Dynamic Hub-and-Spoke Memory for Streaming Video Understanding

Anonymous ACL submission

Abstract

Streaming video understanding requires answering questions at arbitrary times over a continuously growing visual stream. The central challenge is to compactly remember long-range history while effectively retrieving question-relevant evidence. We propose **D**ynamic **H**ub-and-**S**poke **M**emory **(D-HSM)**, a *training-free* framework that represents distant history as structured textual memory while preserving the recent frames as visual tokens for fine-grained perception. Specifically, D-HSM turns selected historical video chunks into typed textual observations and stores them in an entity-centered hub-and-spoke memory, with entities as hubs and related evidence as spokes. When answering a question, D-HSM dynamically retrieves a compact question-aware memory subset, expands it through hub-and-spoke links, and combines it with the recent visual window for frozen-VLM answer prediction. Extensive experiments on both streaming and long video benchmarks show that D-HSM consistently and substantially improves VLM backbones and outperforms other state-of-the-art online and offline video understanding baselines.

1 Introduction

> *"Remembering is not the re-excitation of innumerable fixed, lifeless and fragmentary traces. It is an imaginative reconstruction, or construction."*
>
> — Frederic C. Bartlett

Streaming video understanding faces a memory problem in Bartlett's sense: the model must reconstruct question-relevant evidence rather than merely replay the past. The requirement arises because a vision-language model (VLM) needs access to long-range visual evidence, while the continuously growing video history makes it infeasible to preserve the entire stream in raw visual form.

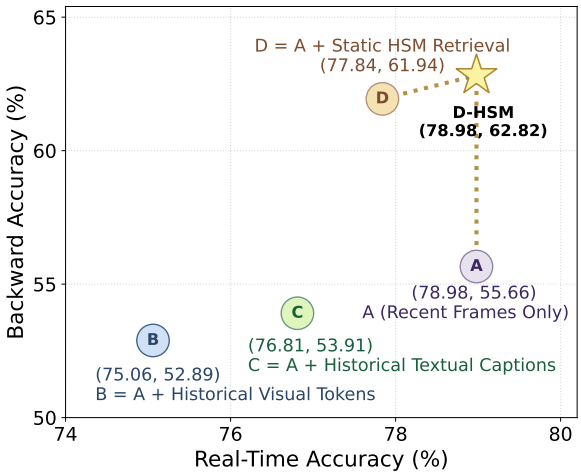


Figure 1: **Comparison of different historical strategies on OVO-Bench (Li et al., 2025c), measured by accuracy on "Real-Time" and "Backward" tasks.** The "Static HSM Retrieval" setting retrieves a fixed top-K budget of memory entries for each question."Real-Time" means that the questions can be answered based on the recent video frames, while 'Backward" questions require evidence from history video chunks.

This raises a central question: *how can a streaming model compactly remember the past while efficiently retrieving the evidence?*

Existing streaming video methods largely approach history from a token-budget perspective, reducing the past through sparse token selection (Yao et al., 2025; Shu et al., 2025; Wu et al., 2019), visual-token compression (Wu et al., 2024; Li et al., 2025b), bounded memory banks (Zhang et al., 2024; Wang et al., 2026), or retrieval over cached visual representations (Di et al., 2025; Yang et al., 2025). While these mechanisms improve online efficiency, they mainly ask *which* past visual units should be retained, but leave open *how* the retained history should be represented and organized for future questions. A simple diagnostic in Fig. 1 illustrates the representation gap: appending historical video tokens (B) is less effective than converting

the same segments into textual captions (C). This suggests that converting distant history into textual semantic traces can provide a useful way to retain it under a limited context budget. Yet flat captions remain segment-level summaries and lack explicit links among recurring entities, actions, and relations across time. Therefore, the following question naturally arises: ① *How should a streaming model organize long-range history into structured semantic memory?*

Besides, different streaming questions require different evidence. As shown in Fig. 1, questions about the current moment are often better answered from the recent visual window alone (A), while adding historical context (B, C, D) can introduce irrelevant entities or events and distract the VLM. In contrast, backward-looking or temporal questions require evidence from earlier segments, and cannot be answered reliably without retrieving history. A useful memory must therefore not only be well-organized, but also be queried adaptively. It should retrieve little or no history for real-time questions, and recover sufficient evidence for questions that depend on the past. Thus, along with question ①, another question needs to be answered for streaming video understanding: ② *How should a streaming model dynamically retrieve question-relevant history?*

To address both questions, we propose ***D-HSM***, a training-free **D**ynamic **H**ub-and-**S**poke **M**emory framework for streaming video understanding. D-HSM handles the asymmetry between long history and the current moment by storing distant history as compact textual memory, while keeping the recent window as visual frames for fine-grained perception. For long-range history, D-HSM first converts selected historical video chunks into structured textual observations that describe visible entities and their associated evidence. It then organizes these observations into an *entity-centered hub-and-spoke memory*, where recurring entities serve as **hubs** and related evidence is attached as **spokes**. This design turns flat segment captions into structured memory traces, allowing evidence about the same entity to be accumulated, merged, and localized across time.

When answering a question, D-HSM does not replay history in temporal order. Instead, it dynamically retrieves and constructs the relevant evidence on demand. It first applies keyword-based gating to determine whether memory retrieval is needed. When retrieval is triggered, D-HSM selects a memory subset according to the similarity between the question and memory entries. Notably, rather than using a fixed top-K budget, D-HSM determines the retrieval size by detecting a salient cutoff. It then expands the selected subset through the hub-and-spoke structure. The retrieved textual evidence of the subset is then combined with the recent visual tokens and passed to a frozen VLM for answer prediction. As illustrated by Fig. 1, D-HSM reconstructs question-relevant evidence from structured history while avoiding distraction from unnecessary historical information.

Together, the structured hub-and-spoke memory and the question-adaptive retrieval give a frozen VLM a way to remember the past compactly and access it selectively, without any additional training. Extensive experiments show that D-HSM achieves state-of-the-art performance on streaming benchmarks such as StreamingBench (Lin et al., 2024b) and OVO-Bench (Li et al., 2025c), outperforming strong proprietary models, as well as open-source online and offline video understanding baselines. Meanwhile, D-HSM also retains strong performance on conventional offline long-video understanding benchmarks, including LongVideoBench (Wu et al., 2024), MLVU (Zhou et al., 2025), and VideoMME (Fu et al., 2025). These results demonstrate the robustness and generality of D-HSM for compactly remembering long video history and retrieval question-relevant evidence on demand.

2 Related Work

Long and Streaming Video Understanding. Recent methods address long-range temporal reasoning mainly by retaining or compressing visual tokens. Offline long-video models (Chen et al., 2025; Wang et al., 2024b; Shu et al., 2025; Song et al., 2024; Wang et al., 2024a; Corona et al., 2025; Lin et al., 2026) extend visual context by sampling more frames or scaling representations, but their computational costs grow rapidly with video length. Online or streaming approaches (Chen et al., 2024a; Huang et al., 2024; Zeng et al., 2026; Carreira and Zisserman, 2017; Feichtenhofer et al., 2019; Bertasius et al.; Arnab et al., 2021; Liu et al., 2026; Zhang et al.) instead process frames sequentially. To manage history, sampling-based compression (Yao et al., 2025; Shu et al., 2025; Wu et al., 2019; Liu et al., 2025a) reduces input tokens but discards fine-grained spatial or temporal details. Alternatively, storage-based compression (Zeng et al., 2026; Gu and Dao, 2024; De et al., 2024; Wang

