame |
| $k_b$ | Tokens / temporal event | 2 | 4 |

### 7.5 OBJECT MEMORY CONSTRUCTION AND EFFICIENCY ANALYSIS.

We further analyze different object memory construction strategies under the same Qwen2.5-VL-7B backbone on OVO-Bench. As shown in Tab. 5, Patch Top-K directly selects salient patch tokens without explicit object modeling. Although memory-efficient, it yields lower performance and much

Table 5: Comparison of object-memory construction strategies on OVO-Bench using Qwen2.5-VL-7B. We report peak GPU memory, time to first token (TTFT), time per output token (TPOT), and real-time visual perception performance.

| Variant | Peak Mem↓ | TTFT↓ | TPOT↓ | Perf↑ |
|---|---|---|---|---|
| *Reference Baselines* | | | | |
| Qwen2.5-VL-7B | 40.50GB | 13.431s | 0.336s | 63.3 |
| + FluxMem | 25.78GB | **6.641s** | 0.252s | 67.2 |
| *Object Construction Variants* | | | | |
| Patch Top-K | **19.89GB** | 16.223s | 0.320s | 72.8 |
| DINOv2 Cluster | 23.09GB | 18.120s | 1.230s | 72.3 |
| SAM+DINOv2 | 26.78GB | 19.060s | 0.785s | **73.8** |
| **Latent Objects (Ours)** | **19.89GB** | 6.697s | **0.240s** | 73.3 |

higher TTFT, showing that unstructured token selection is insufficient for streaming visual memory. DINOv2 Clustering introduces external patch features but incurs additional memory and latency overhead while still underperforming our method. SAM+DINOv2 achieves the highest accuracy using segmentation-based regions, yet requires expensive segmentation and feature extraction, resulting in substantially higher latency and memory cost. In contrast, ObjectStream discovers objects directly in the backbone visual token space and combines them with temporal retention, achieving 73.3 performance with only 19.89 GB peak memory and 6.697s TTFT.

### 7.6 Pseudo-code

Algorithm 1 summarizes the streaming procedure of ObjectStream.

### 7.7 Analysis of retained token ratio

As shown in Fig. 3, we analyze the effect of the retained visual token ratio on VideoMME-Long. The Qwen2.5-VL-3B baseline keeps 100% visual tokens and achieves a performance of 49.3. However, With only 5% retained tokens, ObjectStream obtains 48.1, slightly below the dense baseline. Once the keep ratio increases to 7.5%, ObjectStream surpasses the baseline and continues to improve, reaching the best performance of 52.9 at 17.5%. The performance remains strong at 20%, with a score of 52.3. These results indicate that ObjectStream does not rely on retaining dense visual tokens; instead, a compact set of object-centric tokens is sufficient to preserve useful long-video evidence when an appropriate memory budget is used.

### 7.8 Analysis of Saliency Quantile and Threshold

We also analyze the sensitivity of ObjectStream to two key hyper-parameters: the saliency quantile $q$ for object candidate selection and the matching threshold $\theta$ for cross-frame object association. As shown in Fig. 4, ObjectStream remains stable across a wide range of settings. When varying the saliency quantile from 0.4 to 0.8, the OVO real-time visual perception average stays within a narrow range from 73.4 to 73.1. Similarly, when changing the matching threshold from 0.60 to 0.84, the performance only fluctuates between 73.1 and 74.0. In all cases, ObjectStream consistently outperforms the Qwen2.5-VL-7B baseline by a large margin. These results indicate that our method is not sensitive to specific hyper-parameter choices and can maintain robust performance under different object selection and association settings.

### 7.9 Analysis of Decay

We further analyze the sensitivity of ObjectStream to EMA decay $\beta$ on the six realtime tasks of OVO-Bench. As shown in Fig. 5, ObjectStream is robust across a broad range of decay values, with performance varying by only 0.9 points. The best result of 73.5 is obtained at $\beta = 0.8$, while the default setting $\beta = 0.9$ achieves a comparable score of 73.3. A smaller decay factor ($\beta = 0.7$) updates object features more aggressively and may introduce short-term visual noise, whereas a larger value ($\beta = 0.95$) adapts too slowly to appearance changes, reducing the score

