STREAMEM: CAN MLLMS UNDERSTAND REAL-WORLD STREAMS WITH LONG-TERM MEMORY?

Anonymous authors
Paper under double-blind review

ABSTRACT

Deployment of multimodal large language models (MLLMs) demands systems that process continuous multimodal streaming inputs at a fixed rate, respond instantly, and retain dynamically evolving long-term memory, capabilities that current MLLMs still largely lack. To rigorously assess such capabilities, we build **StreaMem**, the first benchmark to jointly evaluate MLLMs' real-time processing and long-term memory under a real-streaming protocol, covering four scenarios and 12 subtasks with 210 videos of more than 30 minutes and over 3,000 open-ended QA pairs, together with four metrics. Evaluation of 7 existing models on StreaMem confirms that real-time processing and long-term memory remain unsolved under real-streaming conditions. We then propose **SOAM**, a **S**treaming **O**mni **A**gentic **M**emory that equips MLLMs with streaming understanding and external long-term memory without training. SOAM parses the incoming stream into semantically coherent events via event boundary detection, organizes them into a heterogeneous two-layer memory graph with biometric entity anchoring, routes each question through an LLM Router that decides whether memory retrieval is needed, and retrieves evidence along complementary semantic, entity co-occurrence, and temporal-anchor paths. SOAM achieves state-of-the-art performance among open-source models and methods of comparable scale, with substantial improvements in long-term memory persistence. The code is anonymously available[1], and the dataset will be released upon acceptance.

1 INTRODUCTION

Recently, multimodal large language models (MLLMs) have achieved significant progress across a wide range of vision-language tasks, including image understanding and video question answering (Bi et al., 2025; Lin et al., 2024a; Grauman et al., 2022). As MLLMs move toward real-world deployment, a new set of demands has emerged: systems must receive and process continuous multimodal streaming inputs at a fixed rate, respond instantly without interrupting the incoming stream, and retain reliable long-term memory of events that occurred tens of minutes earlier.

Existing MLLMs, however, are not built for this setting. Mainstream offline models (OpenAI, 2024; Xu et al., 2025a) can only ingest a bounded, pre-recorded video and must restart inference whenever the context is exhausted. Online and streaming models (MiniCPM-o Team, OpenBMB, 2025; Guan et al., 2026; Zhang et al., 2025a; Chen et al., 2024a) maintain only shallow working memory (KV caches or continuously compressed visual tokens), which fades as the stream grows. Memory-augmented agents such as M3-Agent (Long et al., 2026) construct explicit long-term memory, but operate offline on complete videos and rely on costly training. As a result, no existing approach provides reliable long-term memory under real-streaming conditions.

Rigorously evaluating such capabilities is itself challenging: as detailed in Section 3, no existing benchmark supports real-streaming evaluation. We therefore build **StreaMem**, *the first benchmark to jointly evaluate MLLMs' real-time processing and long-term memory under a real-streaming protocol*, in which video frames are fed at a fixed rate and questions are posed at specific time points, requiring the model to draw on its accumulated memory. StreaMem covers four scenarios across 12 subtasks (Figure 1), featuring 210 videos of more than 30 minutes and over 3,000 open-ended QA

[1] https://anonymous.4open.science/r/SOAM-FB1324wrc/

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.

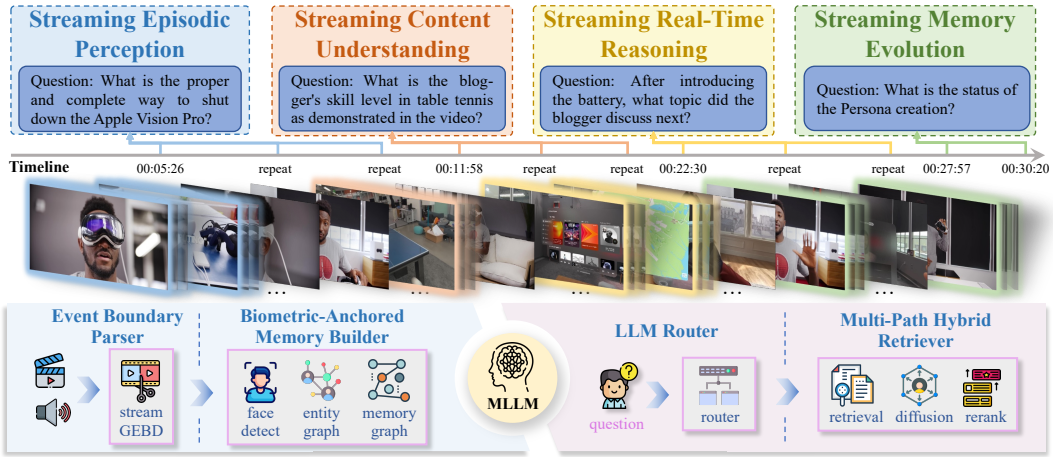


Figure 1: Overview of StreaMem, spanning four streaming scenarios, and SOAM, which continuously builds memory from the incoming stream.

pairs constructed by LLM-based generation followed by human verification. By repeatedly querying the same question at different time points, StreaMem introduces First-mention Accuracy (FMA) and Memory Persistence Time (MPT), complemented by Average Processing Latency (APL) and Average Response Latency (ARL) for real-world usability.

To equip MLLMs with these capabilities, we propose **SOAM**, a **S**treaming **O**mni **A**gentic **M**emory that gives MLLMs streaming understanding and external long-term memory without training. SOAM operates along two asynchronous pathways (Figure 1): a *memory generation pathway* that segments the stream into semantically coherent events and organizes them into a biometric-anchored two-layer memory graph, and a *question answering pathway* in which an *LLM Router* decides whether each question requires memory retrieval and a *multi-path hybrid retriever* gathers evidence along complementary semantic, entity co-occurrence, and temporal-anchor paths for answer synthesis.

We evaluate 7 existing models from proprietary, open-source offline, and open-source online categories. The results reveal systematic capability gaps in dynamic memory updating and anticipatory prediction, and expose a tension between response latency and answer accuracy that pseudo-streaming protocols cannot surface. In contrast, SOAM, built on Qwen2.5-Omni-7B (Xu et al., 2025a), achieves state-of-the-art performance among open-source models and methods of comparable scale, with substantial improvements in long-term memory persistence.

The main contributions of this paper are summarized as follows:

- **Problem.** We identify and formalize the problem of real-time streaming understanding with long-term memory for MLLMs, and show that it is unaddressed on both sides: existing models cannot maintain reliable memory under real-streaming conditions, and existing benchmarks' pseudo-streaming protocols cannot expose such failures.
- **Benchmark.** We build **StreaMem**, the first benchmark to jointly evaluate streaming processing and long-term memory under a real-streaming protocol, covering four scenarios and 12 subtasks with four metrics (FMA, MPT, APL, ARL) and a real-streaming evaluation framework that delivers frames at a fixed rate without waiting for model responses.
- **Method.** We propose **SOAM**, a training-free streaming omni agentic memory built on event-based segmentation, a biometric-anchored two-layer memory graph, LLM-based routing, and multi-path hybrid retrieval, achieving state-of-the-art results among comparable-scale open-source systems. The benchmark and code will be publicly released.

2 Related Work

Multimodal Large Language Models for Video Understanding. The advent of large language models (Brown et al., 2020; Winata et al., 2021; Ouyang et al., 2022; Wang et al., 2024a) has spurred

substantial progress in multimodal understanding. Recent MLLMs, including proprietary models such as GPT-4o (OpenAI, 2024) and Gemini (Team et al., 2024), and open-source models such as LLaVA (Bi et al., 2025) and InternVL (Chen et al., 2024b), have achieved strong performance across diverse vision-language tasks. For online video understanding, VideoLLM-Online (Chen et al., 2024a), Flash-VStream (Zhang et al., 2025a), VideoChat-Online (Huang et al., 2025), and omni-modal models such as VITA (Fu et al., 2025b) have explored real-time streaming processing from different perspectives. Beyond these, memory-augmented approaches equip MLLMs with explicit long-term memory: EgoLife (Yang et al., 2025b) builds hierarchical textual memories for egocentric daily streams, and M3-Agent (Long et al., 2026), trained via reinforcement learning, constructs an entity-centric multimodal memory graph. However, these memory agents operate offline on complete videos and rely on costly training, and none of the above models maintains reliable long-term memory under real-streaming conditions.

Streaming and Memory Evaluation Benchmarks. A range of benchmarks evaluate MLLMs' video understanding abilities, including MVBench (Li et al., 2024), Video-MME (Fu et al., 2025a), MLVU (Zhou et al., 2025a), and LVBench (Wang et al., 2025) for general video comprehension, and EgoThink (Cheng et al., 2024), EgoExoLearn (Huang et al., 2024), and X-LeBench (Zhou et al., 2025b) for egocentric scenarios. However, all operate under offline protocols. For streaming evaluation, StreamingBench (Lin et al., 2024b), OVO-Bench (Niu et al., 2025), and OVBench (Huang et al., 2025) simulate online understanding by posing questions at different time points, but all adopt batch-feeding protocols that wait for model responses before delivering the next segment, failing to expose latency or input accumulation failures. RIVER (Shi et al., 2026) defines temporal relationships between cues, questions, and answers, but still evaluates each query over pre-recorded content via offline replay, without wall-clock streaming pressure. M3-Bench (Long et al., 2026) assesses agent memory over long videos but under an offline protocol. TeleEgo (Yan et al., 2025) introduces MPT as a concept but provides no measurement mechanism or results. In contrast, the StreaMem benchmark adopts a real-streaming protocol and enables measurement of both First-mention Accuracy and Memory Persistence Time through re-querying at different time points.

3 STREAMEM

Table 1 compares StreaMem with recent MLLM evaluation benchmarks. Existing benchmarks either evaluate under batch-feeding protocols that cannot expose latency or input-accumulation failures, or target long-video understanding without any streaming protocol; none supports real-streaming evaluation. StreaMem fills this gap as the first benchmark to jointly evaluate real-time processing and long-term memory under a real-streaming protocol.

