TRACE: TEMPORAL RESOURCE-AWARE CHAINS OF EVIDENCE FOR LONG-RANGE VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming long-video understanding requires a model to answer user queries at any moment while maintaining visual and conversational context across multiple rounds. However, existing methods either reduce memory through flat representations that lose the temporal relations among events, or maintain such relations in structures that grow continuously with the stream. To address these issues, we propose TRACE, a training-free online framework that organizes streaming memory into queryable chains of evidence with resource-aware retention and retrieval. During ingestion, we compare each observation with its predicted semantic state and retain surprising event changes in full detail while merging repeated content. The retained observations are then parsed into subject-action-object event nodes and organized into a spatio-temporal chain that preserves entity trajectories and event relations. When a query arrives, an intention-driven resource-aware router formulates evidence access as a budgeted selection problem and dynamically allocates the limited resource budget to maximize the utility of complementary evidence for response generation. Extensive experiments show that our framework achieves superior performance with accuracy of 76.2% and answer-quality score of 3.88 on StreamBench. It also demonstrates high throughput of 94.05 FPS with low latency of 3.36 s and maintains generalization capabilities on the other complementary benchmarks.

1 INTRODUCTION

Multimodal Large Language Models (MLLMs) have substantially advanced video understanding by extending language-based reasoning to complex visual sequences (Maaz et al., 2024; Zhang et al., 2025b; Chen et al., 2024b; Xu et al., 2024; Alayrac et al., 2022; Liu et al., 2023; Zhang et al., 2023). However, most existing models assume that the complete video is available before inference. This assumption does not hold in streaming scenarios, where observations arrive continuously and users may ask questions at any time (Chen et al., 2024a; Huang et al., 2025; Xiong et al., 2025; Wang et al., 2025a). In multi-turn streaming video dialogue, a later question may depend on an earlier event, its evolution across subsequent events, and information established in previous turns. The central challenge is therefore to preserve query-relevant temporal evidence over an unbounded stream while controlling memory growth and inference cost.

Existing methods control the growing visual context through either temporal memory compression or selective token retention. Memory-based approaches summarize historical observations into sparse or recurrent states (Song et al., 2024; He et al., 2024; Wang et al., 2025b; Zeng et al., 2025; Wu et al., 2024), whereas token-based approaches merge or prune visually redundant representations (Lee et al., 2024; Shen et al., 2025; Li et al., 2026; Yao et al., 2025; Kim et al., 2025). Online systems further maintain recent frames or compact streaming caches to support causal responses (Zhang et al., 2025a; Xiong et al., 2025; Di et al., 2025; Yang et al., 2025a). These strategies reduce context size, but their retention criteria are commonly governed by recency, visual similarity, or cache capacity rather than the role of an event in a future evidence chain. As illustrated in Figure 1, a sliding window may evict an early interaction, while aggressive compression may remove a brief event that connects two later states. The fundamental problem is therefore not only which observations remain in memory, but

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



Figure 1: Motivation for TRACE. (a) Sliding-window memory retains recent observations while early keyframes fall outside the active context. (b) Compressed memory banks may omit brief event details required by later queries. (c) TRACE represents retained subject-action-object events as entity-centric evidence chains for query-time access.

whether the retained observations preserve a coherent and queryable account of how entities and their interactions evolve over time.

We identify the central challenge in streaming video understanding as preserving query-relevant temporal evidence while limiting storage growth and query-time context over an unbounded stream. To address this challenge, we propose TRACE, a training-free framework that organizes retained streaming evidence into entity-centric chains. To organize retained evidence over time, we formulate Entity-Centric Event Trajectory Construction (EETC). EETC represents observations as Subject-Action-Object (SAO) event nodes and connects them through entity identity, temporal order, and cross-entity interactions. The resulting trajectories keep the evolution of entities and their relations traceable across distant moments. To control visual storage, we further design Semantic Surprise-Guided Event Filtering (SSEF). SSEF compares each observation with its predicted semantic state and assigns merge, key-frame, or full-feature storage according to its redundancy and surprise. At query time, the Intention-Driven Resource-Aware Router (IRR) identifies the entities and trajectories relevant to the question. IRR then formulates evidence allocation as a two-budget multiple-choice knapsack problem and selects complementary visual and textual evidence at suitable fidelities. Extensive experiments on StreamBench and six complementary video benchmarks demonstrate strong video understanding across streaming and offline settings. TRACE also maintains high throughput, low response latency, and bounded memory consumption during the testing process.

Our main contributions are summarized as follows:

- We reveal the unresolved challenge of preserving query-relevant temporal evidence over unbounded video streams while limiting storage growth, and propose TRACE, a training-free framework that converts streaming observations into queryable chains of evidence.
- We build up an entity-centric memory paradigm through semantic surprise-guided event filtering and event trajectory construction. The resulting memory suppresses repetitive observations while preserving informative transitions and long-range entity relations.
- We formulate query-time evidence access for long-range video question answering as a two-budget multiple-choice knapsack problem, and propose an intention-driven resource-aware router to select complementary visual and textual evidence at suitable fidelities.
- Extensive experiments on StreamBench and six complementary video benchmarks demonstrate the superior video understanding performance of TRACE across streaming and offline settings, together with high throughput, low response latency, and bounded memory consumption.

2 Related Work

2.1 Long-Video Representation Compression

Long-video representation compression methods can be grouped into sparse memory construction, recurrent memory aggregation, and token-level compression. MovieChat (Song et al., 2024) builds sparse memories by combining short-term and long-term visual information. MA-LMM (He et al., 2024) maintains a memory bank to aggregate multimodal information over long videos, while VideoLLaMB (Wang et al., 2025b) propagates historical information through recurrent memory bridges. StreamForest (Zeng et al., 2025) further introduces persistent event memory for online video understanding. Token-level methods reduce the visual context by removing or merging redundant representations. Video Token Merging (Lee et al., 2024) merges similar video tokens, LongVU (Shen et al., 2025) performs spatiotemporal adaptive compression, and VideoChat-Flash (Li et al., 2026) applies hierarchical compression to long-context video modeling. TimeChat-Online (Yao et al., 2025) analyzes the redundancy of visual tokens in streaming videos, while InfiniPot-V (Kim et al., 2025) compresses the KV cache under memory constraints. These methods mainly reduce the size of visual tokens, memory states, or KV caches, but they do not explicitly organize retained observations according to entity evolution and future query needs. TRACE instead combines semantic surprise-guided retention, entity-centric temporal trajectories, and intention-driven evidence retrieval for streaming video understanding.