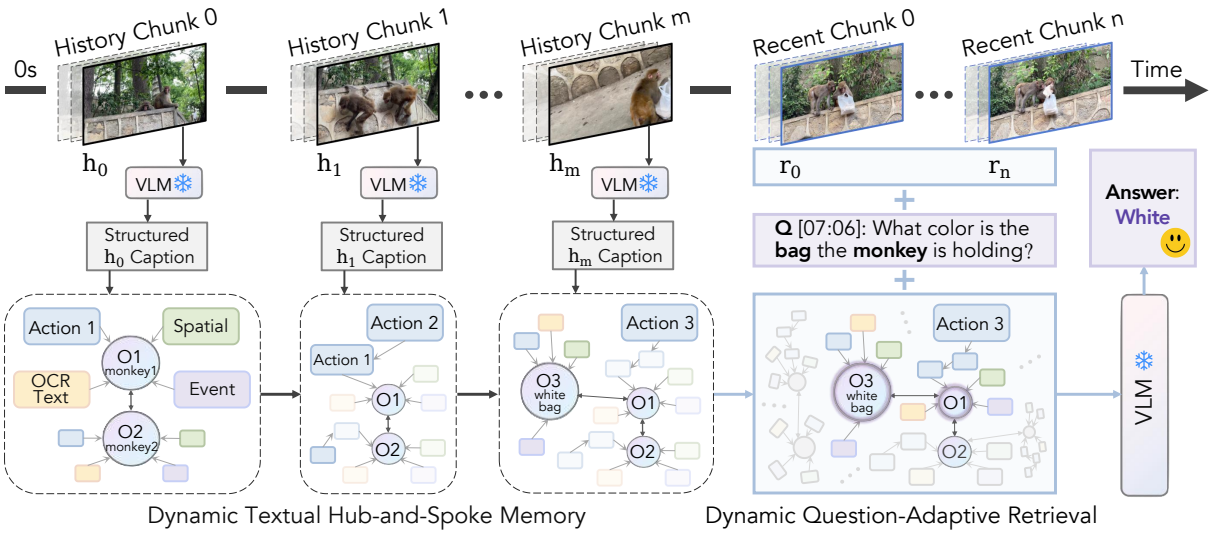


Figure 2: **Overview of D-HSM.** Historical chunks are converted by a frozen VLM into structured textual observations and incrementally organized into a dynamic hub-and-spoke memory. When a question arrives, D-HSM dynamically retrieves a compact question-adaptive subset from memory and expands it through hub-and-spoke links. Then, it combines the retrieved textual evidence with the recent visual tokens for frozen-VLM answer prediction.

et al., 2025; Li et al., 2024; He et al., 2024) maintains visual memory after encoding. These methods largely retain history as features without explicitly organizing the relationship between different entities. D-HSM instead stores distant history as an entity-centered hub-and-spoke memory, enabling compact and question-adaptive retrieval.

Video Large Language Models. Video Large Language Models (Video-LLMs) map spatiotemporal visual features into LLM spaces to enable open-ended dialogue and reasoning (Maaz et al., 2024; Lin et al., 2024a; Wang et al., 2024c; Li et al., 2023; Liu et al., 2024; Achiam et al., 2023). Standard architectures establish modality alignment via abstractors (Li et al., 2023; Ye et al., 2023), projection layers (Liu et al., 2024; Zhu et al., 2024), or temporal pooling operators (Zhang et al., 2023) over established vision backbones (Radford et al., 2021; Sun et al., 2023). Recent variants optimize efficiency via dynamic resolution encoding (Wu et al., 2025), 3D position scaling (Liu et al., 2025b), parameter-efficient tuning (Hu et al.), and sparse Mixture-of-Experts (MoE) topologies (Li et al., 2025d,a). To alleviate dense visual redundancy, token acceleration paradigms are categorized into pre-LLM configurations that merge tokens via visual clustering (Bolya et al.; Tao et al., 2025; Shao et al., 2026), and intra-LLM configurations that use text-to-visual attention for query-adaptive token pruning (Chen et al., 2024b; Huang et al., 2025).

3 Method

3.1 Framework

For the streaming video understanding, we require balance two forms of evidence: long-range historical context and immediate perception (Zeng et al., 2026; Lin et al., 2024b; Qian et al., 2024). These two information streams are inherently asymmetric, as long-range history is too voluminous to retain in raw form, while the present moment demands fine-grained visual detail that resists lossy abstraction. Building on this observation, we propose Dynamic Hub-and-Spoke Memory (D-HSM), a training-free framework that preserves recent video evidence as visual frames and consolidates longer-range history into a textual hub-and-spoke memory of compact semantic traces.

Specifically, as shown in Figure 2, D-HSM maintains a textual memory M_t for the historical stream. At each streaming step t, the historical chunk h_t is converted into a typed textual observation z_t, which is incrementally integrated into the entity-centered hub-and-spoke memory:

$$z_t = \phi_{\text{obs}}(h_t), \quad M_t = \mathcal{U}(M_{t-1}, z_t),$$

where M_t is the textual hub-and-spoke memory accumulated up to step t.

Questions may arrive at arbitrary streaming steps. When a question q_t arrives at step t, D-HSM dynamically selects a compact question-aware hub-and-spoke evidence subset C_t from the current

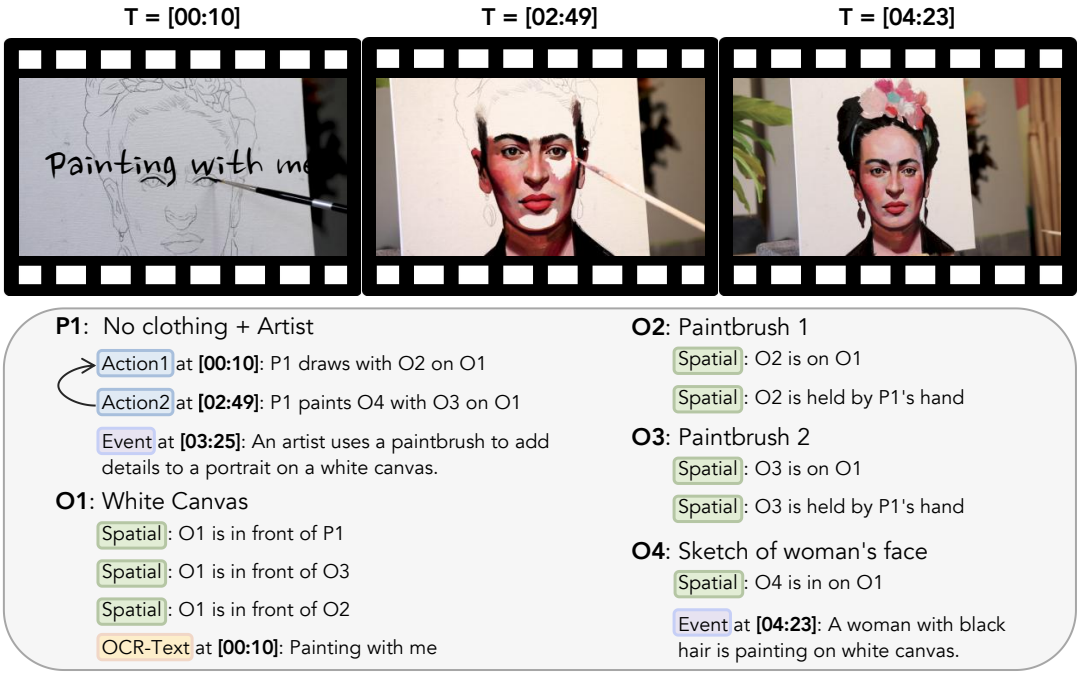


Figure 3: **Example of the hub-and-spoke memory representation.** Entities such as the "artist", "canvas", "paintbrushes", and "sketch" are stored as hubs with consistent identifiers, while their associated actions, spatial relations, OCR text, and events are stored as timestamped spokes. Repeated or related evidence is attached to the corresponding entity hub, enabling D-HSM to localize and connect evidence across different moments in the video.

memory M_t and combines it with the recent visual window r_t:

$$C_t = \mathcal{R}(M_t, q_t), \quad a_t = f_{\text{VLM}}(C_t, r_t, q_t),$$

where $\mathcal{R}$ denotes dynamic hub-and-spoke retrieval, f_{VLM} is the frozen VLM used for answer prediction, and a_t is the predicted answer.

3.2 D-HSM Construction and Update

D-HSM constructs the historical textual memory M_t in three stages: (i) it dynamically selects a bounded set of historical video chunks, (ii) it converts the selected chunks into structured textual observations, and (iii) it integrates these observations into an entity-centered Hub-and-Spoke memory.

Dynamic Historical Observation Budget. Before constructing memory, D-HSM adapts the density of historical observations to the length of the available video prefix. We construct historical chunks by uniformly sampling frames from the video, with each sampled frame treated as the representative of one chunk. Let $\mathcal{H}_t = \{h_1, \ldots, h_t\}$ be the candidate historical chunks at step t. D-HSM selects an active observation set:

$$\mathcal{S}_t = \begin{cases} \mathcal{H}_t, & t \leq B, \\ \Pi_B(\mathcal{H}_t), & t > B, \end{cases}$$

where B is the history budget and $\Pi_B(\cdot)$ denotes approximately uniform selection of B chunks from $\mathcal{H}_t$. As the video grows, the change in the active set is:

$$\Delta_t^+ = \mathcal{S}_t \setminus \mathcal{S}_{t-1}, \quad \Delta_t^- = \mathcal{S}_{t-1} \setminus \mathcal{S}_t.$$

D-HSM generates textual observations only for chunks in Δ_t^+, and excludes evidence supported only by chunks in Δ_t^- from the active memory. This allows short prefixes to be represented densely, while longer prefixes are represented by a sparser set of semantic traces.