**Algorithm 1** Streaming Procedure of ObjectStream

**Input**: Video stream prefix $\mathcal{V}_\tau = \{I_t\}_{t=1}^{\tau}$, question $Q$
**Parameter**: object budget $M$, saliency quantile $q$, matching threshold $\theta$, token budgets $k_i, k_b$, temporal radius $r$, RVGW length $L$
**Output**: Predicted answer $\hat{Y}$

Initialize object set $\mathcal{O}^0 \leftarrow \emptyset$ and temporal set $\mathcal{B}^0 \leftarrow \emptyset$.
**for** $t = 1$ to $\tau$ **do**
  Encode the newly arrived frame: $\mathbf{X}_t \leftarrow \mathrm{Encoder}(I_t)$.
  Compute query-agnostic token saliency $\{s_{t,j}\}_{j=1}^{N}$.
  Select active token set $\mathcal{A}_t$ using the quantile threshold $q$.
  Build spatial graph $\mathcal{G}_t$ over $\mathcal{A}_t$ and obtain object candidates $\mathcal{C}_t$ by connected components.
  **for** each candidate $C_{t,m} \in \mathcal{C}_t$ **do**
    Compute candidate feature $f(C_{t,m})$ by saliency-weighted aggregation.
    Find the nearest object track $O_{i^\star} \in \mathcal{O}^{t-1}$ by feature similarity.
    **if** $\cos(f(C_{t,m}), \mu_{i^\star}) \geq \theta$ **then**
      Assign $C_{t,m}$ to $O_{i^\star}$.
      Update object feature, timestamp, consistency, saliency, and token pool.
      Compute object-level feature change and update temporal set $\mathcal{B}^t$.
    **else**
      Initialize a new object track from $C_{t,m}$.
    **end if**
  **end for**
  **if** $|\mathcal{O}^t| > M$ **then**
    Rank objects by consistency, saliency, and recency.
    Keep top-$M$ objects and evict low-priority tracks.
  **end if**
**end for**
Initialize $\mathcal{R}^{\tau}_{\mathrm{object}} \leftarrow \emptyset$.
**for** each maintained object track $O_i \in \mathcal{O}^{\tau}$ **do**
  Retain top-$k_i$ tokens from $\mathcal{T}_i^{\tau}$ according to saliency.
  Add the retained tokens to $\mathcal{R}^{\tau}_{\mathrm{object}}$.
**end for**
Retain temporal residual tokens around detected temporal events to form $\mathcal{R}^{\tau}_{\mathrm{temporal}}$.
Preserve raw visual tokens from the latest $L$ frames to form $\mathcal{R}^{\tau}_{\mathrm{rvgw}}$.
$\mathcal{R}^{\tau} \leftarrow \mathcal{R}^{\tau}_{\mathrm{object}} \cup \mathcal{R}^{\tau}_{\mathrm{temporal}} \cup \mathcal{R}^{\tau}_{\mathrm{rvgw}}$.
**return** $\hat{Y} \leftarrow \mathrm{LLM}(Q, \mathcal{R}^{\tau})$.

to 72.6. Nevertheless, all settings substantially outperform the Qwen2.5-VL-7B baseline of 63.3, demonstrating that ObjectStream is not sensitive to the exact choice of $\beta$. We use $\beta = 0.9$ by default as a stable balance between temporal consistency and adaptation to object appearance changes.

## 8 Visualization of Spatial Clustering

Fig. 6, Fig. 7 and Fig. 8 and visualize the frame-level latent object candidates produced by ObjectStream. Each colored box denotes the spatial extent of a connected visual-token component projected back onto the input frame. These boxes are used only for visualization and are not generated by an external object detector or segmentation model.

The examples show that the proposed spatial clustering can capture different types of task-relevant evidence. ObjectStream isolates text-bearing regions on product packaging, separates a held object from the person and background, and localizes interaction regions involving hands, tools, and surrounding objects. This allows heterogeneous visual evidence to be organized into object-centric memory while remaining directly aligned with the frozen VLM feature space.