2.2 Online Video-Language Models

Online video-language models mainly focus on causal processing, compact streaming memory, and interactive response generation. VideoLLM-online (Chen et al., 2024a) processes incoming video chunks incrementally and generates responses without observing the complete video. VideoChat-Online (Huang et al., 2025) further studies online video understanding under streaming inputs and interactive queries. Several methods improve the efficiency of long-term streaming perception. Flash-VStream (Zhang et al., 2025a) compresses historical visual features for real-time video understanding, while StreamBridge (Wang et al., 2025a) adapts an offline video large language model to proactive streaming assistance. StreamChat (Xiong et al., 2025) combines streaming perception with memory-enhanced knowledge for multi-round interaction, and VideoLLaMB (Wang et al., 2025b) uses recurrent memory bridges to preserve long-term information. Other studies introduce active interaction or query-dependent memory selection. Dispider (Qian et al., 2025) separates perception, decision, and reaction for active real-time interaction. StreamForest (Zeng et al., 2025) maintains persistent event memory, QueryStream (Zhang et al., 2026) performs query-aware pruning, and EventMemAgent (Wen et al., 2026) develops hierarchical event-centric memory with adaptive tool use. Related structured approaches also explore entity relation graphs and explicit scene graphs for long-video reasoning (Chu et al., 2025; Ma et al., 2026). Despite these advances, existing online systems generally maintain compressed sequential states, recent feature caches, or query-dependent subsets rather than explicit entity-centric temporal trajectories. TRACE organizes retained subject-action-object events into entity trajectories and retrieves query-relevant evidence under bounded visual and textual budgets.

3 Method

We present TRACE, a training-free framework for streaming video understanding with resource-aware retention and retrieval. The input is a video stream that arrives frame by frame, and questions may arrive at any time while the stream continues. The stream is partitioned into action-delimited segments, and each interaction observed in a segment is parsed into a SAO event with its temporal interval and scene context. Our framework processes these events online in three stages. The first stage decides for each event whether it is worth storing and at which visual granularity. The second stage organizes stored events into entity-indexed temporal chains. The third stage retrieves and selects evidence under explicit visual and textual budgets.

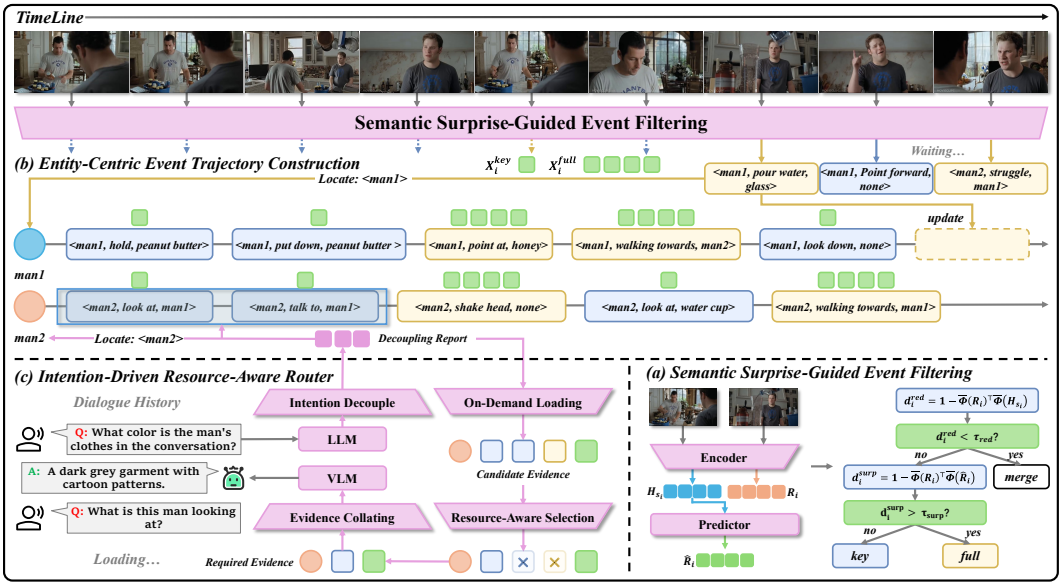


Figure 2: Overview of TRACE. TRACE maintains queryable evidence chains from an unbounded video stream through (a) Semantic Surprise-Guided Event Filtering, (b) Entity-Centric Event Trajectory Construction, and (c) Intention-Driven Resource-Aware Router. It preserves informative event changes, organizes their temporal relations, and retrieves query-relevant evidence under explicit visual and textual budgets.

3.1 Semantic Surprise-Guided Event Filtering

Semantic Event Representation. A streaming video contains persistent actions as well as state-changing events, and these should not cost the same amount of storage. We therefore evaluate every incoming event by the semantic information it adds beyond what is already recorded and what is expected from the current context, following the predictive coding view that deviations from an expected state carry informative signals (Rao & Ballard, 1999). For the i-th interaction extracted from an action segment we define an SAO event $\mathbf{g}_i = \langle s_i, a_i, o_i \rangle$ with subject identity s_i, action a_i, and object o_i. The event also carries its temporal interval $\mathcal{I}_i = [t_i^s, t_i^e]$ and its scene context ℓ_i. We convert these fields into a canonical semantic representation R_i that removes variations in surface expression while preserving the SAO semantics and the temporal interval.

Predictive Semantic State. The source segment provides two candidate visual payloads, namely a representative-frame feature $\mathbf{X}_i^{\text{key}}$ and the complete segment tensor $\mathbf{X}_i^{\text{full}}$. Let H_{s_i} denote the latest semantic state recorded on the chain of subject s_i. From the recorded state H_{s_i}, the scene context ℓ_i, and the interaction object o_i, a semantic predictor Φ_{pred} produces the expected state $\widehat{R}_i = \Phi_{\text{pred}}(H_{s_i}, \ell_i, o_i)$ of the subject. The expected state expresses what the subject should be doing given where it is and what it interacts with, rather than forecasting a future event. We compare observations, recorded states, and expected states in a shared semantic embedding space using sentence-level embeddings (Reimers & Gurevych, 2019). We write $\overline{\Phi}(x) = \Phi(x)/(\|\Phi(x)\|_2 + \varepsilon)$ for the normalized form with a small constant ε. Two distances then drive the retention decision

$$d_i^{\text{red}} = 1 - \overline{\Phi}(R_i)^\top \overline{\Phi}(H_{s_i}), \quad d_i^{\text{surp}} = 1 - \overline{\Phi}(R_i)^\top \overline{\Phi}(\widehat{R}_i). \tag{1}$$

The redundancy distance d_i^{red} measures the increment of the observation beyond the latest recorded state. The surprise distance d_i^{surp} measures its departure from the expected state.

Adaptive Fidelity Assignment. We then assign the retention decision z_i by thresholding both distances. An observation is assigned ***merge*** when $d_i^{\text{red}} < \tau_{\text{red}}$, ***key*** when $d_i^{\text{red}} \geq \tau_{\text{red}}$ and $d_i^{\text{surp}} < \tau_{\text{surp}}$, and ***full*** otherwise. Here τ_{red} and τ_{surp} denote the redundancy and surprise thresholds. A merge event extends an existing state without introducing new visual evidence, a key event is represented by its key-frame feature, and a full event keeps the complete segment tensor. The event tuple and its decision are then passed to EETC.