3.1 STREAMING MEMORY TASK TAXONOMY

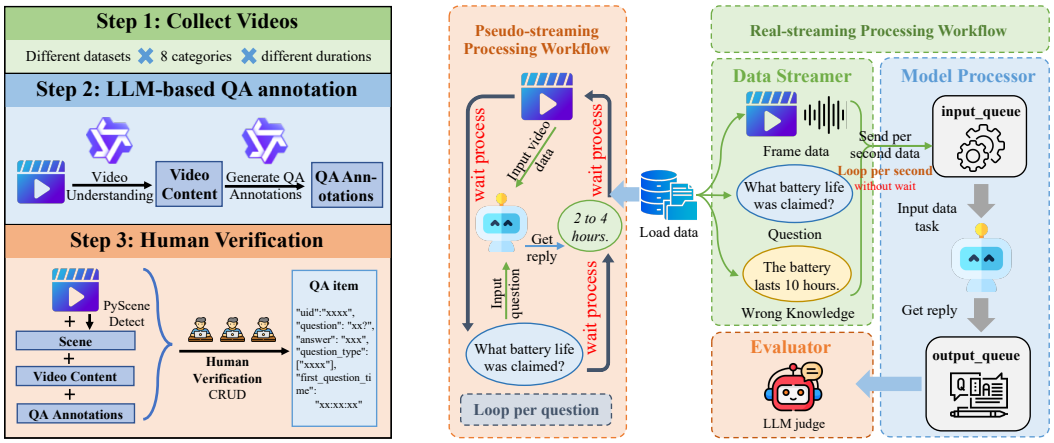
StreaMem organizes its evaluation tasks into four streaming scenarios based on the cognitive demands placed on the model during continuous video input, each comprising three subtasks, 12 in total, with detailed descriptions and examples deferred to Appendix A. *Streaming Episodic Perception* evaluates perceiving and retaining scene-level information within short temporal windows, covering **R**eal-time Scene **I**nformation **B**acktracking, **S**hort-term **E**ntity **T**racking, and Immediate **I**nteraction **C**ontent **R**ecall. *Streaming Content Understanding* evaluates comprehension of multimodal content as it unfolds in real time, emphasizing temporal structure and cross-modal integration, covering **T**emporal **I**nterval **J**udgment, **M**ultimodal **S**ynchronization **M**atching, and **R**eal-time **E**vent **C**omprehension. *Streaming Real-Time Reasoning* evaluates higher-order inference over extended temporal spans, covering **C**ross-temporal **C**ausal **I**nference, **L**ong **T**imeline **I**ntegration, and **M**ulti-scenario **C**orrelation **R**easoning. *Streaming Memory Evolution* assesses the dynamic and adaptive properties of memory, including contradiction handling, updating, and anticipation, covering **R**eal-time **M**emory **R**eflection, **E**pisodic **M**emory **U**pdate, and **P**rospective **C**ontent **I**nsight.

3.2 DATASET CONSTRUCTION

Video Collection. Videos in StreaMem are sourced from public datasets (LVBench (Wang et al., 2025), M3-Bench (Long et al., 2026)) and web-collected content, spanning 8 categories: Cartoon,

Table 1: Comparison of recent MLLM evaluation benchmarks and our proposed StreaMem benchmark. Third and First indicate third-person and first-person perspectives. CE and OE indicate closed-ended and open-ended question answering, respectively.

Benchmark	Length	Videos	QAs	Focus	Perspective	Answer Type	Real-Streaming
OVO-Bench (2025)	<30 min	644	2,814	Online Video Understanding	Third	CE / OE	×
StreamingBench (2024)	<30 min	900	4,500	Online Video Understanding	Third	CE / OE	×
LVBench (2024)	<6 h	103	1549	Long Video Understanding	Third	CE	×
TeleEgo (2025)	>12 h	5	3291	Real-world Settings	First	CE / OE	×
M3-Bench-Robot (2026)	>30 min	100	1276	Agent Memory and Reasoning	First	OE	×
StreaMem (Ours)	>30 min	210	3000	Real-time & Long-term Memory	Third / First	OE	✓



(b) Three-step dataset construction pipeline (c) Pseudo-streaming and real-streaming

Figure 2: (a) Three-step dataset construction pipeline: video collection, LLM-based QA annotation, and human verification. (b) Pseudo-streaming loops per question and waits at each step; real-streaming decouples data delivery from model processing via a producer-consumer architecture.

Documentary, Live, Self-media, Sport, TV, Interview, and First-person Perspective. 86% of videos are 30–60 minutes long, averaging ∼50 minutes, so that long-term memory is consistently tested.

QA Generation. As illustrated in Figure 2 (a), dataset construction follows three steps: (1) videos are collected and organized by category and duration; (2) a large vision-language model generates structured video content descriptions, from which it produces initial QA pairs across the 12 subtask types (Section 3.1), each recording the question, answer, type, and first-query timestamp; (3) the annotations undergo human verification, in which ten volunteer annotators, informed of the intended use of the data and trained with detailed guidelines, review each annotation against the original video, scene boundaries obtained with a boundary detection tool, and content descriptions, correcting factual errors, removing low-quality or ambiguous items, and supplementing missing coverage. The final benchmark comprises 210 videos and over 3,000 open-ended QA pairs.

3.3 Real-Streaming Processing Workflow

StreaMem adopts a real-streaming evaluation protocol that differs from pseudo-streaming benchmarks in the core timing mechanism, as illustrated in Figure 2 (b).

Pseudo-streaming Workflow. In the pseudo-streaming protocol, evaluation proceeds question by question in a strictly sequential loop: load data, wait for the model to process, input the question, wait for the response, record, and repeat. Each stage is gated by the previous one, so the data feed is always synchronized with model speed and no input accumulates, artificially eliminating the latency pressure inherent in real-streaming settings.

Real-streaming Workflow. Our real-streaming protocol decouples data delivery from model processing via a producer-consumer architecture with three components. The *Data Streamer* sends one second of frames every second at a fixed rate, without waiting for any acknowledgment, and injects questions and wrong knowledge items at their designated timestamps. The *Model Processor*

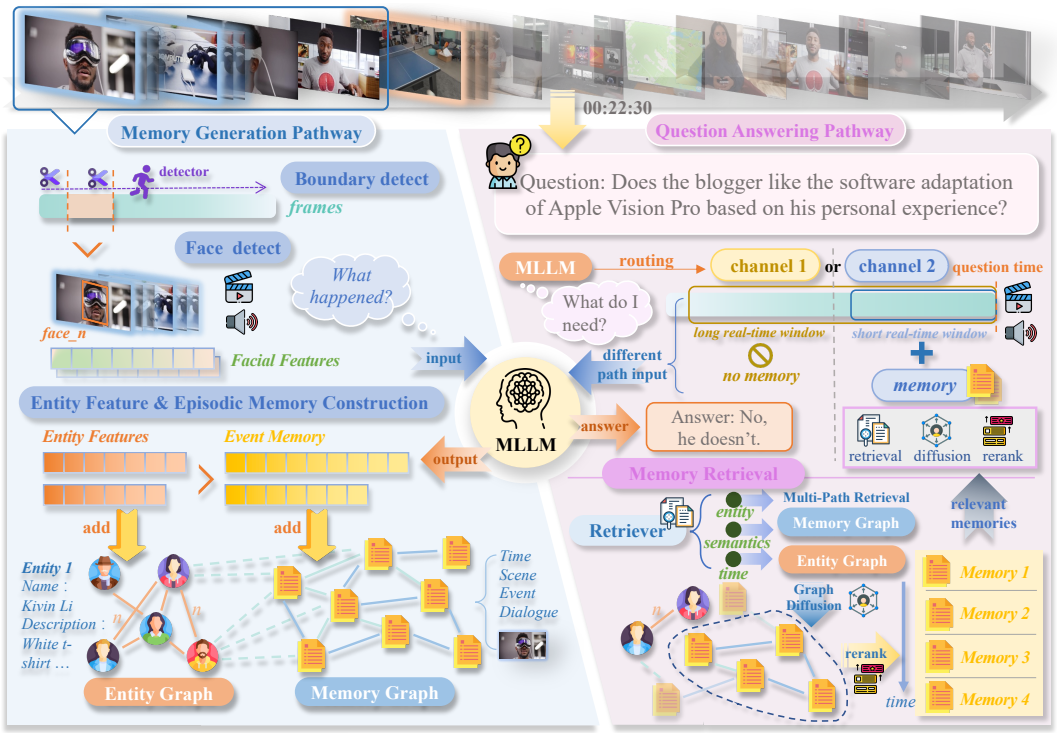


Figure 3: Overview of SOAM, consisting of a memory generation pathway (left) and a question answering pathway (right).

enqueues incoming data in an *input_queue*; a separate thread continuously dequeues tasks, invokes the model, and places replies in an *output_queue*, so inputs accumulate whenever processing falls behind the streaming rate, exposing response latency under realistic load. The *Evaluator* asynchronously scores replies from the *output_queue* with an LLM judge, recording answer correctness and the elapsed time from the question timestamp to the reply timestamp as the processing latency. Slower models are thus penalized through measurable latency rather than artificially serialized, exposing the performance–efficiency trade-off under real-streaming conditions.

4 SOAM: Streaming Omni Agentic Memory

4.1 Overview

We consider the real-streaming setting of Section 3.3: at every second t, the system receives that second's video frames and audio, and questions q may arrive at arbitrary times τ, each requiring an immediate answer without interrupting the stream. SOAM equips a MLLM with an external long-term memory under this setting, without training. SOAM consists of two asynchronous pathways (Figure 3). The *memory generation pathway* runs continuously and converts the incoming stream into a persistent, structured memory bank $\mathcal{M}_t$; the *question answering pathway* is triggered by each question and composes an answer from $\mathcal{M}_\tau$ and a real-time window of the most recent seconds. The two pathways never block each other, so memory construction keeps pace with the stream, and once a question arrives, SOAM can immediately retrieve relevant memories and answer.

4.2 Memory Generation Pathway

Event Boundary Parser. Chunking the stream at fixed intervals can cut through an action or event, so that one memory mixes fragments of unrelated activities and retrieval returns incomplete, incoherent evidence. We therefore segment the stream at *semantic event boundaries*, which keep each event intact within a single memory. A stream-adapted efficient generic event boundary

detector (Zheng et al., 2024) runs incrementally *every second* over the buffered frames, so a boundary is emitted in the same second it occurs and events are finalized on the fly as the stream progresses.