Structured Textual Observations. For each selected historical chunk, D-HSM applies a fixed-schema prompt (details in Appendix B) to instruct a frozen VLM, yielding a textual observation: OBJECTS, PEOPLE, ACTIONS, OCR-TEXT, SPATIAL, and EVENT. OBJECTS and PEOPLE assign distinct identifiers to different visible entities ($O_1, O_2, \ldots$ for objects and $P_1, P_2, \ldots$ for people), and reuse the same identifier for the same entity across different chunks. ACTIONS describes interactions among these entities, OCR-TEXT captures screen text across optical character recognition, SPATIAL records positional relations, and EVENT summarizes the main event in the current chunk.

The schema converts each chunk into typed evidence rather than an unstructured caption. Per-

sistent entity identifiers make it easier to track the same person or object across chunks by assigning each visible entity a unique symbol, such as P_i or O_i, and reusing the same symbol whenever that entity reappears. The remaining fields preserve complementary evidence in separable forms. Together, these typed observations expose the information needed for later memory construction and retrieval.

Entity Hub-Centered Memory Construction and Update. As shown in Fig. 3, D-HSM is organized around *entity hubs* instantiated from OBJECTS and PEOPLE, which provide anchors for *who* or *what* is tracked across time. Other chunk-level evidence types are stored as *spokes* attached to the relevant entity hubs. These spokes answer *what happened*, *where it happened*, and *when it was observed*. Besides, each hub and spoke also stores its timestamps and occurrence count, allowing D-HSM to localize evidence in time.

As illustrated in Fig. 2, D-HSM updates the Hub-and-Spoke memory online. For each incoming observation at time t, it applies the following rules: (i) *Entity update.* If the observation mentions an existing entity identifier, D-HSM updates the corresponding hub instead of creating a duplicate. The hub refreshes its timestamps, and increments its occurrence count. If the identifier has not appeared before, a new entity hub is created. (ii) *Spoke insertion and merging.* For spokes that mention entity identifiers such as *P1* or *O2*, D-HSM attaches the spoke to the corresponding entity hub. When the spoke text does not mention any entity identifier, D-HSM compares the spoke embedding with existing entity-hub embeddings, and links it to the most similar hub. Then, D-HSM updates existing spokes for repeated facts by refreshing their timestamps and occurrence count, while inserting novel facts as new spokes. (iii) *Local relational update.* Entities that appear in the same chunk are connected with co-occurrence edges. These edges record local entity context, so retrieving one entity can also surface other entities observed in the same chunk. (iv) *Temporal action update.* For each entity hub, D-HSM links action spokes according to their observation order. These edges record the local order of actions performed by the same entity, so a retrieved action can bring in neighboring actions performed by the same entity.

3.3 Dynamic Hub-and-Spoke Retrieval

Given the current memory M_t and query q_t, D-HSM retrieves a compact, question-aware subset of the Hub-and-Spoke memory rather than passing the entire memory to the VLM. Retrieval proceeds in two steps: query-adaptive evidence selection and hub-and-spoke evidence expansion.

Question-Adaptive Evidence Selection. D-HSM first uses a lightweight keyword-based gate to skip memory retrieval for explicitly current-state questions (e.g., cues such as *now*, *currently*, or *recent frame*). For history-dependent questions, D-HSM embeds the query q_t and the memory entries, and computes cosine similarity scores. After applying a base threshold θ and a maximum budget K, we obtain a ranked candidate list $\{(e_i, s_i)\}_{i=1}^{n}$, where $s_i \geq s_{i+1}$ and $n \leq K$. D-HSM truncates this list at the most prominent *score gap*, when such a gap is statistically distinguishable from the rest of the distribution. Specifically, we examine the gaps:

$$\Delta_i = s_i - s_{i+1}, \quad i = 1, \ldots, n-1,$$

and locate the largest one, $\Delta^\star = \Delta_{i^\star}$ with $i^\star = \arg\max_i \Delta_i$. We accept $i^\star$ as the truncation point only if $\Delta^\star$ passes two tests. First, it must exceed a confidence-adaptive threshold:

$$\gamma_t = \max\big(\lambda_0,\ \lambda_1[s_1 - \theta]_+\big), \quad [x]_+ = \max(x, 0),$$

where λ_0 is an absolute floor and the second term tightens the threshold when the top score s_1 is high, so that a sharper gap is required when the retriever is confident. Second, $\Delta^\star$ must be at least twice the median gap, $\Delta^\star \geq 2\,\mathrm{median}(\{\Delta_i\})$, which rules out cases where scores decay smoothly and no single gap is genuinely salient. When both tests pass, we set $k_t = i^\star$; otherwise (including the degenerate case $n < 2$) we fall back to $k_t = n$. D-HSM then retrieves $\mathcal{E}_t = \{e_i\}_{i=1}^{k_t}$.

Hub-and-Spoke Evidence Expansion. Given the selected evidence entries $\mathcal{E}_t$, D-HSM expands them using the Hub-and-Spoke structure of the memory. For an entity hub, the expansion includes its attached spokes, together with entities that co-occurred in the same chunk as discussed in Section 3.2. Notably, when attaching action spokes, D-HSM follows a short next-action chain to recover neighboring actions performed by the same entity. For time-aware questions, it also retrieves timestamped events from the stored timeline.

Then, the resulting evidence set is linearized into a textual context with entity-centered profiles, screen-text snippets, chronological event lines, and counting aggregates as illustrated in Table B1. This context is then combined with the recent visual frames and the query for frozen-VLM.

Table 1: **Comparison results of different methods on StreamingBench Real-Time understanding tasks (Lin et al., 2024b).** D-HSM is evaluated with Qwen2.5-VL (Bai et al., 2025b) and Qwen3-VL (Bai et al., 2025a) using 4 or 8 recent frames, and compared with different proprietary, open-source offline, and open-source online video understanding models. "20+4" and "20+8" denote using 20 historical frames for memory construction and 4 or 8 recent frames for visual perception.

Model	Size	#Frames	OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	Overall
Human	-	-	89.5	92.0	93.6	91.5	95.7	92.5	88.0	88.8	89.7	91.3	91.5
PROPRIETARY MODELS													
GPT-4o	-	64	77.1	80.5	83.9	76.5	70.2	83.8	66.7	62.2	69.1	49.2	73.3
Claude 3.5 Sonnet	-	20	80.5	77.3	82.0	81.7	72.3	75.4	61.1	61.8	69.3	43.1	72.4
Gemini 1.5 pro	-	1 fps	79.0	80.5	83.5	79.7	**80.0**	84.7	77.8	64.2	72.0	48.7	75.7
OPEN-SOURCE OFFLINE MODELS													
VILA-1.5	8B	14	53.7	49.2	71.0	56.9	53.4	53.9	54.6	48.8	50.1	17.6	52.3
LongVA	7B	128	70.0	63.3	61.2	70.9	62.7	59.5	61.1	53.7	54.7	34.7	60.0
MiniCPM-v2.6	7B	32	71.9	71.1	77.9	75.8	64.6	65.7	70.4	56.1	62.3	53.4	67.4
LLaVA-OneVision	7B	32	80.4	74.2	76.0	80.7	72.7	71.7	67.6	65.5	65.7	45.1	71.1
Qwen2.5-VL	7B	1 fps	78.3	80.5	78.9	80.5	76.7	78.5	79.6	63.4	66.2	53.2	73.7
OPEN-SOURCE ONLINE MODELS													
Flash-VStream	7B	1 fps	25.9	43.6	24.9	23.9	27.3	13.1	18.5	25.2	23.9	48.7	23.2
VideoLLM-online	8B	2 fps	39.1	40.1	34.5	31.1	46.0	32.4	31.5	34.2	42.5	27.9	36.0
Dispider	8B	1 fps	74.9	75.5	74.1	73.1	74.4	59.9	76.1	62.9	62.2	45.8	67.6
TimeChatOnline	7B	1 fps	80.2	82.0	79.5	83.3	76.1	78.5	78.7	64.6	69.6	**58.0**	75.4
Streamforest	7B	1 fps	83.1	**82.8**	82.7	84.3	77.5	78.2	76.9	69.1	75.6	54.4	77.3
Qwen2.5-VL+D-HSM (4f)	7B	20+4	85.3	64.0	88.8	87.9	79.1	90.0	**87.6**	79.7	79.6	47.8	82.5
Qwen2.5-VL+D-HSM (8f)	7B	20+8	88.0	70.4	93.1	88.2	78.5	**93.8**	81.9	**80.9**	83.1	50.0	84.7
Qwen3-VL+D-HSM (4f)	8B	20+4	86.9	64.8	92.1	89.2	74.7	90.7	85.7	77.6	79.6	54.5	83.1
Qwen3-VL+D-HSM (8f)	8B	20+8	**88.6**	67.2	**95.1**	**90.8**	79.8	92.5	**87.6**	77.2	**84.0**	55.6	**85.4**