3.2 ENTITY-CENTRIC EVENT TRAJECTORY CONSTRUCTION

SAO Event Nodes and Entity Indexing. The semantic surprise-guided event filtering stage decides what is retained. The next stage decides how retained events become a queryable structure, and this structure is related to prior structured representations of actions and dynamic scene relations (Ji et al., 2020; Cong et al., 2021). We maintain one temporally ordered chain for every registered entity and collect these chains in the chain memory

$$\mathcal{M}^{(i)} = \left\{ e \mapsto \mathcal{T}_e^{(i)} : \; n_{e,1} \xrightarrow{\Delta t_{e,1}} n_{e,2} \xrightarrow{\Delta t_{e,2}} \cdots \xrightarrow{\Delta t_{e,L_e-1}} n_{e,L_e} \right\}_{e \in \mathcal{E}^{(i)}}. \tag{2}$$

The entity identity serves as the access key, so later retrieval follows entities rather than stream positions. The link gap $\Delta t_{e,j} = t^s_{e,j+1} - t^e_{e,j}$ records the elapsed time between two consecutive events of entity e. A trajectory link therefore exposes both the order and the spacing of events along one timeline. Each retained event becomes a node

$$n_i = (\mathbf{g}_i,\, R_i,\, \mathcal{I}_i,\, \ell_i,\, F_i,\, z_i)\,, \tag{3}$$

that binds the event semantics, its visual evidence, and its retention decision into one unit. We later select among these same units at query time. Here $\mathbf{g}_i$ holds the SAO triple and R_i holds the canonical relation. The interval $\mathcal{I}_i$ locates the event in the stream. The pair (ℓ_i, F_i) carries the payload level and the payload itself, and z_i records the SSEF decision. We attach the node to the chain of its subject s_i. The object o_i preserves the interacting entity, and a match with another registered entity creates an interaction link toward that entity's chain. Trajectory links join consecutive events of one entity, and interaction links stitch the entity chains together. Every retained event is therefore reachable along its own timeline or across the entities it interacts with.

Incremental Trajectory Update. The chain is written incrementally as events arrive. The visual payload F_i follows the SSEF decision and takes either the key-frame feature $\mathbf{X}_i^{\text{key}}$ or the complete segment tensor $\mathbf{X}_i^{\text{full}}$. A ***merge*** decision creates no new node and instead extends the interval of the latest node on the subject chain. A ***key*** or ***full*** decision appends the node to the tail of that chain. The chain update is

$$\mathcal{T}_{s_i}^{(i)} = \begin{cases} \text{Extend}\Big(\mathcal{T}_{s_i}^{(i-1)},\, [t^s_{i'}, \max(t^e_{i'}, t^e_i)]\Big)\,, & z_i = \textbf{\textit{merge}}, \\ \mathcal{T}_{s_i}^{(i-1)} \oplus n_i, & z_i \in \{\textbf{\textit{key}}, \textbf{\textit{full}}\}, \end{cases} \tag{4}$$

where i' denotes the latest node on the subject chain and $\oplus$ appends a node chronologically. A subject without an existing chain opens a new one. The recorded state H_{s_i} moves to R_i only when a new node is created, so redundant observations keep comparing against the same recorded state. Each update therefore touches only the chain of the current subject while all unrelated chains stay unchanged. The merge rule consolidates repeated observations into continuous temporal states. Informative transitions stay individually addressable through their SAO semantics and timestamps together with their visual evidence. Recent frames and dialogue turns are kept alongside the chains as auxiliary context for query-time assembly.

Storage Efficiency Analysis. The event-centered representation changes the dominant storage term from the number of processed frames to the number of entities and their semantic transitions. Let L denote the number of processed frames, D the dense visual-feature dimension, N the number of registered entities, and $\overline{K}$ the average number of retained semantic changes per entity. A flat frame memory follows $M_{\text{flat}} = \mathcal{O}(LD)$, whereas the entity-centric event trajectory construction stores lightweight event records together with only the visual payloads selected by the semantic surprise-guided event filtering:

$$M_{\text{EETC}} = \mathcal{O}\big(N\,\overline{K}\,d_g + N_{\text{key}} D_{\text{key}} + N_{\text{full}} D_{\text{full}}\big)\,, \tag{5}$$

where $d_g \ll D$ is the dimension of a symbolic event record, and N_{key} and N_{full} count key-frame and full-tensor payloads, respectively. Since semantic surprise-guided event filtering merges repeated observations and allocates full tensors only to surprising transitions, $N\overline{K}$ is typically much smaller than L, while the heterogeneous payload terms remain explicitly controlled by the retention policy. Thus, the expression explains why EETC reduces frame-proportional storage growth by retaining semantic transitions and heterogeneous visual payloads instead of every observed frame.

3.3 INTENTION-DRIVEN RESOURCE-AWARE ROUTER

Intent Decoupling and Candidate Retrieval. At query time the binding resource is the answer context, so evidence must be selected rather than loaded in full, consistent with retrieval-based context construction and memory-aware inference (Lewis et al., 2020; Li et al., 2024). When a question q arrives we first parse it into $\boldsymbol{\rho}_q = \big(\mathcal{E}_q,\, \eta_q^{\text{temp}},\, \eta_q^{\text{vis}},\, \eta_q^{\text{hist}}\big)$. Here $\mathcal{E}_q$ collects the queried entities. The indicator η_q^{temp} expresses the demand for temporal relations, η_q^{vis} the demand for fine-grained visual evidence, and η_q^{hist} the demand for dialogue history. We access the chains of the queried entities and locate the SAO nodes relevant to the question. Neighboring nodes are added when temporal evolution is required, and object references lead to the chains of interacting entities. If no entity is identified we fall back to relevance matching over the complete entity index. The result is a candidate node set $\mathcal{C}_q$.

Multi-Fidelity Evidence Options. Each candidate can be exposed to the answer model only at a fidelity that SSEF retained. A ***key*** can be represented through structured text and key-frame feature. A ***full*** node additionally offers its complete tensor and supports detailed visual verification. We include omission as an explicit zero-cost option. For the omission option $h = \varnothing$, we set $U_{j,\varnothing}(q) = 0$, $c^{\text{vis}}_{j,\varnothing} = 0$, and $c^{\text{text}}_{j,\varnothing} = 0$. Lowering fidelity is always possible, but visual evidence discarded by SSEF can never be recovered. Let $U_{j,h}(q)$ be the utility of exposing node n_j at fidelity h, combining the semantic relevance of the event, its contribution to the required trajectory, and the match between fidelity and parsed demand. The visual-token and text-token costs of this exposure are $c^{\text{vis}}_{j,h}$ and $c^{\text{text}}_{j,h}$. The answer context window is first reserved for the required dialogue history and recent frames. We then select exactly one access option per candidate under the remaining budgets B_q^{vis} and B_q^{text}:

$$\max_{\{x_{j,h}\}} \sum_{n_j \in \mathcal{C}_q} \sum_{h \in \mathcal{H}_j} U_{j,h}(q)\, x_{j,h} \quad \text{s.t.} \sum_{n_j \in \mathcal{C}_q} \sum_{h \in \mathcal{H}_j} c^{\text{vis}}_{j,h} x_{j,h} \le B_q^{\text{vis}},\ \sum_{n_j \in \mathcal{C}_q} \sum_{h \in \mathcal{H}_j} c^{\text{text}}_{j,h} x_{j,h} \le B_q^{\text{text}}, \tag{6}$$

where h ranges over the fidelities available to node n_j, $x_{j,h} \in \{0,1\}$ and $\sum_{h \in \mathcal{H}_j} x_{j,h} = 1$. For a ***key*** node, $\mathcal{H}_j$ contains omission, structured text, and the key-frame feature. For a ***full*** node, it additionally contains the full segment tensor. The equality constraint therefore assigns exactly one option to every candidate, while omission allows the router to exclude a node without consuming either budget.