Biometric-Anchored Memory Builder. Since events are segmented independently, the same person would otherwise appear as disconnected fragments across memories (e.g., "a man" in one event, "a visitor" in another), leaving cross-segment questions about that person unanswerable. We therefore anchor person identities across the stream, converting each event into a structured memory in two stages. First, entity profiles are completed: faces detected in the event are embedded, clustered within the event, and matched against an entity registry, reusing a registered identity above a similarity threshold and otherwise assigning a new one; only identities still lacking a name or appearance description trigger a completion call, with each attribute locked to its first valid observation. Second, the MLLM produces the main structured memory: a scene description, a list of events, and dialogues with speakers attributed to face identities.

The resulting memories are organized into a heterogeneous two-layer graph, so that the graph can later be traversed both by entity and by time. The *entity graph* stores one node per identity holding its name and description, with undirected edges weighted by co-occurrence and an inverted name-to-identity index. The *memory graph* stores one node per event memory holding its structured text, time span, a keyframe, and the involved identity tokens, which serve as cross-layer links, with undirected temporal edges between consecutive events. Every memory is additionally embedded into a dense index for semantic retrieval, and all writes are incremental and asynchronous, so memory construction never stalls the per-second ingestion loop. An example memory unit is provided in Appendix B.

4.3 Question Answering Pathway

LLM Router. Retrieval is not free: it adds latency, and stale or loosely related memories can mislead the answer. Not every question needs memory: questions about the ongoing scene are answered faster and more reliably directly from fresh frames, while questions about the past require retrieval. An LLM Router therefore routes each question q arriving at time τ into one of two channels. Questions judged answerable from the present are answered directly from a *long real-time window* with a short prompt; questions judged to require looking back enter the retrieval channel with only a *short real-time window* of the most recent 10s kept for freshness.

Multi-Path Hybrid Retriever. Memory-demanding questions are anchored to different cues, such as what happened, who was involved, or when it happened, and no single index covers all three. Evidence is therefore gathered along three complementary paths over the memory bank $\mathcal{M}_\tau$ and the two-layer graph. (1) *Semantic-temporal*: the question is encoded and matched against the dense index over memory texts; the resulting scores are then diffused along the temporal edges of the memory graph, so that cause–effect pairs split across adjacent events are recovered together ($S(e') \leftarrow \max\big(S(e'), \gamma \cdot S(e)\big)$, $\gamma = 0.5$). (2) *Entity co-occurrence*: person references in the question are resolved to entity-graph identities by exact name matching or appearance-description similarity, then expanded to their strongest co-occurrence neighbors to collect the linked memories. (3) *Temporal anchor*: any explicit timestamp in the question directly activates the events whose time span covers it, assigning them the maximal score. Scores of each path are normalized to $[0, 1]$ by its maximum and fused by max-merge,

$$S_{\text{final}}(e) = \max\big(S_{\text{sem}}(e),\ S_{\text{ent}}(e),\ S_{\text{time}}(e)\big), \tag{1}$$

so that a strong signal from any single path is never diluted by the others. The top-K memories by S_{final} are kept, with ties broken by recency. Before entering the prompt, they are reranked chronologically so that the MLLM reads the evidence as a coherent narrative rather than a ranked list, with identity tokens resolved to their latest known names; the answer is then composed from these memories, their keyframes, the short real-time window, and its raw audio.

5 Experiments

5.1 Experimental Setups

Models. We evaluate eight systems in total. *Proprietary Multimodal Models* include GPT-4o (OpenAI, 2024) and Gemini-2.5-Pro (Comanici et al., 2025). *Open-source Multimodal Models* in-

clude Qwen2.5-Omni-7B (Xu et al., 2025a), Qwen3-Omni-30B (Xu et al., 2025b), MiniCPM-o-2.6 (MiniCPM-o Team, OpenBMB, 2025), VST (Guan et al., 2026), Flash-VStream (Zhang et al., 2025a), and our SOAM, covering a range of model scales and architectures. SOAM is built upon Qwen2.5-Omni-7B with off-the-shelf frozen components. Among these, MiniCPM-o-2.6, VST, Flash-VStream, and SOAM natively support online streaming, as marked in the tables. All models are evaluated under our real-streaming protocol described in Section 3.3, with model configurations deferred to Appendix C.

Evaluation Metrics. All experiments follow the real-streaming evaluation protocol of StreaMem. For tasks outside Streaming Memory Evolution, a correctly answered question is repeatedly re-queried every $d = 120$ seconds until the model fails; tasks in Streaming Memory Evolution are queried only once, and a question (containing two sub-questions or a wrong knowledge injection) counts as correct only if all sub-questions are answered correctly. We report four metrics, formulated in detail in Appendix D:

(1) *First-mention Accuracy (FMA)*, the accuracy upon each question's first query; (2) *Memory Persistence Time (MPT)*, the average duration for which correct knowledge is retained, defined over the set $\mathcal{C}$ of first-attempt-correct questions as

$$\text{MPT} = \frac{\sum_{i \in \mathcal{C}} (n_i - 1) \times d}{|\mathcal{C}|}, \tag{2}$$

with n_i the correct answer count of question i; (3) *Average Processing Latency (APL)*, the total time from the Data Streamer delivering a question to the model producing a reply, including queue waiting; and (4) *Average Response Latency (ARL)*, the time from the Model Processor receiving a question to producing the reply, isolating pure inference latency. All experimental results in this paper are from single-run experiments.

Answer Scoring. Evaluating open-ended model responses is challenging (Liu et al., 2023; Zheng et al., 2023). We adopt Qwen3-4B (Yang et al., 2025a) as the automatic judge to assess the semantic correctness of generated answers. Rather than relying on surface-level similarity metrics, Qwen3-4B focuses on semantic alignment between the model output and the reference answer. The judge assigns a score from 1 to 5 to each model response, judging the accuracy of the prediction relative to the reference; responses with a score of 3 or above are considered correct. The reliability of this automatic judge is validated in detail in Appendix E.

5.2 Main Results

Table 2: First-mention Accuracy (FMA, %) of all evaluated models across 12 subtasks and four streaming scenarios. Proprietary models are shaded gray, and the best result among open-source models in each column is **bolded**. All: average FMA across all subtasks.

Method	Online	FMA												
		RIB	SET	ICR	TIJ	MSM	REC	CCI	LTI	MCR	RMR	EMU	PCI	All
Proprietary Multimodal Models														
GPT-4o (2024)	-	67.09	62.81	58.99	44.17	59.77	55.19	38.18	20.29	40.00	55.29	45.83	12.44	47.45
Gemini-2.5-Pro (2025)	-	51.58	45.45	44.24	25.00	46.88	39.61	21.82	14.49	32.14	67.06	29.58	11.56	36.48
Open-source Multimodal Models														
Qwen2.5-Omni-7B (2025)	×	51.58	35.54	40.29	22.92	40.23	42.21	27.27	19.57	47.14	21.18	30.42	8.00	33.16
Qwen3-Omni-30B (2025)	×	**61.39**	44.63	39.57	34.58	52.34	35.71	31.36	24.28	50.36	**48.63**	30.83	4.89	39.06
MiniCPM-o-2.6 (2025)	✓	**61.39**	41.74	52.16	**37.50**	46.88	49.03	48.18	**32.25**	56.07	41.96	25.00	7.56	42.63
Flash-VStream (2025)	✓	30.70	16.94	19.78	22.92	17.58	23.38	8.18	11.59	24.29	24.71	9.58	2.22	18.30
VST (2026)	✓	47.15	31.40	35.25	8.33	41.02	32.79	34.09	21.74	46.79	10.98	20.00	4.00	28.70
SOAM (ours)	✓	60.76	**45.45**	**54.68**	28.75	**54.69**	**51.62**	**48.64**	27.17	**59.29**	36.47	**32.92**	**18.22**	**44.10**

Baseline Performance under Real-Streaming Conditions. Tables 2 and 3 present the FMA across 12 subtasks and the MPT, APL, and ARL of all evaluated systems, respectively. Among the seven baselines, GPT-4o achieves the highest average FMA, while MiniCPM-o-2.6 leads among open-source systems. Two patterns stand out. First, FMA and memory persistence are decoupled: GPT-4o attains the best FMA yet retains correct knowledge for only 170.02s, and Gemini-2.5-Pro exhibits the lowest MPT of all systems despite a moderate FMA. Second, PCI and EMU remain low across all models: most exceed 40 on RIB yet fall below 15 on PCI, confirming that anticipatory prediction and dynamic memory updating are unsolved challenges regardless of model scale or architecture, as they demand belief updating and anticipation, not recall.

SOAM Performance. SOAM attains an average FMA of 44.10, the best among all open-source offline and online systems, surpassing MiniCPM-o-2.6 by 1.47 points. This gain is not inherited from a stronger backbone: SOAM is built upon Qwen2.5-Omni-7B, which alone reaches only 33.16, so the explicit memory architecture contributes an absolute improvement of nearly 11 points. The advantage is largest on reasoning-intensive subtasks: SOAM achieves the best CCI, MCR, and PCI across the entire table, including proprietary models, indicating that explicit memory construction benefits cross-temporal reasoning and forward-looking prediction. A representative case is visualized in Figure 6 (Appendix F): the same backbone fails without memory but succeeds with it.

Memory Persistence and Latency. SOAM achieves the highest MPT of all systems at 668.41s, nearly $4\times$ GPT-4o and about $1.5\times$ MiniCPM-o-2.6, demonstrating that its memory pathways retain correct knowledge substantially longer than both proprietary and streaming-specific baselines. Crucially, this long-term persistence does not come at the cost of real-time responsiveness: SOAM's 19.25s APL and 19.02s ARL are on par with 7B-scale offline models, and its near-zero APL–ARL gap indicates that no queue backlog accumulates during streaming. These results show that explicit memory management and low-latency streaming inference can be achieved simultaneously. Further memory analyses, including unconditional MPT, survival curves, and time-weighted accuracy, are provided in Appendix F, where SOAM's memory lead persists under each.

Table 3: MPT, APL, and ARL (seconds) for all evaluated models.