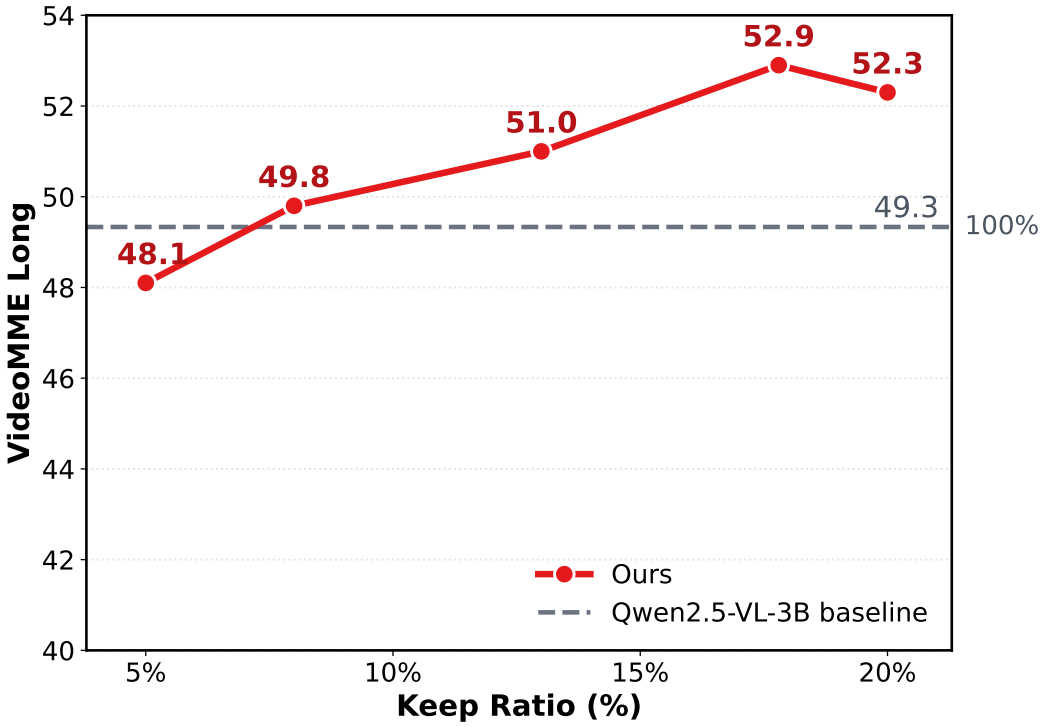



Figure 3: Effect of the retained visual token ratio on VideoMME-Long. ObjectStream surpasses the baseline when the retained ratio is at least 7.5% and achieves the best result at 17.5%.

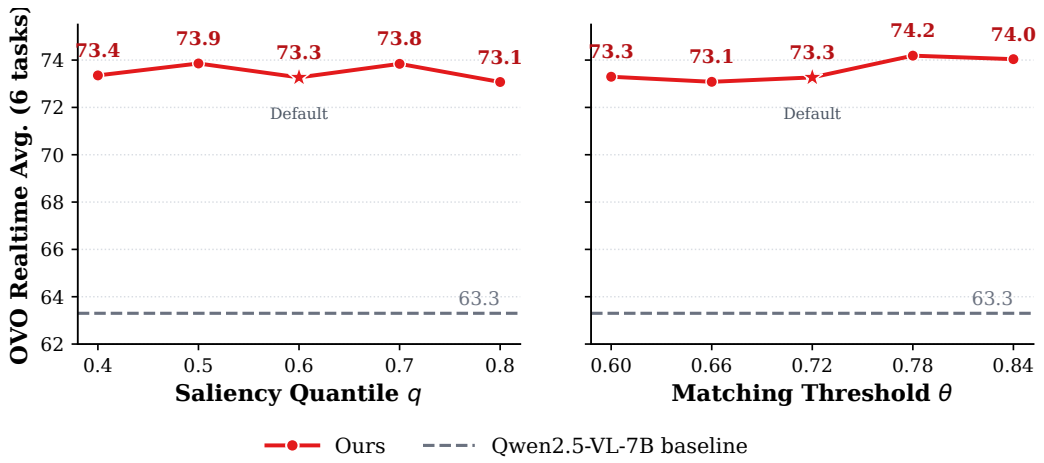



Figure 4: Analysis of the Saliency Quantile and Matching Threshold on OVO-realtime benchmark.