Budgeted Routing and Runtime Protection. Exact solving is impractical at query time, so we allocate greedily by query-conditioned utility. Each candidate starts from its most informative available option. When either budget is exceeded, IRR downgrades lower-priority candidates from full features to key-frame features, structured text, or omission. Because omission has zero cost, the final assignment remains feasible under both budgets. The two budgets regulate visual and textual evidence separately. A VRAM Guard then measures the runtime footprint of the assembled context and compresses visual features through dynamic pooling or truncates lower-priority text until the context fits device memory. In this way, intention-driven resource-aware router realizes the same resource-constrained objective online while preserving the option to fall back to lower-cost evidence when the available budget changes.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Implementation Details. We implement TRACE using LongVA-7B-DPO as the long context vision language backbone for visual understanding and response generation. A frozen Qwen2.5-7B-Instruct model predicts the semantic evolution of incoming events, while all-MiniLM-L6-v2 provides sentence-level embeddings for semantic comparison and evidence retrieval. All pretrained components remain frozen throughout evaluation; TRACE therefore requires neither task-specific fine-tuning nor additional training on the evaluated benchmarks. Frames are processed causally in temporal order, and the model can access only observations received before each query. Unless otherwise specified, we use a streaming interval of $1s$ and set the redundancy and surprise thresholds to $\tau_{\text{red}} = 0.35$ and $\tau_{\text{surp}} = 0.85$ respectively. We rerun all compared methods on the same evaluation instances under matched input and output settings, and rescore their generated answers with the same judge described in Appendix B.2. All experiments are conducted on a server equipped with 8 $\times$ NVIDIA RTX 4090 GPUs.

Table 1: Model's Performance on StreamBench. SF, SM, LM, CI, KG, and OS denote Simple Factual, Short-term Memory, Long-term Memory, Contextual Inference, Knowledge-Grounded, and Object Search, respectively. Acc. denotes accuracy in percent, and Sco. denotes the answer-quality score. Bold denotes the best reported value. Avg. is computed over all StreamBench questions, weighting each task category by its number of evaluated questions.

Method	SF		SM		LM		CI		KG		OS		Avg	
	Acc.	Sco.	Acc.	Sco.	Acc.	Sco.	Acc.	Sco.	Acc.	Sco.	Acc.	Sco.	Acc.	Sco.
Offline														
MovieChat	35.8	2.05	26.5	1.76	20.4	1.42	42.3	2.28	67.2	3.39	18.6	1.45	34.9	2.05
LongVU	74.2	3.83	59.9	3.27	59.1	3.23	70.7	3.69	63.1	3.20	53.9	2.93	63.4	3.36
Online + Training														
VideoLLM-online	64.1	3.39	52.9	3.01	48.8	2.87	62.7	3.31	69.2	3.58	41.4	2.61	56.4	3.13
Flash-Vstream	59.2	3.17	48.6	2.78	44.5	2.64	58.1	3.13	66.4	3.34	37.1	2.38	52.2	2.90
StreamForest	74.8	3.84	62.0	3.35	60.1	3.31	72.4	3.85	71.1	3.41	54.2	2.94	65.7	3.45
TimeChat-Online	70.2	3.56	55.8	3.08	51.8	2.91	65.3	3.47	72.2	3.44	46.2	2.66	60.1	3.19
StreamBridge	72.5	3.76	59.6	3.30	52.5	2.90	74.4	3.84	67.8	3.32	45.8	2.66	62.0	3.30
Dispider	74.8	3.75	55.5	2.97	50.5	2.85	66.7	3.50	70.8	3.46	49.5	2.29	61.1	3.13
Online + Training Free														
ReKV	78.2	**3.98**	63.1	3.40	62.7	3.42	75.1	3.98	83.9	4.09	56.5	3.05	69.7	3.64
StreamChat	68.8	3.56	56.1	3.15	52.9	2.87	67.6	3.82	87.9	4.37	50.5	2.93	63.8	3.44
TRACE	**78.8**	3.97	**73.8**	**3.71**	**69.3**	**3.65**	**76.8**	**4.04**	**88.3**	**4.39**	**70.9**	**3.56**	**76.2**	**3.88**

Table 2: Comprehensive cross-benchmark evaluation. Accuracy scores are reported across diverse video understanding tasks. OVBench, SVBench, and StreamBench are evaluated under the streaming protocol, whereas MSVD, MSRVTT, Video-MME, and NExT-QA are evaluated under the offline single-turn protocol. Avg. represents the arithmetic mean of the scores from the seven datasets.

Method	MSVD		MSRVTT		Video-MME		NExT-QA		OVBench		SVBench		StreamBench		Avg	
	Acc.	Sco.	Acc.	Sco.	Acc.	Sco.	Acc.	Sco.	Acc.	Sco.	Acc.	Sco.	Acc.	Sco.	Acc.	Sco.
Offline																
MovieChat	57.9	2.97	43.0	2.15	45.0	2.30	35.6	2.05	25.3	1.34	26.8	1.98	34.9	2.05	38.4	2.12
LongVU	62.5	3.18	51.8	2.31	52.1	2.32	54.6	2.78	48.0	2.33	54.2	2.29	63.4	3.36	55.2	2.65
Online + Training																
VideoLLM-online	54.2	2.87	38.2	2.06	41.6	2.11	35.8	2.03	32.2	1.98	40.3	2.24	56.4	3.13	42.7	2.35
Flash-Vstream	56.1	2.91	39.8	2.08	42.7	2.15	36.1	2.06	35.7	1.58	36.2	2.09	52.2	2.90	42.7	2.25
StreamForest	65.1	3.39	50.6	2.47	59.9	2.73	57.3	2.95	60.5	2.53	63.2	3.13	65.7	3.45	60.3	2.95
TimeChat-Online	68.8	3.46	50.2	2.52	58.5	2.47	53.2	2.50	49.8	2.43	59.7	2.86	60.1	3.19	57.2	2.77
StreamBridge	70.5	3.54	53.2	2.67	57.9	2.60	56.2	2.82	58.7	2.71	59.7	2.60	62.0	3.30	59.7	2.89
Dispider	58.4	3.13	50.6	2.48	54.3	2.62	56.3	2.54	46.7	2.53	56.3	2.44	61.1	3.13	54.8	2.70
Online + Training Free																
ReKV	60.5	3.04	54.1	2.68	54.8	2.91	62.3	3.15	50.5	2.52	63.7	3.35	69.7	3.64	59.4	3.04
StreamChat	58.7	3.08	43.4	2.38	43.8	2.21	50.5	2.84	45.3	2.36	55.9	2.86	63.8	3.44	51.6	2.74
TRACE	**73.2**	**3.66**	**55.8**	**2.81**	**63.7**	**2.97**	**62.4**	**3.22**	**62.8**	**2.92**	**64.9**	**3.42**	**76.2**	**3.88**	**65.6**	**3.27**

Benchmark Protocol. StreamBench v0.3 (Xiong et al., 2025) serves as our primary benchmark for causal streaming video understanding, contains 307 videos and 1,838 question instances. We evaluate models under the same protocol as the streaming video understanding method StreamChat (Xiong et al., 2025), which evaluates six complementary capabilities: simple factual retrieval (SF), short-term memory (SM), long-term memory (LM), contextual inference (CI), knowledge-grounded question answering (KG), and object search (OS). These tasks jointly assess online perception, event retention, temporal reasoning and multi-turn question answering. To evaluate its generalization beyond StreamBench, we further consider MSVD (Chen & Dolan, 2011), MSRVTT (Xu et al., 2016), Video-MME (Fu et al., 2025), NExT-QA (Xiao et al., 2021), SVBench (Yang et al., 2025b) and OVBench (Huang et al., 2025). These benchmarks cover both online and conventional offline video question answering under diverse temporal scales and reasoning requirements. Additional details of the automatic judge, including its model, prompt and scoring protocol are provided in Appendix B.2.