Method	MPT ↑	APL ↓	ARL ↓
Gemini-2.5-Pro	31.37	10.25	10.13
GPT-4o	170.02	11.70	11.42
Qwen2.5-Omni-7B	257.83	30.84	22.19
Qwen3-Omni-30B	324.21	36.50	23.45
MiniCPM-o-2.6	456.17	**1.74**	**1.34**
VST	389.89	26.51	23.46
Flash-VStream	444.22	16.44	12.88
SOAM	**668.41**	19.25	19.02

Table 4: Module ablation on the 30-video extended subset. All: average FMA across all subtasks.

Method	FMA (%) ↑				MPT (s) ↑
	All	MCR	LTI	RMR	
SOAM	48.05	64.00	26.67	44.12	762.77
w/o Event Boundary	47.40	52.00	26.67	50.00	723.26
w/o Biometric Graph	47.19	56.00	28.89	41.18	772.62
w/o LLM Router	45.67	50.00	13.33	35.29	354.55
w/o Hybrid Retrieval	45.89	52.00	24.44	41.18	746.82
Qwen2.5-Omni-7B	33.77	48.00	17.78	17.65	300.00

Table 5: Evaluation results on OVO-Bench (Niu et al., 2025). Proprietary models are shaded gray, and the best result among open-source models in each column is **bolded**.

Model	Online	Real-Time						Backward			Forward			Overall
		OCR	ACR	ATR	STU	FPD	OJR	EPM	ASI	HLD	REC	SSR	CRR	Avg.
Proprietary Multimodal Models														
Gemini 1.5 pro	-	85.9	67.0	79.3	58.4	63.4	62.0	58.6	76.4	52.6	35.5	74.2	61.7	63.0
GPT-4o	-	69.8	64.2	71.6	51.1	70.3	59.8	57.9	75.7	48.7	27.6	73.2	59.4	59.5
Open-source Multimodal Models														
Qwen2-VL-72B	×	65.8	60.6	69.8	51.7	69.3	54.4	52.5	60.8	57.5	38.8	64.1	45.0	56.3
LLaVA-Video-7B (TMLR'25)	×	69.1	58.7	68.8	49.4	74.3	59.8	56.2	57.4	7.5	**34.1**	70.0	60.4	52.9
LLaVA-OV-7B (TMLR'25)	×	66.4	57.8	73.3	53.4	71.3	62.0	54.2	55.4	21.5	25.6	67.1	58.8	52.7
LongVU-7B (ICML'25)	×	53.7	53.2	62.9	47.8	68.3	59.8	40.7	59.5	4.8	12.2	69.5	60.8	46.7
VideoLLM-online-8B (CVPR'24)	✓	8.1	23.9	12.1	14.0	45.5	21.2	22.2	18.8	12.2	-	-	-	-
Dispider-8B (CVPR'25)	✓	57.7	49.5	62.1	44.9	61.4	51.6	48.5	55.4	4.3	18.1	37.4	48.8	41.8
TimeChatOnline-7B (MM'25)	✓	75.2	46.8	70.7	47.8	69.3	61.4	55.9	59.5	9.7	31.6	38.5	40.0	46.7
Streamforest-7B (NeurIPS'25)	✓	68.5	53.2	71.6	47.8	65.4	60.9	**58.9**	64.9	32.3	32.8	70.6	57.1	55.6
Streamo-7B (CVPR'26)	✓	77.2	66.1	76.7	45.5	66.3	72.8	55.6	58.1	33.9	30.8	57.6	**82.5**	57.9
VST-7B (ECCV'26)	✓	80.5	55.1	72.4	55.1	**76.2**	64.1	56.9	64.9	**48.4**	33.0	66.9	62.1	59.3
SOAM (ours)	✓	**90.6**	**68.8**	**80.2**	**61.8**	71.3	**77.7**	57.2	**67.6**	26.9	28.1	**71.5**	61.3	**59.8**

Results on OVO-Bench. OVO-Bench (Niu et al., 2025) stresses timestamp-sensitive online understanding through three question scenarios: real-time visual perception, backward tracing over past events, and forward active responding. We compare SOAM with Gemini 1.5 Pro (Team et al., 2024), Qwen2-VL-72B (Wang et al., 2024b), LLaVA-Video-7B (Zhang et al., 2025b), LLaVA-OneVision-7B (Li et al., 2025), and LongVU-7B (Shen et al., 2025) as offline baselines, and VideoLLM-Online (Chen et al., 2024a), Dispider (Qian et al., 2025), TimeChat-Online (Yao et al., 2025), StreamForest (Zeng et al., 2025), and Streamo (Xia et al., 2026) as online baselines. As shown in Table 5, SOAM achieves the best overall average among open-source models at 59.8, surpassing GPT-4o at 59.5 and trailing only Gemini 1.5 Pro at 63.0. The advantage is broadest on real-time perception: SOAM attains the highest score of all evaluated models on five of the six subtasks, indicating that event-based memory construction preserves the fine-grained details required for immediate perception. On the memory-centric backward-tracing subtasks, SOAM leads all open-source models on ASI and comes within 0.7 points of GPT-4o on EPM, while its SSR is likewise the best among open-source models. These results indicate that SOAM's memory design generalizes to independent online-video benchmarks beyond StreaMem.

5.3 Analysis

Ablation Study. We ablate each of SOAM's four modules on an extended subset of 30 videos, including 5 memory-intensive ones, using the same Qwen3-4B judge and d=120s re-query delay as in the main experiments. Each variant replaces one module with a simpler counterpart: *w/o Event Boundary Parser* uses fixed 10s sliding-window chunking; *w/o Biometric Graph* uses a flat database without the entity registry and the two-layer graph; *w/o LLM Router* answers every question directly from the 40s real-time window; and *w/o Hybrid Retrieval* uses only BGE-M3 dense retrieval. The last row of Table 4 is the raw Qwen2.5-Omni-7B backbone, marking the lower bound. Removing any single module lowers the overall FMA, confirming that all four modules contribute to SOAM's streaming memory capability. Ablating the LLM Router disables the entire memory pathway: MPT collapses by 53% to 354.55s with LTI halved to 13.33, demonstrating that the accumulated long-term memory is the primary source of SOAM's advantage. MCR degrades under every ablation, from 64.00 down to 50.00–56.00, indicating that cross-scenario reasoning relies on the entire memory pipeline rather than on any single component. The remaining variants degrade more modestly; given the small per-subtask question counts, differences of a few points fall within the noise range. We further analyze SOAM's key hyperparameters in Appendix F.

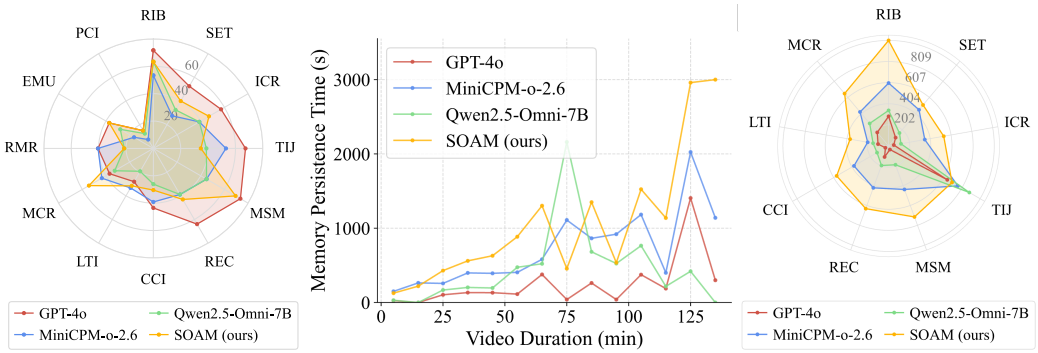


Figure 4: (a) FMA on longer videos across subtasks. (b) MPT across video durations. (c) MPT distribution across subtasks.

Further Analysis. Figure 4 presents further analyses. (a) On videos exceeding 45 minutes, per-subtask FMA patterns are consistent with the full-benchmark results in Table 2: SOAM remains the strongest open-source system, even surpassing GPT-4o on subtasks such as MSM, while PCI and EMU stay the most challenging for all systems regardless of video length, reinforcing that dynamic memory updating and anticipatory prediction are fundamental gaps rather than artifacts of video duration. (b) MPT rises with video duration for all systems, partly because longer streams permit more re-query rounds; SOAM holds the highest MPT over nearly all durations, with its lead widening to nearly 3000s on the longest videos where all baselines stay below 2000s, whereas GPT-4o's persistence remains flat despite its leading FMA, echoing the decoupling observed in Section 5.2. (c) Broken down by subtask, SOAM attains the highest persistence on nearly every subtask, with large margins on perception- and reasoning-heavy ones such as RIB and MCR, indicating that its explicit memory pipeline benefits both scene-level backtracking and cross-scenario reasoning.

6 Conclusion

We present SOAM, a training-free streaming omni agentic memory that equips MLLMs with external long-term memory, together with StreaMem, the first benchmark to jointly evaluate real-time processing and long-term memory under a real-streaming protocol. Real-streaming experiments confirm the problem we identify: existing MLLMs cannot maintain reliable long-term memory while keeping pace with the stream, a failure that pseudo-streaming protocols cannot expose. SOAM closes much of this gap without sacrificing real-time responsiveness. We hope this work provides a rigorous foundation for developing MLLMs genuinely capable of real-world streaming deployment.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic datasets, to clean and reformat the dataset, to design research experiments, to implement methods, and to assist with translation. We have not used generative AI tools to develop theoretical models or conceptual frameworks, to support qualitative and thematic data analysis, or to interpret results; the remaining tasks with required disclosure (formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, and proposing or refining hypotheses) are not applicable to this work. Additionally, we used generative AI tools to suggest experimental parameters, to create or modify scientific figures and images, to summarize and analyze existing literature, to discover research topics and identify research gaps, to identify relevant literature, to suggest the structure of this paper, to edit the paper for improved readability, and to format references. We have reviewed all AI-assisted work: AI-generated data were verified by the authors against the source videos, and the outputs of data cleaning and reformatting were manually inspected for correctness; AI-suggested experimental designs were reviewed and finalized by the authors; LLM-generated code was verified and tested for correctness by the authors; and AI-assisted translations were proofread for technical accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

StreaMem is built from videos collected from publicly accessible online platforms and is used solely for non-commercial research purposes. Our method performs face detection and recognition to anchor person identities across video segments; these components are used solely to improve streaming question answering, not to identify, profile, or track individuals, and no biometric features or identity labels beyond those already present in the source videos are released. Upon release, the dataset will be distributed for research use with a takedown mechanism to address privacy or copyright concerns from content owners. We do not foresee other ethical concerns raised by this work.