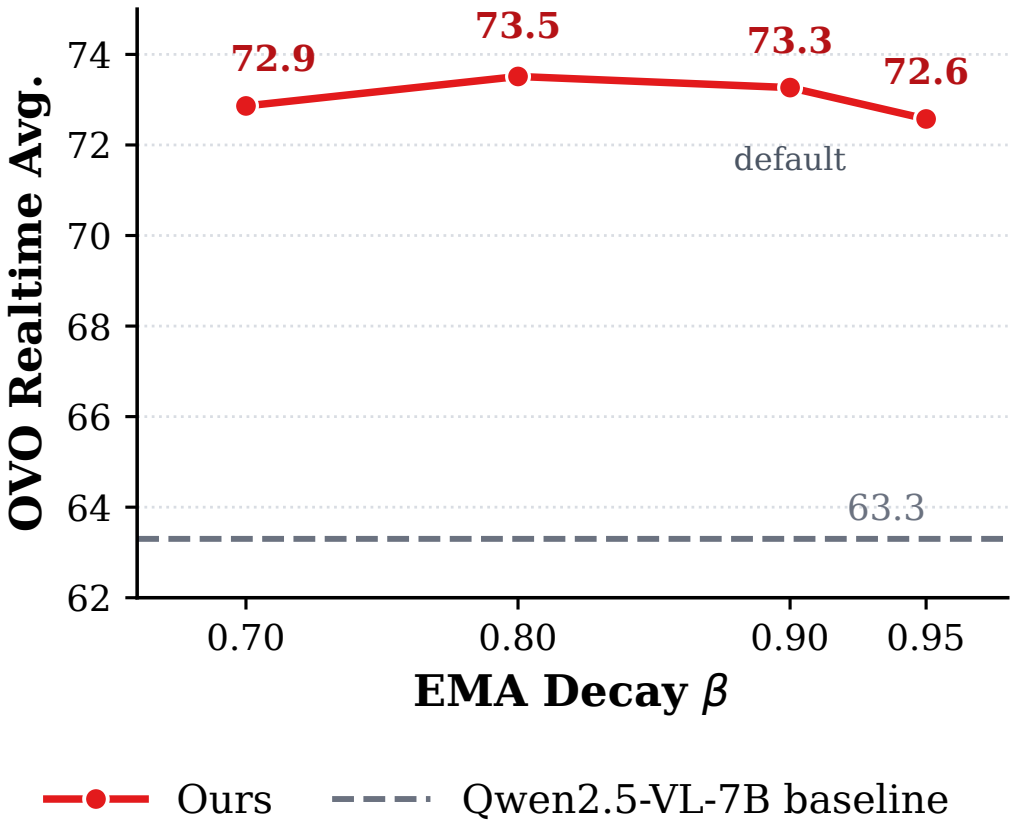



Figure 5: Analysis of the EMA Decay on OVO-realtime benchmark.