4.2 Main Results

Performance on StreamBench. Table 1 reports the performance on StreamBench. TRACE achieves an average accuracy of 76.2% and an answer quality score of 3.88. Relative to StreamChat, it improves accuracy by 10.0, 17.7, 16.4, 9.2, 0.4 and 20.4 percentage points on SF, SM, LM, CI, KG and OS, respectively. The largest improvements are observed in OS, SM, and LM, which require evidence to be associated across different time intervals. This pattern is consistent with the intended scope of entity-centric retrieval. TRACE also exceeds StreamChat and ReKV in answer quality

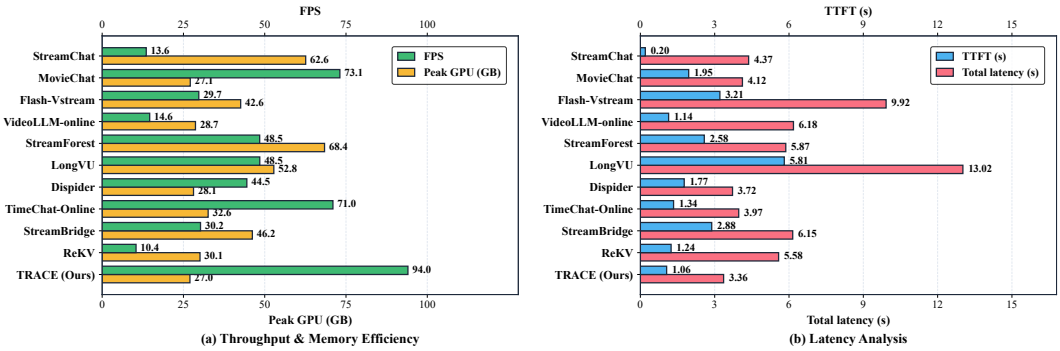


Figure 3: Efficiency comparison on streaming video understanding. (a) FPS and peak GPU memory usage of the evaluated methods. (b) Time to first token (TTFT) and total response latency.

by 0.44 and 0.24 points, respectively. These results indicate that entity-centric event chains and intention-driven evidence selection are particularly effective for questions requiring delayed recall and long-range temporal association.

Generalization across Benchmarks. Table 2 compares TRACE with offline, trained online, and training-free online approaches across seven benchmarks. TRACE obtains the highest average accuracy and answer-quality score, reaching 65.6% and 3.27, respectively. Compared with StreamForest, the strongest baseline in average accuracy, TRACE improves accuracy by 5.3 percentage points. It also exceeds ReKV, the strongest baseline in average answer quality, by 0.23 points. The improvement remains consistent across evaluation settings. On the streaming benchmarks OVBench, SVBench, and StreamBench, TRACE achieves 62.8%, 64.9%, and 76.2% accuracy, outperforming the strongest corresponding baselines by 2.3, 1.2, and 6.5 percentage points. On the offline benchmarks MSVD, MSRVTT, Video-MME, and NExT-QA, it reaches 73.2%, 55.8%, 63.7%, and 62.4%, respectively. The gains across both protocols show that TRACE does not depend on a benchmark-specific streaming format. Its event-centric memory remains effective when questions involve short clips, long temporal dependencies, or continuously arriving observations.

Efficiency under Resource Constraints. Figure 3 evaluates the accuracy-efficiency trade-off on StreamBench. TRACE achieves the highest accuracy while reporting the lowest total response latency, the highest processing throughput, and the lowest peak GPU memory among the compared methods. It reaches 76.2% accuracy with a total latency of 3.36 s, a throughput of 94.05 FPS, and a peak memory footprint of 27.0 GB. Its time to first token is 1.06 s, which is not the lowest reported value but remains substantially below several long-video baselines. Compared with ReKV, the strongest training-free baseline in StreamBench accuracy, TRACE improves accuracy by 6.5 percentage points, reduces total latency by 2.22 s, and lowers peak memory by 3.1 GB. Its throughput increases from 10.42 to 94.05 FPS. Compared with StreamChat, TRACE improves accuracy by 12.4 percentage points while reducing total latency from 4.37 to 3.36 s and peak memory from 62.6 to 27.0 GB. These comparisons show that TRACE improves long-range understanding while using selective visual retention and bounded query-time evidence budgets.

4.3 Ablation Study

Figure 4 reports cumulative ablations of SSEF, EETC and IRR. The complete system improves average StreamBench accuracy from 61.1% for the LongVA-7B-DPO backbone to 76.2%. Because the modules are introduced cumulatively, each comparison measures the additional contribution of a module given all preceding components, rather than its isolated effect.

SSEF: The Compression-Detail Trade-off. Adding the semantic surprise-guided event filtering increases average accuracy from 61.1% to 63.4%. The largest task-level gains appear in OS and CI, which improve by 6.1 and 6.0 percentage points, followed by LM with a 4.1-point gain. SF also improves by 2.0 points. In contrast, SM decreases by 3.3 points and KG by 0.8 points, indicating that predictive retention alone cannot preserve every form of short-range or knowledge-dependent evidence. SSEF introduces additional prediction and semantic-comparison costs: total latency

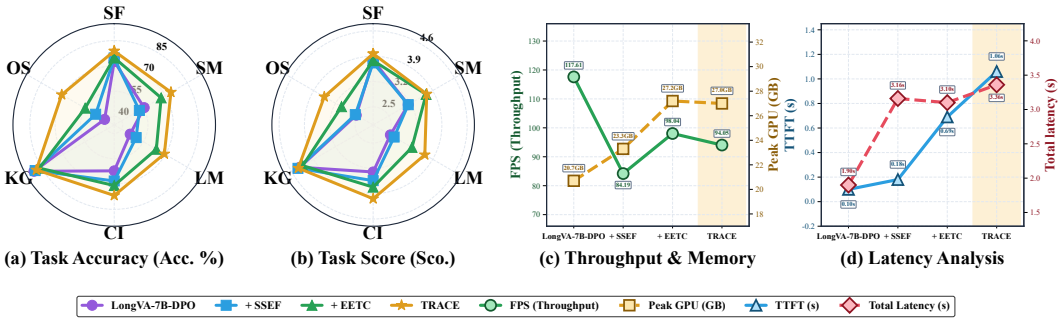


Figure 4: Cumulative ablation analysis of TRACE. The radar charts report (a) task accuracy and (b) answer-quality score across six StreamBench tasks. The line plots compare (c) throughput and peak GPU memory and (d) time to first token (TTFT) and total response latency for LongVA-7B-DPO, the cumulative SSEF and EETC configurations, and TRACE.

increases from 1.90 to 3.16 s, throughput decreases from 117.61 to 84.19 FPS, and peak memory rises from 20.7 to 23.3 GB. The improvement therefore reflects more selective event retention rather than a cost-free compression mechanism.