4 Experiments

4.1 Implementation Details.

D-HSM is training-free and keeps all VLM parameters frozen. We use Qwen2.5-VL-7B (Bai et al., 2025b) and Qwen3-VL-8B as backbone models (Bai et al., 2025a). For each video, we use 20 historical chunks by default to construct the hub-and-spoke memory. We set 4 recent frames and dynamic retrieval with a maximum budget of $K = 12$ as default setting. Memory entries and questions are embedded with a lightweight embedding encoder (bge-small-en-v1.5) (Xiao et al., 2023), and the dynamic cutoff gap $\lambda_0 = 0.05, \lambda_1 = 0.12$. All experiments are conducted on NVIDIA RTX A6000 GPUs without any task-specific fine-tuning.

4.2 Performance Comparison.

Streaming Video Understanding. Table 1 and Table 2 report results on StreamingBench (Lin et al., 2024b) and OVO-Bench (Li et al., 2025c). Across both benchmarks, *all D-HSM variants consistently outperform other methods*. On StreamingBench, D-HSM raises the overall score from 73.7 to 84.7 using Qwen2.5-VL, and further reaches 85.4 with Qwen3-VL. D-HSM surpasses the strongest proprietary model, Gemini 1.5 Pro, and the strongest open-source online model, Streamforest, by 9.7 and 8.1 points, respectively. The improvement suggests that the gains are not tied to a particular backbone, but are closely related to how D-HSM represents and retrieves long-range history.

On OVO-Bench, the improvement is not limited to a single question type: D-HSM achieves competitive results across Real-Time Visual Perception, Backward Tracing, and Forward Active Responding tasks. This indicates that D-HSM can dynamically adapt its evidence use, relying on the recent visual window for current-state questions while retrieving historical evidence for questions that require earlier moments.

Offline Long Video Understanding. Table 3 evaluates D-HSM on offline long-video understanding benchmarks, including LongVideoBench (Wu et al., 2024), MLVU (Zhou et al., 2025), and VideoMME (Fu et al., 2025). With Qwen2.5-VL as the backbone, D-HSM achieves 60.7 on LongVideoBench, 67.3 on MLVU, and 63.9 on VideoMME, outperforming the Qwen2.5-VL baseline and several open-source long-video models. This shows that, beyond the streaming setting, D-HSM can serve as an effective compact representa-

Table 2: **Comparison results of different methods on OVO-Bench (Li et al., 2025c).**

Model	Size	#Frames	Real-Time							Backward				Forward				Overall
			OCR	ACR	ATR	STU	FPD	OJR	**Avg.**	EPM	ASI	HLD	**Avg.**	REC	SSR	CRR	**Avg.**	**Avg.**
Human Agents	-	-	94.0	92.6	94.8	92.7	91.1	94.0	93.2	92.6	93.0	91.4	92.3	95.5	89.7	93.6	92.9	92.8
PROPRIETARY MODELS																		
GPT-4o	-	64	69.8	64.2	71.6	51.1	70.3	59.8	64.5	57.9	75.7	48.7	60.8	27.6	73.2	59.4	53.4	59.5
Gemini 1.5 pro	-	1 fps	85.9	67.0	79.3	58.4	63.4	62.0	69.3	58.6	**76.4**	52.6	62.5	35.5	**74.2**	61.7	57.2	63.0
OPEN-SOURCE OFFLINE MODELS																		
LongVU	7B	1 fps	53.7	53.2	62.9	47.8	68.3	59.8	57.6	40.7	59.5	4.8	35.0	12.2	69.5	60.8	47.5	46.7
LLaVA-OV	7B	64	66.4	57.8	73.3	53.4	71.3	62.0	64.0	54.2	55.4	21.5	43.7	25.6	67.1	58.8	50.5	52.7
LLaVA-Video	7B	64	69.1	58.7	68.8	49.4	74.3	59.8	63.5	56.2	57.4	7.5	40.4	34.1	70.0	60.4	54.8	52.9
Qwen2-VL	72B	64	65.8	60.6	69.8	51.7	69.3	54.4	61.9	52.5	60.8	57.5	57.0	**38.8**	64.1	45.0	49.3	56.3
OPEN-SOURCE ONLINE MODELS																		
VideoLLM-online	8B	2 fps	8.1	23.9	12.1	14.0	45.5	21.2	20.8	22.2	18.8	12.2	17.7	-	-	-	-	-
Dispider	8B	1 fps	57.7	49.5	62.1	44.9	61.4	51.6	54.6	48.5	55.4	4.3	36.1	18.1	37.4	48.8	34.7	41.8
TimeChatOnline	7B	1 fps	75.2	46.8	70.7	47.8	69.3	61.4	61.9	55.9	59.5	9.7	41.7	31.6	38.5	40.0	36.7	46.7
Streamforest	7B	1 fps	68.5	53.2	71.6	47.8	65.4	60.9	61.2	**58.9**	64.9	32.3	52.0	32.8	70.6	57.1	52.5	55.6
Streamo	7B	1 fps	77.2	66.1	76.7	45.5	66.3	72.8	67.4	55.6	58.1	33.9	49.2	30.8	57.6	**82.5**	57.0	57.9
Qwen2.5-VL+ D-HSM (4f)	7B	20+4	94.0	70.7	**86.2**	66.9	**75.3**	81.0	79.0	52.9	63.5	**72.0**	**62.8**	36.4	71.9	67.1	58.5	**66.8**
Qwen2.5-VL+ D-HSM (8f)	7B	20+8	**96.0**	73.4	81.0	66.3	**75.3**	82.1	79.0	52.2	60.8	62.4	58.5	36.4	**74.2**	67.1	**59.2**	65.6
Qwen3-VL+ D-HSM (4f)	8B	20+4	95.0	**82.6**	84.5	**72.5**	71.3	**83.7**	**81.6**	54.6	62.2	67.2	61.3	25.0	61.9	74.6	53.8	65.6
Qwen3-VL+ D-HSM (8f)	8B	20+8	94.0	81.7	80.2	67.4	72.3	81.5	79.5	55.2	65.5	62.4	61.1	24.9	64.7	74.6	54.7	65.1

Table 3: **Evaluation results on offline long-video understanding benchmarks.** We evaluate D-HSM on LongVideoBench (Wu et al., 2024), MLVU (Zhou et al., 2025), and VideoMME (Fu et al., 2025) against different methods. D-HSM results are reported with Qwen2.5-VL using 4 recent frames.

Model	Size	LongVideoBench (Val)	MLVU (M-Avg)	VideoMME (w/o subs)
Video Duration (Avg.)		473s	651s	1010s
Video Length		8s – 60min	3min – 120min	1min – 60min
PROPRIETARY MODELS				
GPT-4o	–	66.7	64.6	71.9
Gemini 1.5 Pro	–	64.0	–	75.0
OPEN-SOURCE OFFLINE MODELS				
LongVA	7B	–	56.3	52.6
Kangaroo	8B	54.8	61.0	56.0
LongVU	7B	–	65.4	60.6
Apollo	7B	58.5	70.9	61.3
SF-LLaVA-1.5	7B	62.5	71.5	63.9
InternVL2.5	8B	60.0	68.9	64.2
NVILA	8B	57.7	70.1	64.2
VideoLLaMA3	7B	59.8	73.0	66.2
OPEN-SOURCE ONLINE MODELS				
Dispider	7B	–	61.7	57.2
Streamforest	7B	–	**69.6**	61.9
TimeChatOnline	7B	57.7	65.4	62.5
D-HSM	7B	**60.7**	67.3	**63.9**

Table 4: **Ablation on memory composition.** We compare recent frames only, HSM only, and the full D-HSM to evaluate the contribution of each memory component. All results use Qwen2.5-VL-7B with 4 recent frames.