REPRODUCIBILITY STATEMENT

We have made the following efforts to ensure reproducibility. The complete implementation of SOAM and the StreaMem evaluation framework is available in the repository linked in the abstract. All components of SOAM are off-the-shelf and frozen, with model configurations detailed in the appendix. The real-streaming evaluation protocol, including the re-query mechanism and all four metrics, is specified in Sections 3.3 and 5.1. The StreaMem dataset will be open-sourced upon acceptance.

REFERENCES

Jinhe Bi, Yujun Wang, Haokun Chen, Xun Xiao, Artur Hecker, Volker Tresp, and Yunpu Ma. LLaVA steering: Visual instruction tuning with 500x fewer parameters through modality linear representation-steering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 15230–15250, 2025.

Tom Brown et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pp. 1877–1901, 2020.

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

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24185–24198, 2024b.

Sijie Cheng, Zhicheng Guo, Jinawen Wu, Kechen Fang, Peng Li, Huaping Liu, and Yang Liu. EgoThink: Evaluating first-person perspective thinking capability of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14291–14302, 2024.

Gheorghe Comanici et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Chaoyou Fu et al. 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)*, pp. 24108–24118, 2025a.

Chaoyou Fu et al. VITA: Towards open-source interactive omni multimodal LLM. *arXiv preprint arXiv:2408.05211*, 2025b.

Kristen Grauman et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18973–18990, 2022.

Yiran Guan, Liang Yin, Dingkang Liang, Jianzhong Ju, Zhenbo Luo, Jian Luan, Yuliang Liu, and Xiang Bai. Video streaming thinking: VideoLLMs can watch and think simultaneously. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.

Yifei Huang, Guo Chen, Jilan Xu, Mingfang Zhang, Lijin Yang, Baoqi Pei, Hongjie Zhang, Lu Dong, Yali Wang, Limin Wang, and Yu Qiao. EgoExoLearn: A dataset for bridging asynchronous ego- and exo-centric view of procedural activities in real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22072–22086, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3328–3338, 2025.

J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

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

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Lou, Limin Wang, and Yu Qiao. MVBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22195–22206, 2024.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVA: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5971–5984, 2024a.

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

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, 2023.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. In *The Fourteenth International Conference on Learning Representations*, 2026.

MiniCPM-o Team, OpenBMB. MiniCPM-o 2.6: A GPT-4o level MLLM for vision, speech, and multimodal live streaming on your phone, 2025.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. OVO-Bench: How far is your Video-LLMs from real-world online video understanding? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18902–18913, 2025.

OpenAI. GPT-4o system card. Technical report, OpenAI, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, pp. 27730–27744, 2022.

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

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. *Proceedings of Machine Learning Research*, 267:54582–54599, 2025.

Yansong Shi, Qingsong Zhao, Tianxiang Jiang, Xiangyu Zeng, Yi Wang, and Limin Wang. RIVER: A real-time interaction benchmark for video LLMs. *arXiv preprint arXiv:2603.03985*, 2026.

Gemini Team et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. OpenChat: Advancing open-source language models with mixed-quality data. In *The Twelfth International Conference on Learning Representations*, 2024a.

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

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, and Jie Tang. LVBench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967, 2025.

Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung. Language models are few-shot multilingual learners. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pp. 1–15, 2021.

Jiaer Xia, Peixian Chen, Mengdan Zhang, Xing Sun, and Kaiyang Zhou. Streaming video instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 31219–31229, 2026.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-Omni technical report. *arXiv preprint arXiv:2503.20215*, 2025a.

Jin Xu et al. Qwen3-Omni technical report. *arXiv preprint arXiv:2509.17765*, 2025b.

Jiaqi Yan, Ruilong Ren, Jingren Liu, Shuning Xu, Ling Wang, Yiheng Wang, Xinlin Zhong, Yun Wang, Long Zhang, Xiangyu Chen, Changzhi Sun, Jixiang Luo, Dell Zhang, Hao Sun, Chi Zhang, and Xuelong Li. TeleEgo: Benchmarking egocentric AI assistants in the wild. *arXiv preprint arXiv:2510.23981*, 2025.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, et al. EgoLife: Towards egocentric life assistant. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28885–28900, 2025b.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, et al. TimeChat-Online: 80% visual tokens are naturally redundant in streaming videos. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10807–10816, 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*, pp. 75804–75835, 2025.

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 (ICCV)*, pp. 21059–21069, 2025a.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video instruction tuning with synthetic data. *Transactions on Machine Learning Research*, 2025b.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, pp. 46595–46623, 2023.

Ziwei Zheng, Zechuan Zhang, Yulin Wang, Shiji Song, Gao Huang, and Le Yang. Rethinking the architecture design for efficient generic event boundary detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024.

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. MLVU: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701, 2025a.

Wenqi Zhou, Kai Cao, Hao Zheng, Yunze Liu, Xinyi Zheng, Miao Liu, Per Ola Kristensson, Walterio W. Mayol-Cuevas, Fan Zhang, Weizhe Lin, and Junxiao Shen. X-LeBench: A benchmark for extremely long egocentric video understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 15206–15222, 2025b.

A Detailed Task Examples

Figure 5 presents representative question-answer examples for all 12 subtasks across the four streaming scenarios in StreaMem. Each example is accompanied by the corresponding video frames and audio waveform, illustrating the multimodal and temporally grounded nature of the benchmark. Below we provide a detailed description of each subtask and its corresponding example.

Streaming Episodic Perception. This scenario tests the model's ability to perceive and retain scene-level information within a short temporal window. *RIB* requires recalling spatial details from shortly before the query time. In the example, the model is asked where a robot should place table tennis balls, demanding accurate persistence of the recent scene layout. *SET* focuses on tracking the movement or state of a specific entity over a brief interval. The example asks where a dog named Nala was fed at a specific timestamp, requiring the model to maintain entity-level representations across consecutive frames. *ICR* targets actions or interactions involving a person and an object within a recent time window. The example asks what a chef did during a specific time interval, capturing fine-grained interaction-level episodic content.

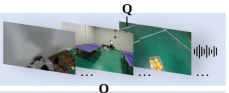
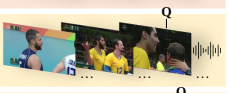
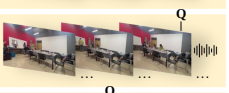
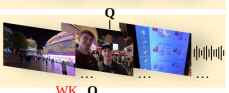
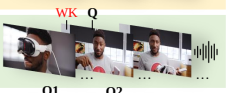
Streaming Episodic Perception	RIB	Real-time Scene Information Backtracking	**Q: Where should the robot place the table tennis balls?** *A: Beside the chair.*	
	SET	Short-term Entity Tracking	**Q: Where was the dog Nala fed at 00:16:00?** *A: Inside the house, near the kitchen.*	
	ICR	Immediate Interaction Content Recall	**Q: What does the chef do from 00:01:06-00:01:24?** *A: Find the container and wash it.*	
Streaming Content Understanding	TIJ	Temporal Interval Judgment	**Q: How long did it take for the two ladies to switch sides for the first time after they started playing table tennis?** *A: About 3 minutes and 15 seconds.*	
	MSM	Multimodal Synchronization Matching	**Q: Which clothes should Sarah wear tomorrow?** *A: A pink jacket and a sweatshirt.*	
	REC	Real-time Event Comprehension	**Q: Does the car driver accept the request at 00:09:05?** *A: No.*	
Streaming Real-Time Reasoning	CCI	Cross-temporal Causal Inference	**Q: At 01:36:57, why are the Brazilian team's athletes very happy?** *A: Because they won the game.*	
	LTI	Long Timeline Integration	**Q: What cleaning tools were used?** *A: Broom, dustpan, cloth.*	
	MCR	Multi-scenario Correlation Reasoning	**Q: What does the chef do from 00:01:06-00:01:24?** *A: McDonald's.*	
Streaming Memory Evolution	RMR	Real-time Memory Reflection	Wrong Knowledge Given: The battery lasts 10 hours. **Q: What battery life was claimed?** *A: 2 to 4 hours was claimed, not 10 hours.*	
	EMU	Episodic Memory Update	**Q1: How many people are there in the office now?** *A1:Three.* **Q2: How many people are there in the office now?** *A2: Two.*	
	PCI	Prospective Content Insight	**Q1: What will the woman in black clothes do after 14:30?** *A1: She is going to take photos.* **Q2: What will the woman in black clothes do after 28:05?** *A2: She is going to do exercise.*	

Figure 5: Overview of the 12 subtasks in StreaMem across four streaming scenarios, with task abbreviations, full names, and representative question-answer examples for each subtask.

Streaming Content Understanding. This scenario evaluates comprehension of multimodal content as it unfolds in real time. *TIJ* requires computing the elapsed time between two events from video timestamps rather than reading the answer directly from the stream. The example asks how long it took two players to switch sides after starting a table tennis game, demanding accurate internal timeline maintenance. *MSM* requires integrating audio and visual information that appears asynchronously. The example asks which clothes Sarah should wear tomorrow, where the answer requires binding a spoken instruction to a visually presented item. *REC* targets event-level understanding at a specific timestamp. The example asks whether a car driver accepts a request at a given moment, requiring immediate interpretation of the ongoing event without relying on future context.

Streaming Real-Time Reasoning. This scenario probes higher-order inference over extended temporal spans. *CCI* requires identifying causal relationships between events separated by a significant time interval. The example asks why Brazilian athletes are happy at a specific timestamp, where the cause occurred much earlier in the stream. *LTI* requires aggregating information distributed across the entire video timeline to answer an overall query. The example asks what cleaning tools were used throughout the video, demanding long-range memory retrieval. *MCR* tests the ability to connect entities or events across multiple distinct scenes. The example identifies what a chef does in a specific interval by associating information from different scenes within the same streaming session.