EETC: Entity-Centric Temporal Organization. Adding the entity-centric event trajectory construction on top of SSEF raises average accuracy from 63.4% to 69.5%. The largest incremental gains occur in SM and LM, which improve by 14.9 and 13.8 percentage points, respectively. OS improves by 7.2 points, while CI gains another 2.7 points. These task-level changes are consistent with the intended role of EETC, although the cumulative ablation does not isolate its effect from the preceding SSEF component. In this cumulative configuration, the structured organization is accompanied by an increase in throughput from 84.19 to 98.04 FPS and slightly reduces total latency from 3.16 to 3.10 s, although peak memory increases from 23.3 to 27.2 GB because additional event nodes and visual references are maintained.

IRR: Query-Conditioned Evidence Allocation. The complete TRACE system adds the intention-driven resource-aware router and improves average accuracy from 69.5% to 76.2%. IRR produces the largest incremental gain on OS, increasing accuracy by 16.0 percentage points. It also improves SM, CI, LM, and SF by 6.7, 5.9, 5.5, and 4.0 points, respectively. KG recovers by 2.1 points but remains below the backbone-only result, suggesting that intention-driven retrieval primarily benefits visually grounded and temporally distributed questions. The additional routing stage increases total latency from 3.10 to 3.36 s and reduces throughput from 98.04 to 94.05 FPS, while peak memory decreases slightly from 27.2 to 27.0 GB. Thus, the cumulative addition of IRR improves accuracy with a modest routing overhead and no increase in peak memory.

5 Conclusion

This paper presents TRACE, a training-free online framework for long-range video understanding with resource-aware retention and retrieval. We organize streaming memory into queryable chains of evidence that preserve temporal relations while controlling storage and inference costs. During ingestion, the semantic surprise-guided event filtering compares each observation with its predicted semantic state, merges repeated content, and retains informative changes at different visual granularities. The entity-centric event trajectory construction then parses retained observations into subject-action-object event nodes and organizes them into entity-centric temporal chains. At query time, the intention-driven resource-aware router retrieves and selects query-relevant evidence under visual and textual budgets through a budgeted multiple-choice selection problem. Experiments on StreamBench and six complementary video benchmarks demonstrate strong long-range video understanding, efficient streaming processing, and bounded memory consumption.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Saeed Samangooei, Maria Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Rodrigo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

David L. Chen and William B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.

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

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, Ethan He, Hongxu Yin, Pavlo Molchanov, Jan Kautz, Linxi Fan, Yuke Zhu, Yao Lu, and Song Han. Longvila: Scaling long-context visual language models for long videos. In *International Conference on Learning Representations*, 2024b.

Meng Chu, Yicong Li, and Tat-Seng Chua. Understanding long videos via llm-powered entity relation graphs. In *Proceedings of the ACM International Conference on Multimedia*, 2025.

Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-temporal transformer for dynamic scene graph generation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

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

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. Online video understanding: Ovbench and videochat-online. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. Infinipot-v: Memory-constrained kv cache compression for streaming video understanding. In *Advances in Neural Information Processing Systems*, 2025.

Seon-Ho Lee, Jue Wang, Zhikang Zhang, David Fan, and Xinyu Li. Video token merging for long-form video understanding. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, Yu Qiao, Yali Wang, and Limin Wang. Videochat-flash: Hierarchical compression for long-context video modeling. In *International Conference on Learning Representations*, 2026.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Ke Ma, Jiaqi Tang, Bin Guo, Xueting Han, Ruonan Xu, Qingfeng He, Ziheng Wang, Xu Wang, Qifeng Chen, Zhiwen Yu, and Yunhao Liu. Response-g1: Explicit scene graph modeling for proactive streaming video understanding. In *Proceedings of the Association for Computational Linguistics*, 2026.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. 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)*, 2024.

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

Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, 1999.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

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. In *Proceedings of Machine Learning Research*, volume 267, pp. 54582–54599, 2025.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. Moviechat: From dense token to sparse memory for long video understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Haibo Wang, Bo Feng, Zhengfeng Lai, Mingze Xu, Shiyu Li, Weifeng Ge, Afshin Dehghan, Meng Cao, and Ping Huang. Streambridge: Turning your offline video large language model into a proactive streaming assistant. In *Advances in Neural Information Processing Systems*, 2025a.

Yuxuan Wang, Yiqi Song, Cihang Xie, Yang Liu, and Zilong Zheng. Videollamb: Long streaming video understanding with recurrent memory bridges. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b.

Siwei Wen, Zhangcheng Wang, Xingjian Zhang, Lei Huang, and Wenjun Wu. Eventmemagent: Hierarchical event-centric memory for online video understanding with adaptive tool use. *arXiv preprint arXiv:2602.15329*, 2026.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: a benchmark for long-context interleaved video-language understanding. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, 2024.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

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

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*, 2024.

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

Zhenyu Yang, Yuhang Hu, Zemin Du, Dizhan Xue, Shengsheng Qian, Jiahong Wu, Fan Yang, Weiming Dong, and Changsheng Xu. SVBench: A benchmark with temporal multi-turn dialogues for streaming video understanding. In *The Thirteenth International Conference on Learning Representations*, 2025b.

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: 80% visual tokens are naturally redundant in streaming videos. In *Proceedings of the ACM International Conference on Multimedia*, 2025.

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. In *Advances in Neural Information Processing Systems*, 2025.

Hang Zhang, Xin Li, and Lidong Bing. 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*, 2023.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-VStream: Efficient real-time understanding for long video streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025a.

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

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *Transactions on Machine Learning Research*, 2025b.

A Additional Method Details

A.1 Streaming Event Filtering

Algorithm 1 expands the SSEF decision in Section 3.1 into an online update. The input is the sequence of action-delimited interactions, not individual decoded frames. For each interaction, the semantic representation R_i is compared with the latest recorded state H_{s_i} and the expected state $\widehat{R}_i = \Phi_{\text{pred}}(H_{s_i}, \ell_i, o_i)$. The distances and thresholds are exactly those in Equation 1. A subject without a recorded state starts a new chain with a *key* node; subsequent interactions follow the two-distance decision. A *merge* extends the latest node interval without updating H_{s_i}. Both *key* and *full* create a node and set $H_{s_i} \leftarrow R_i$, as specified by Equation 4.

Algorithm 1 Online semantic surprise-guided event filtering and trajectory update.