	Real-Time	**Backward**
HSM Only	44.79	58.02
Recent Only	78.98	55.66
D-HSM	**78.98**	**62.82**

Table 5: **Ablation on dynamic retrieval.** We compare fixed top-12 retrieval with the proposed dynamic retrieval strategy, whose maximum retrieval budget is also set to 12 (Qwen2.5-VL-7B, 4 recent frames).

Dataset	**Fixed Top-k**	**Dynamic**
OVO-Bench	64.71	**66.84**
StreamingBench	79.47	**82.45**

tion for long-range evidence in offline long-video understanding.

4.3 Ablation Study

Effect of Memory Composition. Table 4 ablates the two evidence sources in D-HSM. Recent frames alone achieve strong real-time accuracy but underperform on backward questions, while HSM alone improves backward reasoning but hurts real-time accuracy due to historical distraction. The full D-HSM combines both strengths, confirming the need to preserve recent visual perception while retrieving structured historical memory.

Effect of Dynamic Retrieval. We compare the proposed dynamic retrieval strategy with a fixed top-K strategy that always retrieves the maximum number 12 of memory entries. As shown in Table 5, dynamic retrieval improves performance on both benchmarks, indicating that adapting retrieval to different question types is beneficial.

Effect of Retrieval Budget. We vary the maximum retrieval budget K to study how much memory evidence should be made available during retrieval. As shown in Fig. 4, performance improves as K increases from 4 to 12, suggesting that a larger budget helps recover supporting long-range evidence. However, further increasing K to 16 does not bring additional gains and slightly reduces performance. This suggests that retrieving more memory entries

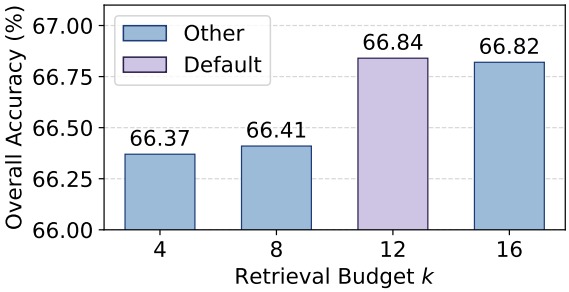


Figure 4: **Ablation on the maximum retrieval budget K on OVO-Bench** (Qwen2.5-VL-7B, 4 recent frames).

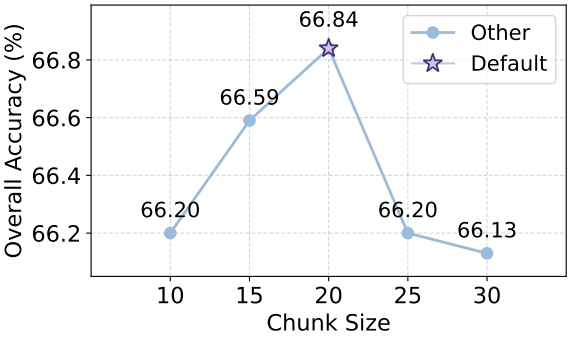


Figure 5: **Ablation on the historical observation budget on OVO-Bench** (Qwen2.5-VL-7B, 4 recent frames).

is not always beneficial, since weakly related evidence may introduce unnecessary tokens and distract the model. The result further supports the need for adapting the amount of retrieved evidence instead of always using a larger fixed budget.

Effect of Historical Observation Budget. We vary the historical observation budget by changing the number of chunks used to construct memory. As shown in Fig. 5, performance first improves as the budget increases, reaching the top with 20 chunks. This indicates that a larger historical budget can provide broader temporal coverage and preserve more useful long-range evidence. However, further increasing the budget to 25 or 30 chunks reduces performance. We attribute this drop to redundant or weakly relevant historical observations, which can introduce noise into memory and make retrieval less focused. These results suggest that D-HSM benefits from a moderate historical budget that balances coverage and compactness.

4.4 In-Depth Analysis

Inference Efficiency Analysis. We further evaluate the inference efficiency of D-HSM. As shown in Table 6, D-HSM requires only ~ 0.82 seconds of latency and ~ 17.8 GB of GPU memory per query.

Table 6: **Efficiency of D-HSM.** Latency and GPU memory are measured on StreamingBench (Qwen2.5-VL-7B, 4 recent frames).

Dataset	Latency (s)	GPU Memory (GB)
StreamingBench	~ 0.82	~ 17.8

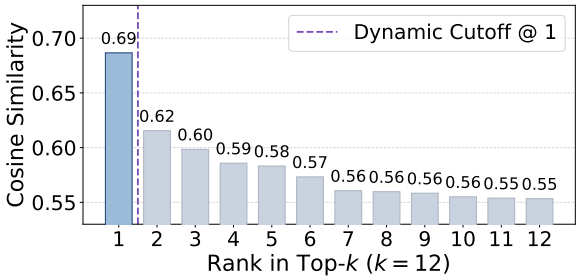


Figure 6: **Dynamic cutoff analysis for the question "What does the person do after chopping the onion?".** The similarity scores quickly flatten after the high-similarity prefix.

This indicates that D-HSM can provide strong performance while keeping cost manageable.

Dynamic Cutoff Analysis. We further analyze why dynamic retrieval is useful by visualizing the top 12 similarity distribution of the history memory entries. As in Fig. 6, the top entries have clearly higher similarity to the question, while the scores quickly flatten after the first few ranks. D-HSM cuts off retrieval after the high-similarity prefix, filtering weakly related memory entries and keeping the evidence compact. It helps reduce unnecessary historical context and prevents the VLM from being distracted by evidence that is weakly related to the question. Besides, this reduces the average number of retrieved tokens (e.g., by about 51% on OVO-Bench), thereby lowering inference cost.

5 Conclusion

We presented D-HSM, a training-free framework that balances long-range memory and immediate perception for streaming video understanding. D-HSM stores distant history as entity-centered hub-and-spoke textual memory while keeping recent frames for fine-grained perception. Its dynamic retrieval selects and expands question-relevant evidence on demand, reducing unnecessary historical context. Experiments on streaming and offline long-video benchmarks show that D-HSM outperforms existing online and offline baselines. These results highlight the value of reconstructing the past through structured, question-aware memory.

6 Limitations

D-HSM relies on frozen VLMs to generate structured textual observations, so errors in entity recognition, OCR, or action descriptions may propagate into memory. Since long-range history is stored as text, fine-grained details from distant moments may also be lost. Future work can explore more robust entity linking.

7 Ethical Considerations

Our work uses publicly available datasets and open-source models. Potential risks include hallucinated or biased model outputs and possible misuse of streaming video understanding systems in sensitive applications. However, our work does not collect private user data or involve human subjects. We encourage responsible research and deployment practices. All datasets and open-source models used in this work are publicly available and are used in accordance with their respective licenses and terms of use.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. 2021. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding?

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308.

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

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

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, and 1 others. 2025. Longvila: Scaling long-context visual language models for long videos. In *International Conference on Learning Representations*, volume 2025, pages 18227–18246.

Enric Corona, Andrei Zanfir, Eduard Gabriel Bazavan, Nikos Kolotouros, Thiemo Alldieck, and Cristian Sminchisescu. 2025. Vlogger: Multimodal diffusion for embodied avatar synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15896–15908.

Soham De, Samuel L Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, and 1 others. 2024. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Tao Zhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. 2025. Streaming video question-answering with in-context video kv-cache retrieval. *Preprint*, arXiv:2503.00540.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211.

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, and 2 others. 2025. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *Preprint*, arXiv:2405.21075.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. 2024. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13504–13514.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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

Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. 2024. Online video understanding: A comprehensive benchmark and memory-augmented method. *arXiv e-prints*, pages arXiv–2501.

Jiaze Li, Yaya Shi, Zongyang Ma, Haoran Xu, Huihui Xiao, Ruiwen Kang, Fan Yang, Tingting Gao, Di Zhang, and 1 others. 2025a. imove: Instance-motion-aware video understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 23959–23975.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. 2024. Videomamba: State space model for efficient video understanding. In *European conference on computer vision*, pages 237–255. Springer.

Ruanjun Li, Yuedong Tan, Yuanming Shi, and Jiawei Shao. 2025b. Videoscan: Enabling efficient streaming video understanding via frame-level semantic carriers. *Preprint*, arXiv:2503.09387.

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. 2025c. Ovo-bench: How far is your video-llms from real-world online video understanding? *Preprint*, arXiv:2501.05510.