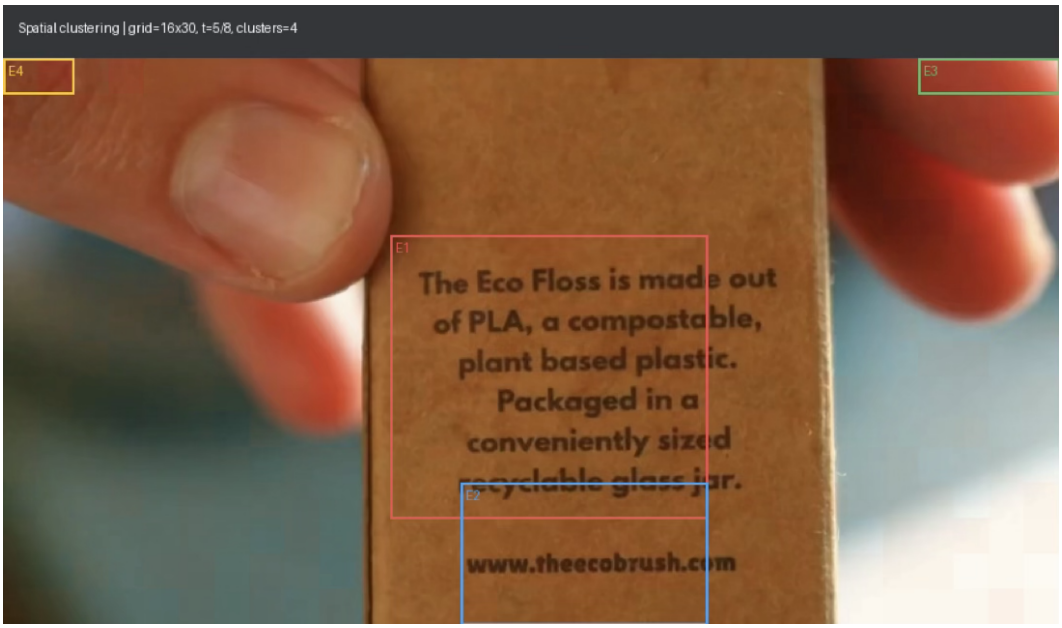



Figure 6: Visualization of spatial clustering on a text-rich object. ObjectStream isolates the printed description and other informative regions on the product package, preserving localized evidence useful for fine-grained recognition and text-related reasoning.




Figure 7: Visualization of spatial clustering for a held object. ObjectStream separates the soup can from the person and surrounding background, demonstrating that task-relevant objects can be localized directly from the VLM visual-token space.

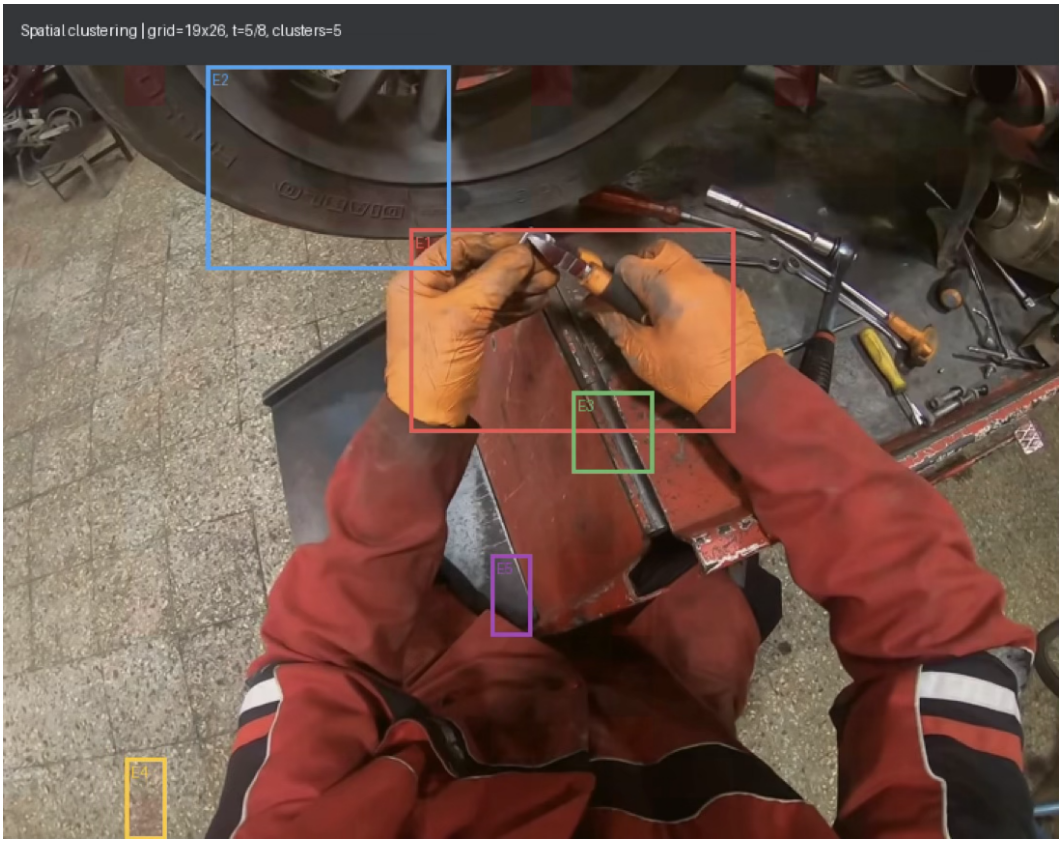



Figure 8: Visualization of spatial clustering in a cluttered first-person interaction. ObjectStream identifies localized regions corresponding to the hands, manipulated components, and surrounding work area, preserving interaction-relevant visual evidence under complex backgrounds.