Require: Incoming interactions $\{(\mathbf{g}_i, \mathcal{I}_i, \ell_i, \mathbf{X}_i^{\text{key}}, \mathbf{X}_i^{\text{full}})\}$; thresholds $\tau_{\text{red}}, \tau_{\text{surp}}$
Ensure: Updated chain memory $\mathcal{M}^{(i)}$ and subject states H_{s_i}

for each incoming interaction i in temporal order **do**
  Construct the canonical relation R_i from $\mathbf{g}_i, \mathcal{I}_i, \ell_i$.
  if s_i has no recorded state **then**
    Set $z_i \leftarrow$ *key*.
  else
    Set $\widehat{R}_i \leftarrow \Phi_{\text{pred}}(H_{s_i}, \ell_i, o_i)$.
    Compute d_i^{red} and d_i^{surp} by Equation 1.
    if $d_i^{\text{red}} < \tau_{\text{red}}$ **then**
      Set $z_i \leftarrow$ *merge*.
    else if $d_i^{\text{surp}} < \tau_{\text{surp}}$ **then**
      Set $z_i \leftarrow$ *key*.
    else
      Set $z_i \leftarrow$ *full*.
    end if
  end if
  if $z_i =$ *merge* **then**
    Extend the latest node interval on $\mathcal{T}_{s_i}^{(i-1)}$; retain H_{s_i}.
  else
    Set $F_i \leftarrow \mathbf{X}_i^{\text{key}}$ if $z_i =$ *key*, else $F_i \leftarrow \mathbf{X}_i^{\text{full}}$.
    Append $n_i = (\mathbf{g}_i, R_i, \mathcal{I}_i, \ell_i, F_i, z_i)$ to $\mathcal{T}_{s_i}^{(i-1)}$.
    Set $H_{s_i} \leftarrow R_i$ and update the entity index in $\mathcal{M}^{(i)}$.
  end if
end for

A.2 Query-Conditioned Evidence Assembly

The router converts a question into $\boldsymbol{\rho}_q = (\mathcal{E}_q, \eta_q^{\text{temp}}, \eta_q^{\text{vis}}, \eta_q^{\text{hist}})$, as defined in Section 3.3. Entity matches first retrieve the relevant trajectories. Temporal demand expands the candidate set $\mathcal{C}_q$ to neighboring nodes, while interaction links expose related entities. The router then assigns one available fidelity to each node under the separate visual and text budgets in Equation 6. A node retained only as *key* cannot later supply a full tensor. The answer context combines the selected evidence with reserved dialogue turns and recent frames. The thesis prompt design motivates separating intent parsing from answer generation. For the current notation, the intended parsing output is schematically expressed as follows, this is an explanatory template, not a claim about the exact prompt bytes used by every backbone.

```
[Question] {q}
Identify the entities mentioned in the question.  Decide
whether answering requires temporal relations, fine-grained
visual evidence, or earlier dialogue.  Return the intent
fields below without adding unsupported entities.
entities:  {E_q}
```

```
temporal demand: {eta_temp}
visual demand: {eta_vis}
history demand: {eta_hist}
```

The following schematic shows the evidence order supplied to the answer model. It likewise describes the current TRACE representation rather than a verbatim model-specific prompt.

```
[Instruction] Answer from the supplied video evidence. Do
not invent missing events.
[Query intent] entities: {E_q}; temporal demand: {eta_temp};
visual demand: {eta_vis}; history demand: {eta_hist}.
[Entity trajectories] {selected SAO nodes, intervals, and
interaction links}.
[Visual evidence] {selected key frames or full segment
features}.
[Dialogue history] {required earlier turns}.
[Recent frames] {eligible frames before the query}.
[Question] {q}
```

The selection is made before generation. If the assembled context exceeds available device memory, the VRAM Guard reduces lower-priority visual or textual evidence as described in Section 3.3, it cannot restore evidence discarded at ingestion.

B Evaluation Protocol

B.1 Causal StreamBench Evaluation

StreamBench v0.3 (Xiong et al., 2025) is evaluated at its question breakpoints. At each breakpoint, TRACE has access only to the video prefix and dialogue turns available at that time. Future frames and later questions are excluded. We retain the six task categories used in the main text: simple factual retrieval (SF), short-term memory (SM), long-term memory (LM), contextual inference (CI), knowledge-grounded question answering (KG), and object search (OS). The evaluation set contains 302 SF, 339 SM, 303 LM, 297 CI, 298 KG, and 299 OS questions, totaling 1,838 question instances. The same causal restriction applies to the supplementary StreamBench analyses below.

B.2 Automatic Answer-Quality Judge

The original protocols of MSVD, MSRVTT, Video-MME, and NExT-QA do not share a five-point answer-quality measure. We follow each benchmark's input and answer-generation protocol, but compute every Acc. and Sco. value reported in our comparison tables from the rerun outputs using the same semantic judge. The Acc. and Sco. columns in our comparison tables use this common judge. The quality score is a supplementary cross-benchmark measure, not an official metric of those benchmarks. The judge is Meta-Llama-3-8B-Instruct, loaded with 4-bit NF4 quantization, double quantization, and float16 computation. Generation uses `do_sample=False` and `max_new_tokens=64`. For a question q_i, reference answer a_i, and prediction $\widehat{a}_i$, the judge returns `llama_pred` in {`yes`, `no`} and an integer `score` from 0 to 5. Semantic paraphrases count as matches. Over N judged examples, we report

$$\text{Acc.} = \frac{100}{N}\sum_{i=1}^{N}\mathbb{I}[\texttt{llama_pred}_i = \texttt{yes}], \quad \text{Sco.} = \frac{1}{N}\sum_{i=1}^{N}\texttt{score}_i. \tag{7}$$

The prompt, checkpoint, decoding settings, parsing rule, and aggregation procedure are held fixed for all reported comparisons. Scores are first computed for each valid question instance and then aggregated according to the dataset-level rule specified in Table 1 and Table 2. The scorer uses the following system message:

> You are an intelligent chatbot designed for evaluating the correctness of generative outputs for question-answer pairs. Your task is to compare the predicted answer with the correct answer and determine if they match meaningfully. Here's how you can accomplish the task: —— INSTRUCTIONS: Focus on the meaningful match between the predicted answer

Table 3: Efficiency and performance comparison on streaming video understanding. TTFT denotes Time to First Token, Total denotes total latency, FPS denotes frames per second, and Peak GPU denotes peak GPU memory usage in gigabytes. Bold denotes the best reported value.

Method	Latency		FPS	Peak GPU	Accuracy
	TTFT	Total			
StreamChat	**0.20s**	4.37s	13.56	62.6GB	63.8
MovieChat	1.95s	4.12s	73.12	27.1GB	34.9
Flash-Vstream	3.21s	9.92s	29.72	42.6GB	52.2
VideoLLM-online	1.14s	6.18s	14.62	28.7GB	56.4
StreamForest	2.58s	5.87s	48.45	68.4GB	65.7
LongVU	5.81s	13.02s	48.48	52.8GB	63.4
Dispider	1.77s	3.72s	44.50	28.1GB	61.1
TimeChat-Online	1.34s	3.97s	70.98	32.6GB	60.1
StreamBridge	2.88s	6.15s	30.25	46.2GB	62.0
ReKV	1.24s	5.58s	10.42	30.1GB	69.7
TRACE	1.06s	**3.36s**	**94.05**	**27.0GB**	**76.2**

Table 4: Cumulative task-level ablation on StreamBench. Acc. is accuracy in percent and Sco. is the answer-quality score. Each row adds the checked component to the preceding configuration. Bold marks the best value in each metric column.