Yunxin Li, Shenyuan Jiang, Baotian Hu, Longyue Wang, Wanqi Zhong, Wenhan Luo, Lin Ma, and Min Zhang. 2025d. Uni-moe: Scaling unified multimodal llms with mixture of experts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2024a. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 5971–5984.

Jingyang Lin, Jialian Wu, Ximeng Sun, Ze Wang, Jiang Liu, Yusheng Su, Xiaodong Yu, Hao Chen, Jiebo Luo, Zicheng Liu, and 1 others. 2026. Unleashing hour-scale video training for long video-language understanding. *Advances in Neural Information Processing Systems*, 38:17523–17552.

Junming Lin, Zheng Fang, Chi Chen, Zihao Wan, Fuwen Luo, Peng Li, Yang Liu, and Maosong Sun. 2024b. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. *Preprint*, arXiv:2411.03628.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306.

Hua Liu, Yanbin Wei, Fei Xing, Tyler Derr, Haoyu Han, and Yu Zhang. 2026. Graph2video: Leveraging video models to model dynamic graph evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 15315–15323.

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

Zikang Liu, Longteng Guo, Yepeng Tang, Tongtian Yue, Junxian Cai, Kai Ma, Qingbin Liu, Xi Chen, and Jing Liu. 2025b. Vrope: Rotary position embedding for video large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14471–14483.

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

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. 2024. Streaming long video understanding with large language models. *Preprint*, arXiv:2405.16009.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

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

Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. 2025. Video-xl: Extra-long vision language model for hour-scale video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26160–26169.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, and 1 others. 2024. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. 2023. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. 2025. Dycoke: Dynamic compression of tokens for fast video large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18992–19001.

Junxi Wang, Te Sun, Jiayi Zhu, Junxian Li, Haowen Xu, Zichen Wen, Xuming Hu, Zhiyu Li, and Linfeng Zhang. 2026. Streammeco: Long-term agent memory compression for efficient streaming video understanding. *Preprint*, arXiv:2604.09000.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. 2024a. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pages 58–76. Springer.

Xidong Wang, Dingjie Song, Shunian Chen, Chen Zhang, and Benyou Wang. 2024b. Longllava: Scaling multi-modal llms to 1000 images efficiently via hybrid architecture. *arXiv preprint arXiv:2409.02889*, 2(5):6.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, and 1 others. 2024c. Internvideo2: Scaling foundation models for multimodal video understanding. In *European conference on computer vision*, pages 396–416. Springer.

Yiyu Wang, Xuyang Liu, Xiyan Gui, Xinying Lin, Boxue Yang, Chenfei Liao, Tailai Chen, and Linfeng Zhang. 2025. Accelerating streaming video large language models via hierarchical token compression. *arXiv preprint arXiv:2512.00891*.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, and 1 others. 2025. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. 2024. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Preprint*, arXiv:2407.15754.

Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S Davis. 2019. Adaframe: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1278–1287.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

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

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, and 1 others. 2025. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 10807–10816.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, and 1 others. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.

Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, and 1 others. 2026. Streamforest: Efficient online video understanding with persistent event memory. *Advances in Neural Information Processing Systems*, 38:75804–75835.

Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 conference on empirical methods in natural language processing: system demonstrations*, pages 543–553.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. 2024. Flash-vstream: Memory-based real-time understanding for long video streams. *Preprint*, arXiv:2406.08085.

Kecheng Zhang, Zongxin Yang, Mingfei Han, Haihong Hao, Yunzhi Zhuge, Changlin Li, Zhihui Li, Xiaojun Chang, and 1 others. Progressive online video understanding with evidence-aligned timing and transparent decisions. In *The Fourteenth International Conference on Learning Representations*.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2025. Mlvu: Benchmarking multi-task long video understanding. *Preprint*, arXiv:2406.04264.

Deyao Zhu, Xiaoqian Shen, Xiang Li, Mohamed Elhoseiny, and 1 others. 2024. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations*, volume 2024, pages 18378–18394.

Appendix

A D-HSM Algorithm

This section provides the detailed algorithmic procedure of D-HSM. Algorithm 1 summarizes the overall streaming inference process. At each streaming step, D-HSM selects an active set of historical chunks, updates the hub-and-spoke memory with newly selected chunks, and answers a question when it arrives by combining retrieved textual evidence with the recent visual window. Algorithm 2 details the memory update procedure, including entity-hub creation, spoke insertion and merging, co-occurrence edge construction, and temporal action linking. Algorithm 3 describes the question-adaptive retrieval procedure, where D-HSM first skips memory retrieval for explicitly current-state questions and otherwise applies dynamic cutoff over the question-memory similarity distribution.

Algorithm 1 Overall D-HSM Streaming Inference

Require: Historical chunks $\{h_t\}_{t=1}^{T}$; recent visual window r_t; question q_t
Require: History budget B; retrieval budget K; threshold θ; gap parameters λ_0, λ_1
Ensure: Answer a_t when question q_t arrives

```
Initialize memory M_0 ← ∅ and active set S_0 ← ∅
for t = 1, ..., T do
    H_t ← {h_1, ..., h_t}
    S_t ← H_t if t ≤ B, otherwise Π_B(H_t)
    Δ_t^+ ← S_t \ S_{t-1}
    Δ_t^- ← S_{t-1} \ S_t
    M_t ← UpdateMemory(M_{t-1}, Δ_t^+, Δ_t^-)   ▷ Alg. 2
    if question q_t arrives then
        C_t ← Retrieve(M_t, q_t, K, θ, λ_0, λ_1)   ▷ Alg. 3
        a_t ← f_VLM(C_t, r_t, q_t)
    end if
end for
```

B Structured Observation Template

D-HSM does not store free-form captions as its historical memory. Each selected historical segment is a fixed-duration temporal chunk, optionally chosen by uniform subsampling under the memory budget. For each selected chunk, the frozen VLM is prompted to emit a fixed-schema textual observation. This observation acts as the segment summary used to update the hub-and-spoke memory.

The template contains six fields: OBJECTS, PEOPLE, ACTIONS, OCR TEXT, SPATIAL, and EVENT. In the implementation, the prompt uses the header `TEXT`; we refer to this field as OCR TEXT in the paper because it is intended to record readable on-screen text rather than arbitrary generated prose. The VLM is instructed to output `NONE` when a field has no visible evidence.

```
OBJECTS: <semicolon-separated visible objects>
PEOPLE: <semicolon-separated visible people>
ACTIONS: <semicolon-separated actions>
TEXT: <readable on-screen text>
SPATIAL: <semicolon-separated spatial relations>
EVENT: <one sentence summarizing the main event>
```

Objects and People. The object and person fields define entity hubs. Objects are written with stable identifiers $O_1, O_2, \ldots$ and concise visual attributes such as color, size, material, or state. People are written with stable identifiers $P_1, P_2, \ldots$, visible clothing or role cues, and a brief action. The same identifier is reused across segments when the same entity reappears.

Actions. The action field records interactions among visible entities, preferably using the stable identifiers. For example, an action may state that P_1 holds O_2, points toward O_3, or talks to P_2. These entries become action spokes linked to the corresponding entity hubs.

OCR Text. The OCR text field records readable text visible in the frame sequence, including slide text, subtitles, signs, labels, charts, and interface text. The prompt asks the VLM to copy this text verbatim when readable. These entries are stored as screen-text nodes and are retrieved separately for questions that depend on visual text.

Spatial Relations. The spatial field records positional relations between named entities, such as left/right, above/below, behind, or in front of. These relations preserve local visual layout in a textual form.

Event. The event field is a single-sentence summary of the most important thing happening in the segment. Unlike the entity and relation fields, this field is intended to provide a compact chronological event trace for temporal retrieval.

C More Ablation Experiments

Effect of Recent Visual Window Size. Table C1 studies the effect of the recent visual window. In-

Table B1: **Examples of textual evidence blocks rendered from D-HSM memory and provided to the VLM.** The exact entries depend on the retrieved hubs, spokes, screen-text nodes, and timeline events.