Streaming Memory Evolution. This scenario assesses the dynamic and adaptive properties of the model's memory over time. *RMR* injects a wrong knowledge claim into the stream and then queries the model about the actual fact. In the example, the stream claims the battery lasts 10 hours, while the video content shows the actual battery life is 2 to 4 hours; the model must detect the contradiction and report the correct answer from its own memory. *EMU* poses the same question at two temporally separated points where the ground-truth answer changes due to scene evolution. The example asks how many people are in the office at two different times, with the answer changing from three to two as someone leaves; identical question wording is enforced across both queries to ensure response differences reflect genuine memory updating. *PCI* similarly poses the same question at two time points but targets anticipatory prediction rather than factual recall. The example asks what a woman in black clothes will do after a given time, where the correct prediction changes as the stream advances and new context becomes available.

B Memory Unit Example

Table 6 shows an example memory unit produced by the memory builder (Section 4.2).

Table 6: An example memory unit produced by the memory builder. Each event becomes one unit pairing structured text (scene, events, and dialogues attributed to face identities) with its time span, a keyframe, and identity tokens.

Key	Example value
event_id	18
time	131s–136s
scene	A bathroom with a large mirror, a shower with glass doors, and a towel rack with a black towel.
entities	face_1 ("Denzel"): male, young, short black hair, black t-shirt
events	face_1 applies a product to his face in front of the mirror, rubbing it over his forehead and cheeks.
dialogues	face_1: "…they go above and beyond for their child."
keyframe	the middle frame of the event

C Model Configurations

This section details the configurations of all evaluated systems. Table 7 lists the settings shared by all systems, and Table 8 summarizes the model-specific memory management strategies.

Table 7: Settings shared by all evaluated systems.

Item	Value
Frame sampling rate	model-specific; all baselines use 1 fps
Judge	Qwen3-4B (Appendix E)
Re-query schedule	delay $d = 120$s, up to 100 rounds
Evaluation protocol	streaming prefill $\rightarrow$ question-at-time $\rightarrow$ multi-round re-query
Queue handling	shared Data Streamer; inputs accumulate in input_queue whenever processing falls behind the delivery rate
Hardware (open-source)	96GB GPUs; 2$\times$ for Qwen3-Omni-30B and Flash-VStream, 1$\times$ for the other baselines, 7$\times$ for SOAM
API environment (proprietary)	identical internal network environment for all API calls

The frame sampling rate is each model's own choice rather than a fixed evaluation setting: denser sampling only provides more input at the cost of the model's own processing budget, so it cannot confer an unfair advantage, and models are in fact encouraged to operate at higher rates. SOAM samples at 10 fps to feed its event boundary detector, which requires temporally coherent frames for accurate boundary detection; all baselines sample at 1 fps.

All offline models are wrapped with an identical sliding-window processor (frame rate, re-query protocol, and truncation logic unified); the only model-specific parameter is the history length, which is dictated by each model's native context capacity rather than chosen to favor any model. The

online models retain their officially released streaming mechanisms without modification, since forcing them into a uniform wrapper would understate their intended capabilities. More generally, hardware and parameter choices throughout follow each method's own design needs rather than any attempt to manufacture unfairness; for example, SOAM's multi-GPU setup simply instantiates its asynchronous two-pathway design. We welcome more practitioners to evaluate their systems on StreaMem with their best configurations and to report stronger results. Table 8 summarizes the resulting configurations. Table 9 lists the key parameters of SOAM's deployment and memory pipeline.

Table 8: Model-specific memory management configurations.

Model	Memory mechanism	Truncation / compression	Reset strategy
GPT-4o	sliding-window dialogue history	pop oldest ($L_{\max}$=20)	once per video
Gemini-2.5-Pro	sliding-window dialogue history	pop oldest ($L_{\max}$=20)	once per video
Qwen2.5-Omni-7B	sliding-window dialogue history	pop oldest ($L_{\max}$=80)	once per video
Qwen3-Omni-30B	sliding-window dialogue history	pop oldest ($L_{\max}$=80)	once per video
MiniCPM-o-2.6	native KV-cache incremental prefill	N/A (KV-cache path)	once per video; hard reset every 600s
VST	frame buffer + hierarchical textual memory	segment compression every 400s	once per video
Flash-VStream	CSM (temporal) + DAM (spatial) memory	cluster compression; text/ASR ≤ 30	once per video
SOAM	external two-layer memory graph	three-path hybrid retrieval, top-K=10	none (persistent memory)

Table 9: Key parameters of SOAM.

Item	Value
Base MLLM	Qwen2.5-Omni-7B (frozen), 2 vLLM instances
GPUs	7× 96GB (4 and 2 for the vLLM instances, 1 for others)
Boundary detector	stream-adapted EfficientGEBD (Zheng et al., 2024), 10 fps input
Event finalization	5s decision window, 20s max event span
Face pipeline	InsightFace (buffalo_l): RetinaFace + ArcFace, CPU
Dense index	BGE-M3
Direct-answer / retrieval window	40s / 10s
Retrieved memories (top-K)	10
Semantic path (vector / event top-k)	10 / 8
Temporal-anchor tolerance	5s
Entity path expansion	1 hop, top-2 neighbors
Temporal diffusion γ	0.5

For API-based models (GPT-4o and Gemini-2.5-Pro), APL and ARL are measured as client-side wall-clock time from request to full response, which includes network and provider overhead that cannot be separated from pure inference.

D Evaluation Metrics Formulation

We formally define the four evaluation metrics used in StreaMem. Let $\mathcal{Q}$ denote the full set of evaluated questions and $\mathcal{Q}^-$ denote the subset excluding Streaming Memory Evolution tasks (i.e., RMR, EMU, and PCI). A question i is considered correctly answered if its LLM judge score $s_i \geq 3$. For EMU and PCI, which contain two sub-questions per item, correctness requires all sub-questions to be answered correctly.

First-mention Accuracy (FMA). FMA measures the proportion of questions answered correctly on the first query:

$$\text{FMA} = \frac{\sum_{i \in \mathcal{Q}} \mathbf{1}[s_i^{(1)} \geq 3]}{|\mathcal{Q}|}, \tag{3}$$

where $s_i^{(1)}$ denotes the LLM judge score for the first response to question i.

Memory Persistence Time (MPT). MPT quantifies the average duration for which a model retains correct knowledge, computed over questions in $\mathcal{Q}^-$ that were answered correctly on the first query. Let $\mathcal{C} \subseteq \mathcal{Q}^-$ be the set of such questions, n_i the total number of correct answers across all re-queries for question i, and d the re-query delay interval ($d = 120$ seconds). MPT is defined as:

$$\text{MPT} = \frac{\sum_{i \in \mathcal{C}} (n_i - 1) \times d}{|\mathcal{C}|}. \tag{4}$$

Average Processing Latency (APL). APL measures the mean end-to-end time elapsed from when the Data Streamer delivers a question to when the model produces a reply. Let $T_{i,k}^{\text{elapsed}}$ denote the elapsed time for the k-th query of question i. APL is defined as:

$$\text{APL} = \frac{\sum_{i \in \mathcal{Q}} \sum_k T_{i,k}^{\text{elapsed}}}{\sum_{i \in \mathcal{Q}} K_i}, \tag{5}$$

where K_i is the total number of queries for question i across all re-query rounds.

Average Response Latency (ARL). ARL measures the mean inference time within the Model Processor, isolating pure model latency from system-level streaming overhead. Let $T_{i,k}^{\text{process}}$ denote the processing time for the k-th query of question i. ARL is defined as:

$$\text{ARL} = \frac{\sum_{i \in \mathcal{Q}} \sum_k T_{i,k}^{\text{process}}}{\sum_{i \in \mathcal{Q}} K_i}. \tag{6}$$

The difference APL $-$ ARL therefore captures the system-level overhead attributable to streaming data processing and queue waiting time, providing a decomposed view of end-to-end latency that pseudo-streaming protocols cannot measure.

E Judge Reliability Validation

To validate the reliability of the Qwen3-4B automatic judge used for answer scoring (Section 5.1), we conducted a human validation study. We sampled 20 videos, disjoint from the 30-video ablation subset, and 345 common first-attempt QA pairs, and asked the ten annotators from dataset construction (Section 3.2) to independently re-score all model predictions under the same 1–5 rubric, blind to model identity. The study covers GPT-4o, MiniCPM-o-2.6, Qwen2.5-Omni-7B, and Flash-VStream as four representative models of different capability levels, yielding 1,380 re-scored predictions in total.

Human–Judge Agreement. Table 10 reports the agreement between Qwen3-4B and human annotators. Averaged over the four model subsets, Qwen3-4B reaches 89.06% binary agreement under the score$\geq$3 correctness threshold and a quadratic weighted kappa (QWK) of 0.8668, in the "almost perfect" range of the standard interpretation scale (Landis & Koch, 1977). This is consistent with the agreement level reported for GPT-4-family judges, which approaches human–human agreement (Zheng et al., 2023), indicating that Qwen3-4B scores open-ended answers as reliably as established judges.

Table 10: Agreement between Qwen3-4B and human annotators on 345 first-attempt predictions per model subset. Binary: agreement under the score$\geq$3 correctness threshold; QWK: quadratic weighted kappa.

Model Subset	Exact	±1	Binary	Pearson r	QWK	F1
GPT-4o	63.77	86.96	87.25	0.8415	0.8481	0.8650
Flash-VStream	76.52	94.78	93.33	0.9092	0.9025	0.8671
MiniCPM-o-2.6	66.09	91.88	86.96	0.8772	0.8726	0.8443
Qwen2.5-Omni-7B	64.93	89.86	88.70	0.8636	0.8440	0.8539
Average	**67.83**	**90.87**	**89.06**	**0.8729**	**0.8668**	**0.8576**

Convergent Validation with a Second Judge. As convergent evidence, we additionally scored the same 1,380 predictions with GPT-4o, whose reliability as an automatic judge is independently established (Zheng et al., 2023; Liu et al., 2023). As shown in Table 11, Qwen3-4B agrees with GPT-4o at 90.26% binary agreement and 0.8716 QWK, essentially matching its agreement with human annotators at 89.06% and 0.8668. Two independent validation signals thus converge on the reliability of Qwen3-4B, which remains substantially more cost-effective at StreaMem's scale of 3,000+ QA pairs with repeated re-queries.