Method	SSEF	EETC	IRR	SF		SM		LM		CI		KG		OS	
				Acc.	Score	Acc.	Score	Acc.	Score	Acc.	Score	Acc.	Score	Acc.	Score
LongVA-7B-DPO	×	×	×	72.6	3.79	55.5	3.12	45.9	2.55	62.2	3.30	**89.9**	4.34	41.6	2.54
+ SSEF	✓	×	×	74.6	3.72	52.2	3.12	50.0	2.67	68.2	3.54	89.1	**4.40**	47.7	2.57
+ EETC	✓	✓	×	74.8	3.78	67.1	3.69	63.8	3.24	70.9	3.72	86.2	4.25	54.9	3.01
TRACE	✓	✓	✓	**78.8**	**3.97**	**73.8**	**3.71**	**69.3**	**3.65**	**76.8**	**4.04**	88.3	4.39	**70.9**	**3.56**

> and the correct answer. Consider synonyms or paraphrases as valid matches. Evaluate the correctness of the prediction compared to the answer.

The user message is:

> Please evaluate the following video-based question-answer pair:
> Question: {`question`}
> Correct Answer: {`answer`}
> Predicted Answer: {`prediction`}
> Provide your evaluation only as a yes/no and score where the score is an integer value between 0 and 5, with 5 indicating the highest meaningful match. Please generate the response in the form of a Python dictionary string with keys `llama_pred` and `score`, where the value of `llama_pred` is a string of `yes` or `no` and the value of `score` is an integer, not a string. DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION. Only provide the Python dictionary string. For example, your response should look like: {`'llama_pred': 'yes', 'score': 4`}.

The scorer serializes system and user messages in Llama-3 chat format. It extracts the final brace-delimited answer, normalizes Python-style quotation and Boolean tokens, and parses the resulting dictionary. If parsing fails, the historical scorer assigns `no` and score 0. This fallback is stated here because it affects reproducibility of the reported aggregate metrics.

B.3 Efficiency Measurements

Time to first token (TTFT) measures the interval from answer-generation start to the first nonempty streamed output fragment. Total response latency is measured from query arrival to the completion of answer generation. Logging and telemetry export are excluded from the reported latency. Input-processing throughput in frames per second divides the number of decoded segment frames by decoding and preprocessing time. Peak GPU memory is the maximum allocated CUDA memory measured after resetting the device statistics for the evaluated call, for a multi-GPU batch, the per-device peaks are summed. These operational definitions distinguish input-processing throughput from answer-token generation speed.

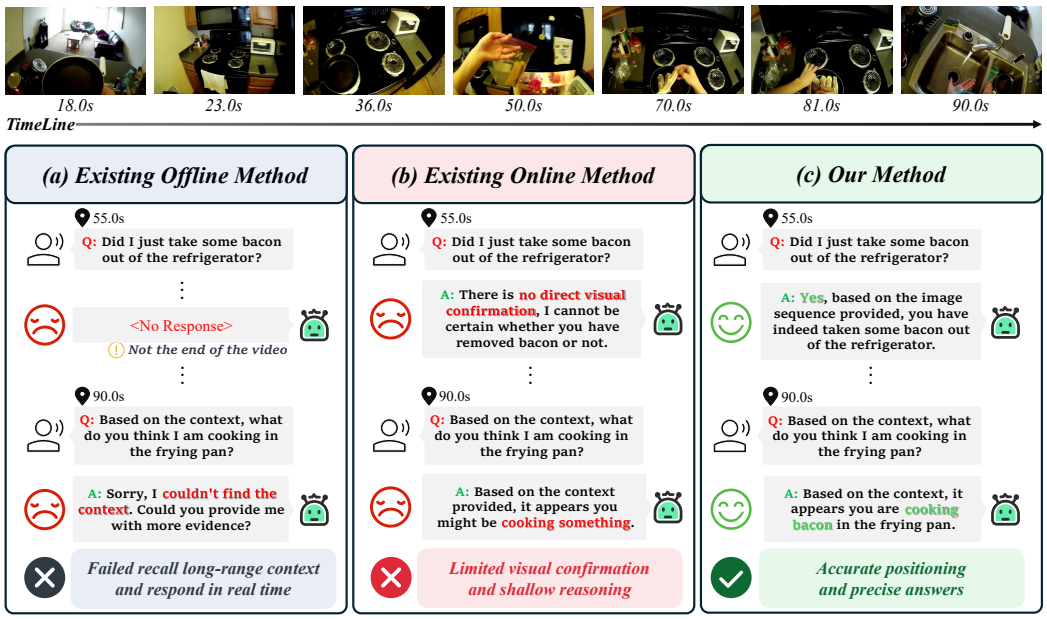


Figure 5: Qualitative case study of long-term delayed recall in a continuous cooking stream. (a) The existing offline method cannot respond during the ongoing stream and fails to recover the earlier context. (b) The existing online method suffers from memory decay, failing to confirm the intermediate event and consequently providing vague reasoning for the final query. (c) Our TRACE retrieves the relevant earlier visual evidence and combines it with the subsequent context to answer both queries.

C Additional Experimental Results

C.1 Efficiency and StreamBench Performance

Table 3 gives the numerical values behind the efficiency comparison in Figure 3. TRACE records the highest StreamBench accuracy and input-processing throughput, with the lowest total response latency and peak GPU memory in this comparison. StreamChat has the shortest TTFT.

C.2 Task-Level Component Ablation

Table 4 expands the cumulative ablation in Section 4.3 by StreamBench task. Adding EETC after SSEF raises short- and long-term memory accuracy from 52.2% to 67.1% and from 50.0% to 63.8%, respectively. Adding IRR produces the largest further gain in object search, from 54.9% to 70.9%. Knowledge-grounded accuracy does not improve over the backbone-only row, so the gains are not uniform across tasks.

C.3 Qualitative Analysis

Figure 5 presents a delayed-recall example from a continuous cooking video. The later question requires the model to recover an earlier visual event and combine it with information established during the intervening dialogue. TRACE associates the relevant event with its entity trajectory and retrieves the full visual evidence from 47.0 s to 52.0 s when detailed confirmation is required. For the final query at 90.0 s, it combines this evidence with the retained dialogue history to infer that the user is cooking bacon in the frying pan. The offline baseline cannot answer during the active stream, while the compared online method cannot confidently confirm the intermediate event and consequently gives a vague final response. This case illustrates how TRACE connects temporally separated visual and conversational evidence without loading the complete observation history.

Table 5: Cumulative component ablation on StreamBench. TTFT and Total are in seconds; Peak GPU is in gigabytes; accuracy is in percent.

Method	SSEF	EETC	IRR	Latency		FPS	Peak GPU	Accuracy
				TTFT	Total			
LongVA-7B-DPO	×	×	×	**0.10s**	**1.90s**	**117.61**	**20.7GB**	61.1
with SSEF	✓	×	×	0.18s	3.16s	84.19	23.3GB	63.4
with EETC	✓	✓	×	0.69s	3.10s	98.04	27.2GB	69.5
TRACE	✓	✓	✓	1.06s	3.36s	94.05	27.0GB	**76.2**

C.4 Component Costs

Table 5 reports the runtime and memory cost of the same cumulative additions. The backbone-only configuration is fastest, whereas TRACE improves accuracy from 61.1% to 76.2% with a total response latency of 3.36 s. The table should be read as incremental system configurations, not as isolated single-module measurements.