Rendered block	**Template shown to the VLM**

Entity-centered profile	`[P1 (blue-shirt woman)]`
	`[00:05] picks up O2 (red mug)`
	`[00:08] walks toward O3 (wooden table)`

Screen text	`[Screen Text]`
	`* "EXIT"`
	`* "12:30"`

Most recent events	`[Most recent events]`
	`[00:21] The person leaves the room.`
	`[00:24] The door closes.`

Timeline	`[Timeline]` `[00:05] A woman picks up a red mug near the table.`
Counting aggregate	`[Counting Aggregate – observed action occurrences across captioned chunks]` `'place' (e.g. "P1 places O2 on O3"): 3 time(s) at [00:05, 00:08, 00:13]`

Table C1: **Ablation on the recent visual window size.**

	OVO-Bench		**StreamingBench**	
Recent Frames	Qwen2.5 (VL-7B)	Qwen3 (VL-8B)	Qwen2.5 (VL-7B)	Qwen3 (VL-8B)
2f	64.93	64.01	80.46	81.18
4f	**66.84**	**65.59**	82.45	83.08
6f	66.53	64.69	83.42	84.86
8f	65.55	65.09	**84.73**	**85.36**

creasing the number of recent frames consistently improves StreamingBench, confirming the importance of fine-grained current perception for real-time streaming questions. In contrast, OVO-Bench peaks at 4 frames, suggesting that more recent visual context is not always beneficial when questions also require backward or temporal evidence. Together, these results show that D-HSM benefits from balancing recent visual perception with retrieved historical memory.

We compare three historical processing strategies: using only the proposed Hub-and-Spoke Memory (HSM), using only recent frames, and the full Qwen2.5-VL+D-HSM (4f) (Ours) framework.

Effect of different historical processing strategies. As shown in Table C2, for OVO-Bench, the method that using HSM Only has better performance than the one using Only Recent 4 Frames on Backward Tasks. While the Recent 4 Frames Only baseline achieves strong real-time performance, it suffers from degraded backward reasoning capability. In contrast, our full method significantly improves backward understanding while preserving strong real-time performance, demonstrating that the proposed D-HSM architecture effectively balances long-term historical context retention and real-time responsiveness.

For StreamingBench, our ablation results demonstrate the importance of jointly modeling recent observations and long-term historical memory. Using only the proposed HSM mechanism leads to substantially degraded performance, indicating that relying solely on compressed historical memory is insufficient for fine-grained real-time understanding. In contrast, the "Recent 4 Frames Only" baseline achieves strong performance on real-time perception tasks but exhibits limited capability in tasks requiring longer temporal reasoning and contextual consistency. By integrating recent frame observations with the proposed Dynamic Hub-and-Spoke Memory (D-HSM) architecture, Qwen2.5-VL+D-HSM (4f) achieves the best overall performance, improving the overall score from 81.22 to 82.45 while yielding notable gains on temporally demanding tasks such as PR and CT. These results demonstrate that D-HSM effectively balances short-term responsiveness with long-term contextual retention in streaming video understanding.

D Detailed Visualization of D-HSM

Figs. D1 and D2 visualize detailed examples of hub-and-spoke memory and integrated retrieval context by D-HSM for StreamingBench questions, while Figs. D3 and D4 show examples for OVO-Bench questions. Given the recent visual window and the question, D-HSM retrieves hub-and-spoke links

Table C2: **Ablation study of different historical processing strategies on OVO-Bench.** HSM means Fixed Hub-and-Spoke Memory, and D-HSM means Dynamic Hub-and-Spoke Memory. The setting of HSM for Top-k is $k = 12$. For Ours and HSM Only, the chunk size = 20. Ours in this table is D-HSM with 4 recent frames, and the backbone is Qwen2.5-VL-7B.

Model	**Real-Time**							**Backward**			
	OCR	ACR	ATR	STU	FPD	OJR	**Avg.**	EPM	ASI	HLD	**Avg.**
HSM Only	44.30	43.12	47.41	42.13	56.44	35.33	44.79	28.62	54.05	91.40	58.02
Recent 4 Frames Only	93.96	70.64	86.21	66.85	75.25	80.98	78.98	54.55	60.81	51.61	55.66
Ours *Qwen2.5-VL+D-HSM (4f)*	93.96	70.65	86.21	66.85	75.25	80.98	78.98	52.87	63.54	72.04	62.82

Table C3: **Ablation study of different historical processing strategies on StreamingBench (Lin et al., 2024b).** Our proposed method, Qwen2.5-VL+D-HSM (4f), significantly outperforms both the "HSM Only" and "Recent 4 Frames Only" baselines in overall performance, demonstrating that our Dynamic Hub-and-Spoke Memory (D-HSM) architecture achieves a superior balance between historical context retention and real-time responsiveness.

Model	**OP**	**CR**	**CS**	**ATP**	**EU**	**TR**	**PR**	**SU**	**ACP**	**CT**	**Overall**
HSM Only	54.50	47.20	68.75	60.00	59.49	53.58	64.76	38.62	46.06	35.56	53.72
Recent 4 Frames Only	85.29	64.00	90.79	88.20	73.42	90.03	76.19	79.27	78.72	36.56	81.22
Ours *Qwen2.5-VL+D-HSM (4f)*	85.29	64.00	88.82	87.87	79.11	90.03	87.62	79.67	79.59	47.78	82.45

from memory and renders them into a textual context for the frozen VLM. The retrieved links include event spokes, entity-centered action spokes, and short temporal action chains. The final retrieval context corresponds to the rendered evidence format shown in Table B1, but provides a more detailed example with concrete retrieved links and entity-specific evidence. It illustrates how D-HSM reconstructs question-relevant history instead of replaying all previous chunks.

E Descriptive Statistics

All reported results are averaged over three independent runs.

F Use of AI Assistants

AI assistants were used for language polishing and minor coding assistance. All technical content, experiments, and conclusions were verified by the authors.

G Package Usage

CUDA=12.4
python=3.10.20
torch==2.5.1
torchvision==0.20.1
accelerate==1.13.0
av==17.0.0
numpy==2.2.6
pillow==12.2.0
qwen-vl-utils==0.0.11
tqdm==4.67.3
transformers==5.5.0
typer==0.24.1
decord==0.6.0
flash-attn==2.8.3

H Model Size and Budget

The model sizes are 7B for Qwen2.5-VL, 8B for Qwen3-VL, and 33M for BGE-Small-EN-v1.5. Regarding inference time, since our method is training-free, there is no training GPU budget. For evaluation on StreamingBench, it takes approximately 1 hour and 20 minutes on 8 A6000 GPUs.

I Dataset Statistics

We summarize the datasets used in our experiments, including StreamingBench (Lin et al., 2024b) and OVO-Bench (Li et al., 2025c), LongVideoBench (Wu et al., 2024), MLVU (Zhou et al., 2025), and and VideoMME (Fu et al., 2025). We used test only splits for evaluation since our method is training-free.

Algorithm 2 Hub-and-Spoke Memory Update

Require: Previous memory M_{t-1}; added chunks Δ_t^+; removed chunks Δ_t^-
Ensure: Updated memory M_t

$M_t \leftarrow M_{t-1}$
Exclude evidence supported only by chunks in Δ_t^- from the active memory
for all $h_j \in \Delta_t^+$ **do**
  $z_j \leftarrow \phi_{\text{obs}}(h_j)$
  Extract entity mentions $\mathcal{N}_j$ from PEOPLE and OBJECTS
  Extract spoke candidates $\mathcal{P}_j$ from ACTIONS, OCR-TEXT, SPATIAL, and EVENT
  for all $u \in \mathcal{N}_j$ **do**
    if u matches an existing entity hub in M_t **then**
      Refresh the hub timestamps and occurrence count
    else
      Create a new entity hub for u
    end if
  end for
  for all $p \in \mathcal{P}_j$ **do**
    if p explicitly mentions entity identifiers **then**
      Attach p to the corresponding entity hubs
    else
      Attach p to the most similar entity hub by embedding similarity
    end if
    Merge p with repeated spokes, or insert it as a new spoke
  end for
  Add co-occurrence edges between entities appearing in h_j
  Link action spokes of each entity according to observation order
end for
return M_t

Algorithm 3 Dynamic Hub-and-Spoke Retrieval

Require: Memory M_t; question q_t; retrieval budget K; base threshold θ; gap parameters λ_0, λ_1
Ensure: Linearized textual context C_t

if q_t contains current-state cues **then**
  return empty textual context $C_t \leftarrow \emptyset$
end if
Compute cosine similarity scores between q_t and all memory entries
Keep entries above θ and retain at most top K entries
Obtain ranked candidates $\{(e_i, s_i)\}_{i=1}^{n}$ with $s_i \geq s_{i+1}$ and $n \leq K$
if $n < 2$ **then**
  $k_t \leftarrow n$
else
  $\Delta_i \leftarrow s_i - s_{i+1}$ for $i = 1, \ldots, n-1$
  $i^\star \leftarrow \arg\max_i \Delta_i, \quad \Delta^\star \leftarrow \Delta_{i^\star}$
  $\gamma_t \leftarrow \max(\lambda_0, \lambda_1[s_1 - \theta]_+)$
  if $\Delta^\star \geq \gamma_t$ and $\Delta^\star \geq 2\,\text{median}(\{\Delta_i\})$ **then**
    $k_t \leftarrow i^\star$
  else
    $k_t \leftarrow n$
  end if
end if
$\mathcal{E}_t \leftarrow \{e_i\}_{i=1}^{k_t}$
Expand $\mathcal{E}_t$ through attached spokes, co-occurrence edges, and temporal action edges
Add timestamped events when the question is time-aware
Linearize the expanded evidence into textual context C_t
return C_t

QUESTION

task **streamingbench**

Which of the following best summarizes the artistic process just shown in the clip?