Table 11: Agreement of Qwen3-4B with human annotators versus with GPT-4o as an independently validated second judge, on the same 1,380 predictions.

	vs. Human		vs. GPT-4o	
Model Subset	**Binary**	**QWK**	**Binary**	**QWK**
GPT-4o	87.25	0.8481	90.14	0.8529
Flash-VStream	93.33	0.9025	93.62	0.8929
MiniCPM-o-2.6	86.96	0.8726	88.41	0.8901
Qwen2.5-Omni-7B	88.70	0.8440	89.86	0.8505
Average	**89.06**	**0.8668**	**90.26**	**0.8716**

Threshold Robustness. The gap between binary agreement and ± 1 agreement is small across subsets, at 0.29–4.92 points, meaning most disagreements are one-point borderline confusions rather than gross errors. The score$\geq$3 threshold in our correctness definition is therefore not a fragile cutoff sensitive to judge noise.

Failure Analysis. We inspected cases at the 2/3 score boundary and stratified agreement by prediction length (Table 12). Agreement declines as answers grow longer, from a Cohen's kappa of 0.846 on short answers to 0.680 on long ones; the judge is thus somewhat less reliable on verbose answers, although agreement remains in the substantial range at every tier. The boundary cases share a common pattern: the prediction overlaps the reference only partially (for example, naming the correct category but not the specific item, or giving the right order of magnitude but not the exact figure), reflecting the inherent ambiguity of partial-credit scoring in open-ended QA rather than systematic judge error, so these disagreements do not undermine the judge's overall reliability.

Table 12: Human–judge agreement stratified by prediction length on the human-validated subset (1,380 predictions).

Length Bin	n	**Agreement**	**Cohen's κ**	**FMA (human)**	**FMA (Qwen3-4B)**
Short ($\leq$2 words)	497	93.16%	0.846	35.21%	31.59%
Mid (3–10 words)	481	88.77%	0.765	40.33%	38.67%
Long ($>$10 words)	402	84.33%	0.680	41.04%	44.28%

F Extended Experimental Results

Table 13: Memory Persistence Time (MPT, s) of all evaluated models across 12 subtasks and four streaming scenarios. Proprietary models are shaded gray, and the best result among open-source models in each column is **bolded**. All: average MPT across all subtasks.

Method	Online	MPT												
		RIB	SET	ICR	TIJ	MSM	REC	CCI	LTI	MCR	RMR	EMU	PCI	All
Proprietary Multimodal Models														
Gemini-2.5-Pro (2025)	×	70.67	8.73	3.9	60.00	12.00	36.39	15.00	12.00	426.7	-	-	-	31.37
GPT-4o (2024)	×	284.15	104.21	50.49	648.68	36.86	112.24	35.71	102.86	169.29	-	-	-	170.02
Open-source Multimodal Models														
Qwen2.5-Omni-7B (2025)	×	340.86	161.86	120.00	892.36	192.23	199.38	128.00	151.11	281.82	-	-	-	257.83
Qwen3-Omni-30B (2025)	×	377.32	137.78	112.36	**1007.7**	247.16	325.09	264.35	82.39	373.62	-	-	-	324.21
MiniCPM-o-2.6 (2025)	✓	603.09	452.67	351.72	765.33	444.00	427.55	382.64	203.6	425.73	-	-	-	456.17
VST (2026)	✓	455.03	165.79	357.55	366.00	290.29	420.59	363.20	228.00	619.24	-	-	-	389.89
Flash-VStream (2025)	✓	649.48	374.63	266.18	949.09	280.00	281.67	160.00	292.05	356.47	-	-	-	444.22
SOAM	✓	**1011.88**	**512.73**	**536.05**	700.87	**724.29**	**640.00**	**574.21**	**374.40**	**655.66**	-	-	-	**668.41**

Tables 13, 14, and 15 report the per-subtask breakdown of MPT, APL, and ARL for all evaluated models.

Memory Persistence Time. Across subtasks, SOAM attains the highest MPT on eight of the nine re-queried subtasks, with the largest margins on RIB at 1011.88s and MSM at 724.29s, confirming that its memory advantage is broad-based rather than confined to specific task types. The only exception is TIJ, where Qwen3-Omni-30B remains highest at 1007.7s versus SOAM's 700.87s; TIJ generally yields the highest MPT among baselines, suggesting that temporal interval information, once correctly encoded, is retained longer than other content types. SET and ICR are the most

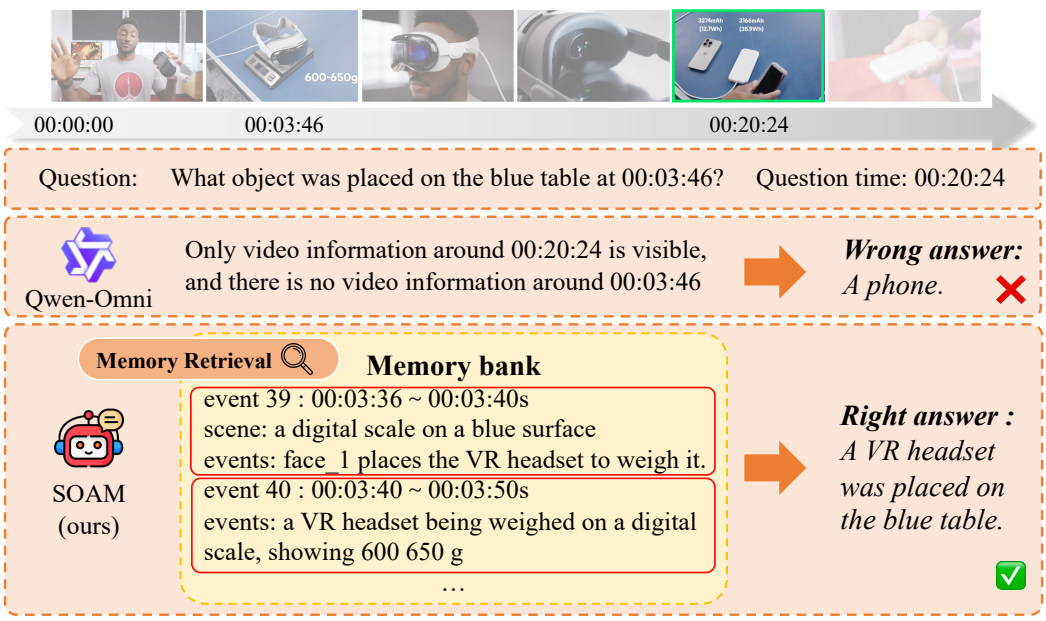


Figure 6: A qualitative example from StreaMem. The question, asked at 00:20:24, concerns an object placed on the blue table at 00:03:46. The backbone Qwen2.5-Omni-7B accesses only the current real-time window and answers incorrectly, while SOAM retrieves the relevant memory units from its memory bank and answers correctly.

Table 14: Average Processing Latency (APL, *s*) of all evaluated models across 12 subtasks and four streaming scenarios. Proprietary models are shaded gray, and the best result among open-source models in each column is **bolded**. All: average APL across all subtasks.

Method	Online	APL												
		RIB	SET	ICR	TIJ	MSM	REC	CCI	LTI	MCR	RMR	EMU	PCI	All
Proprietary Multimodal Models														
Gemini-2.5-Pro (2025)	×	9.87	10.03	10.43	9.98	10.33	10.30	10.64	10.15	10.44	10.59	10.47	10.08	10.25
GPT-4o (2024)	×	10.09	10.03	27.75	9.85	10.49	11.05	9.36	10.41	11.31	11.44	10.23	9.62	11.7
Open-source Multimodal Models														
Qwen2.5-Omni-7B (2025)	×	38.08	25.75	28.39	25.68	41.45	24.92	33.24	28.97	35.39	26.62	26.27	27.89	30.84
Qwen3-Omni-30B (2025)	×	32.46	35.38	37.86	36.75	42.06	36.46	38.85	39.19	37.74	32.36	36.34	33.47	36.50
MiniCPM-o-2.6 (2025)	✓	**1.54**	**1.63**	**1.60**	**2.08**	**1.88**	**1.65**	**1.94**	**2.09**	**1.99**	**1.65**	**1.33**	**1.47**	**1.74**
VST (2026)	✓	23.92	22.86	23.40	27.70	27.49	26.36	32.98	28.66	29.32	26.82	24.31	24.93	26.51
Flash-VStream (2025)	✓	13.71	12.64	15.95	17.61	12.91	13.64	14.73	19.37	13.84	14.21	12.80	37.13	16.44
SOAM	✓	18.58	18.53	17.48	20.33	18.63	18.50	20.63	19.69	20.93	20.56	20.14	20.35	19.25

forget-prone subtasks for baselines, yet SOAM still retains 512.73s and 536.05s on them, about $1.4\times$ and $1.5\times$ the best baselines, Flash-VStream's 374.63s and VST's 357.55s. Gemini-2.5-Pro remains highly uneven, with MCR at 426.7s but SET at only 8.7s.

Average Processing Latency. MiniCPM-o-2.6 achieves the lowest APL across all 12 subtasks, consistently below 2.1s, reflecting the efficiency of its streaming architecture. SOAM's APL is tightly clustered between 17.5s and 20.9s on every subtask, comparable to the 7B-scale offline models despite the added memory construction and retrieval stages, with PCI at only 20.35s, indicating that its retrieval pathway does not disproportionately slow forward-looking queries. Among offline models, Qwen3-Omni-30B exhibits the most uniform APL distribution, ranging from 32.4s to 42.1s, consistent with its larger but more stable inference cost. Flash-VStream shows an unusually high APL on PCI (37.1s vs. average 16.4s), suggesting that forward-looking queries require more processing time under its memory compression mechanism.

Average Response Latency. The ARL patterns largely mirror APL, confirming that inference rather than queue waiting is the dominant latency source for most models. Qwen2.5-Omni-7B and

Table 15: Average Response Latency (ARL, *s*) of all evaluated models across 12 subtasks and four streaming scenarios. Proprietary models are shaded gray, and the best result among open-source models in each column is **bolded**. All: average ARL across all subtasks.