Pred **D** ✓ correct

Hub–Spoke Links (12)

LINK #1

EVENT A painting of an ocean scene is displayed alongside a paint palette and brush.

—

—

—

LINK #2

EVENT A hand uses a paintbrush to add details to a sunset painting on a canvas.

—

—

—

LINK #3

ACTION P1 paints on O1 with O4

ENTITY P1 (no clothing color + artist)

—

ACTION P1 paints on O1 with O3 → ACTION P1 paints on O1 with O2 [towards canvas]

LINK #4

ACTION P1 paints with O4 [on O1] towards O3

ENTITY P1 (no clothing color + artist)

—

ACTION P1 holds canvas with left hand → ACTION P1 paints on O1 with O4

... 8 more link(s) omitted

Retrieval Context (excerpt)

```
[Most recent events]
  [10:39] A sunset painting and an acrylic paint palette with a paint knife are shown.
  [11:26] A hand uses a paintbrush to mix colors on a glass palette and then applies them to a painting.
  [12:11] A woman in a blue shirt paints a sunset scene on a canvas.
  [12:57] A woman in a blue shirt paints a sunset scene on a canvas using a paintbrush.
  [13:42] A painting of an ocean scene is displayed alongside a paint palette and brush.
  [14:28] An artist paints a beach scene using a paintbrush and a palette of colors.

[P1 (no clothing color + artist)] ×11
  [01:31] P1 paints with O4 [on O1] towards O3
  [02:16] P1 holds canvas with left hand
  [03:02] P1 paints on O1 with O4
  [05:19] P1 paints on O1 with O3 (×6)
  [08:23] P1 paints on O1 with O2 [towards canvas]
  [11:26] P1 paints on O1 with O2 using O3

[Other]
  • [01:31] An artist uses a paint knife to apply paint from a palette onto a canvas.
```

... (18 more lines)

Figure D1: **Visualization of hub-and-spoke memory details and the integrated retrieval context for the question from StreamingBench.**

QUESTION

task **streamingbench**

What is the best summary of the actions taken by the individual just now?

Pred **C** ✓ correct

-3s -2s -1s **query** +1s

Hub–Spoke Links (1)

LINK #1

ACTION P1 adds chopped onions from O2 to hot dog in O1 with O4

ENTITY P1 (green shirt + cook)

—

ACTION P1 places food in O1 with O2 → ACTION P1 holds O1 with O2

Retrieval Context (excerpt)

```
[Most recent events]
  [04:21] A cook in a green shirt adds chopped onions to a hot dog using a spoon.
  [04:40] A person wearing gloves wraps food in a paper wrap and places it on a metal tray.
  [04:59] A person wearing a white shirt holds a brown paper bag while another person wearing a white shirt and black gloves
holds a white cup.
  [05:17] A cook wearing white gloves gestures towards a metallic tray.
  [05:36] A cook in a green shirt holds a fork while standing at the counter.
  [05:55] A cook in a green shirt holds a hot dog while another cook in black gloves uses a spatula.

[P1 (green shirt + cook)] ×20
  [02:29] P1 adds chopped onions from O2 to hot dog in O1 with O4
  [02:48] P1 places food in O1 with O2
  [03:07] P1 holds O1 with O2

[Timeline]
  [00:00] A cook in a green shirt holds black gloves.
  [00:18] A cook in a green shirt handles sausages in a metal sink.
  [00:37] A cook wearing black gloves holds a container of hot dogs.
  [00:56] A cook wearing white gloves prepares a hot dog by picking up a cooked sausage from a fryer.
```

... (6 more lines)

Figure D2: **Visualization of hub-and-spoke memory details and the integrated retrieval context for the question from StreamingBench.**

QUESTION

task **OVO** id **1249**

What is the goalkeeper doing?

GT **A** Pred **A** ✓ correct

-3s | -2s | -1s | **query** | +1s

Hub–Spoke Links (12)

LINK #1

ACTION P2 crouches near goal

ENTITY P2 (yellow jersey + player)

—

ACTION P2 stands near goal → ACTION P2 raises arms in the air

LINK #2

ACTION P1 dribbles O2 with right foot

ENTITY P1 (white shirt + referee)

—

—

LINK #3

ACTION P1 runs towards goal

ENTITY P1 (white shirt + referee)

—

ACTION P1 runs towards center of field → ACTION P1 kicks O3 with right foot

LINK #4

ACTION P3 stands near goal

ENTITY P3 (blue shirt + player)

—

ACTION P3 holds ball with both hands → ACTION P3 runs towards center of field

... 8 more link(s) omitted

Retrieval Context (excerpt)

```
[O1 (white shirt)] ×20
  O1 is behind goal

[O2 (black pants)] ×19
  O2 is to the right of goal

[P1 (white shirt + referee)] ×20
  [00:07] P1 runs towards goal (×5)
  [00:15] P1 runs towards center of field (×2)
  [00:19] P1 kicks O3 with right foot
  [01:14] P1 dribbles O2 with right foot

[P2 (yellow jersey + player)] ×16
  [00:07] P2 runs towards goal
  [00:11] P2 runs towards ball (×2)
  [00:15] P2 runs towards center of field
  [00:23] P2 dribbles ball with right foot
  [00:27] P2 crouches near goal
```

... (24 more lines)

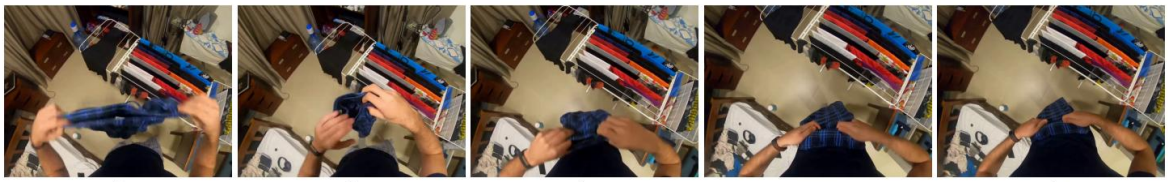
Figure D3: **Visualization of hub-and-spoke memory details and the integrated retrieval context for the question from OVO-Bench.**

QUESTION

task **OVO** id **1236**

What action is he performing with the blue checkered cloth?

GT **B** Pred **B** ✓ correct



-3s -2s -1s **query** +1s

Hub–Spoke Links (12)

LINK #1

ACTION P1 folds blue shirt with both hands

ENTITY P1 (black shirt + man)

—

ACTION P1 reaches for O2 on O4 → ACTION P1 folds O3 with both hands

LINK #2

ACTION P1 hangs clothes on O2 with both hands

ENTITY P1 (black shirt + man)

—

ACTION P1 adjusts O1 with both hands → ACTION P1 holds O1 with both hands

LINK #3

SPATIAL O3 is on P1's torso

ENTITY O3 (white drying rack)

—

—

LINK #4

ACTION P1 folds O1 with both hands

ENTITY P1 (black shirt + man)

—

ACTION P1 folds a dark blue shirt with both hands → ACTION P1 hangs clothes on O2 with both hands

... 8 more link(s) omitted

Retrieval Context (excerpt)

```
[O1 (green shirt)] ×20
  O1 is on the clothes rack
  O1 is on the floor

[O2 (blue chair)] ×20

[O3 (white drying rack)] ×20
  O3 is above O6
  O3 is on the floor
  O3 is to the left of P1
  O3 is on P1's torso
  O3 is in front of P1
  O3, O4, and O5 are on O2
  O3 is in the background (×2)
  O3 is to the right of P1
  O3 is right of O2
  O3 is in P1's hands
  O3 is behind O2
```

... (37 more lines)

Figure D4: **Visualization of hub-and-spoke memory details and the integrated retrieval context for the question from OVO-Bench.**