Method	Online	ARL												
		RIB	SET	ICR	TIJ	MSM	REC	CCI	LTI	MCR	RMR	EMU	PCI	All
Proprietary Multimodal Models														
Gemini-2.5-Pro (2025)	×	9.74	9.90	10.31	9.86	10.20	10.18	10.52	10.03	10.31	10.48	10.34	9.95	10.13
GPT-4o (2024)	×	9.72	9.98	27.31	9.63	10.29	10.85	9.00	10.18	11.09	11.14	9.95	9.42	11.42
Open-source Multimodal Models														
Qwen2.5-Omni-7B (2025)	×	31.82	19.08	20.55	18.57	24.98	20.17	20.91	19.99	20.91	20.10	19.87	20.81	22.19
Qwen3-Omni-30B (2025)	×	22.03	22.53	23.57	20.93	29.94	23.75	24.31	24.86	23.69	22.07	22.59	22.70	23.45
MiniCPM-o-2.6 (2025)	✓	**1.15**	**1.27**	**1.28**	**1.62**	**1.49**	**1.28**	**1.51**	**1.77**	**1.60**	**1.29**	**1.03**	**1.12**	**1.37**
VST (2026)	✓	21.45	20.41	21.60	23.23	24.21	23.20	27.17	24.29	26.28	25.13	21.90	22.52	23.46
Flash-VStream (2025)	✓	10.45	10.31	11.24	12.11	10.24	11.16	12.09	13.56	11.32	10.04	9.82	33.79	12.88
SOAM	✓	18.36	18.30	17.30	20.13	18.42	18.28	20.38	19.51	20.68	20.30	19.84	20.07	19.02

Qwen3-Omni-30B show the largest APL–ARL gaps, with APL exceeding ARL by 8.65s and 13.05s, indicating significant queue backlog under continuous streaming load. For MiniCPM-o-2.6, the gap is negligible across all subtasks, averaging below 0.4s, confirming that its streaming architecture effectively eliminates system-level overhead. SOAM likewise shows a near-zero gap of at most 0.3s on every subtask, confirming that its asynchronous memory construction and retrieval never block question answering.

Unconditional MPT and Selection Bias. Conditional MPT is computed only over first-attempt-correct questions, a self-selected subset whose size shrinks with FMA (from 50% of questions for GPT-4o down to 20% for Flash-VStream). This inflates low-FMA models: under conditional MPT, Flash-VStream (the lowest FMA) scores $2.6\times$ above GPT-4o (the highest FMA), an order reversal. Unconditional MPT counts first-attempt-incorrect questions as zero, making the denominator identical across models. As Table 16 shows, the inversion largely disappears under unconditional MPT: GPT-4o and Flash-VStream become comparable at 85.08s and 88.81s, while most other orderings are preserved. SOAM remains first at 323.69s, about $1.5\times$ the best baseline, with the lowest inflation among open-source systems at 106.5%, confirming that its memory advantage is not an artifact of subset selection. Note that re-query is by design scheduled only for first-attempt-correct questions, since knowledge never acquired cannot be re-tested; conditional MPT therefore measures the forgetting speed of demonstrably encoded knowledge and should be read jointly with, rather than instead of, unconditional MPT.

Table 16: Conditional vs. unconditional MPT and time-weighted accuracy (TWA). Unconditional MPT counts first-attempt-incorrect questions as zero; inflation: relative gap of conditional over unconditional MPT. Proprietary models are shaded gray, and the best result among open-source models in each column is **bolded**.

Method	**FMA (%) ↑**	**Cond. MPT (s)**	**Uncond. MPT (s)**	**Inflation (%) ↓**	**TWA (%) ↑**
Gemini-2.5-Pro (2025)	36.48	31.37	11.37	175.8	4.95
GPT-4o (2024)	47.45	170.02	85.08	99.8	26.11
Qwen2.5-Omni-7B (2025)	33.16	257.83	95.51	169.9	30.33
Qwen3-Omni-30B (2025)	39.06	324.21	136.34	137.8	35.81
MiniCPM-o-2.6 (2025)	42.63	456.17	217.70	109.5	33.52
VST (2026)	28.70	389.89	131.52	196.4	42.16
Flash-VStream (2025)	18.30	444.22	88.81	400.2	29.61
SOAM	**44.10**	**668.41**	**323.69**	**106.5**	**66.66**

Survival Curves. Figure 7 plots strict survival curves over the same first-attempt-correct subset as conditional MPT, since only those questions are ever re-queried and forgetting is defined only for knowledge demonstrably acquired; the unconditional metrics in Table 16 complement this view for cross-model comparison. A question is treated as forgotten once answered incorrectly and does not recover. All systems lose the majority of retained knowledge within the first 10 minutes, and no model demonstrates durable long-term retention, reinforcing our main finding that memory persistence remains unsolved under real-streaming conditions. SOAM attains the highest survival at every horizon: 37.95% / 24.87% / 12.65% at 10 / 20 / 30 minutes, versus 24.20% / 11.01% / 3.73% for the

best baseline MiniCPM-o-2.6. Yet it still decays quickly, indicating that durable retention remains open even with explicit memory management.

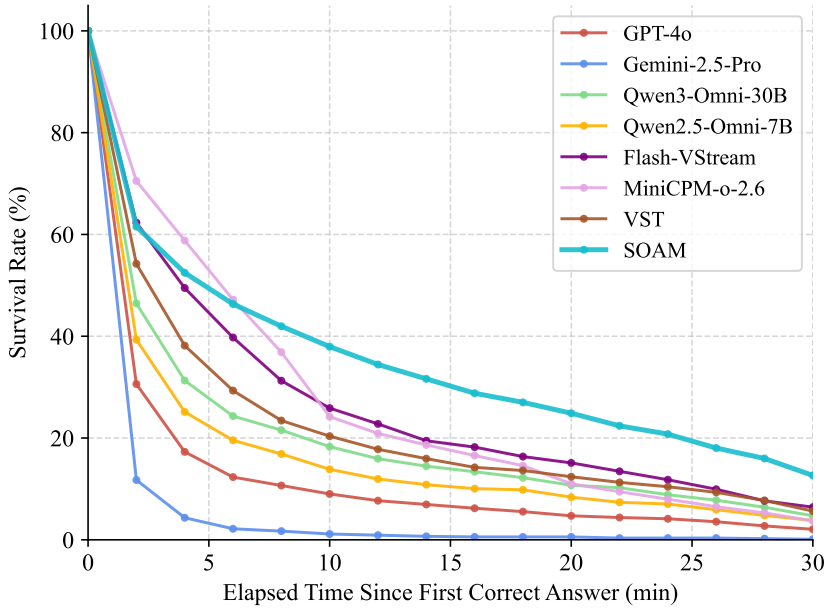


Figure 7: Memory survival curves over first-attempt-correct questions in the first 30 minutes (strict survival; longer tails are sample-sparse).

Time-Weighted Accuracy. TWA jointly penalizes first-attempt failure and subsequent forgetting: each re-queried question is weighted by its retained duration, $w_i = \max(\text{valid_count}_i, 1) \times d$, and labeled by whether it is still answered correctly at the final re-query, so that $\text{TWA} = \sum_i w_i y_i / \sum_i w_i$. Rankings under TWA differ from those under FMA or conditional MPT alone: GPT-4o, the highest-FMA system, reaches only 26.11%, while VST rises to second place at 42.16%, showing that no single metric captures streaming memory quality. SOAM achieves 66.66% TWA, about $1.6\times$ the second-best system, confirming that its advantage holds under the joint metric as well.

Hyperparameter Analysis. We analyze three key hyperparameters of SOAM on the same 30-video extended subset used in the ablation study: the number of retrieved memories (top-K), the retrieval window, and the direct-answer window. Figure 8 shows that both FMA and MPT remain stable across a broad range: FMA varies within 45.0–50.0% across all settings, and MPT within 632–794s, with the default top-K=10, 10s retrieval window, and 40s direct-answer window sitting at or near the peak. SOAM is thus not sensitive to these hyperparameters.

G Dataset Video Statistics

Figure 9 presents the category and duration distribution of videos in StreaMem. The dataset covers 8 diverse content categories: Cartoon, Documentary, Live, Self-media, Sport, TV, Interview, and First-person Perspective. The inclusion of both third-person broadcast content and first-person egocentric recordings ensures broad coverage of real-world streaming scenarios, ranging from structured media content to unscripted daily-life activities.

In terms of duration, the majority of videos fall in the 30–60 minute range, accounting for 86% of the dataset. Videos in the 90–120 minute range constitute 6%, followed by those in the 60–90 minute range at 4%, videos under 30 minutes at 3%, and videos in the 120–150 minute range at 1%. This distribution reflects the benchmark's emphasis on long-form streaming scenarios: nearly all videos exceed 30 minutes, ensuring that long-term memory is consistently tested throughout evaluation.

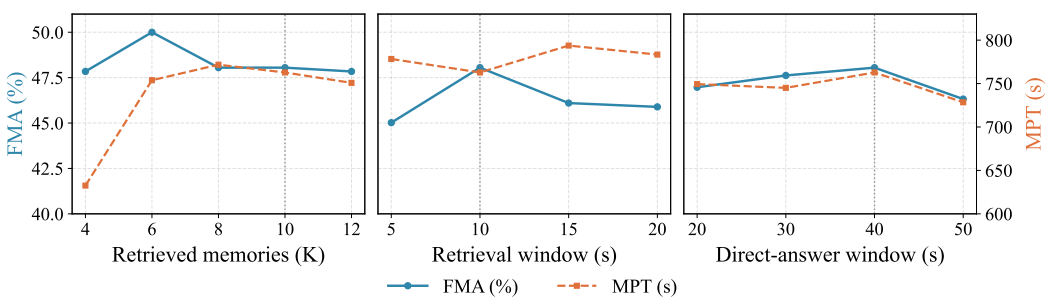


Figure 8: Hyperparameter analysis on the 30-video extended subset: FMA and MPT across the retrieved-memory count, retrieval window, and direct-answer window. Dotted lines mark the default settings.

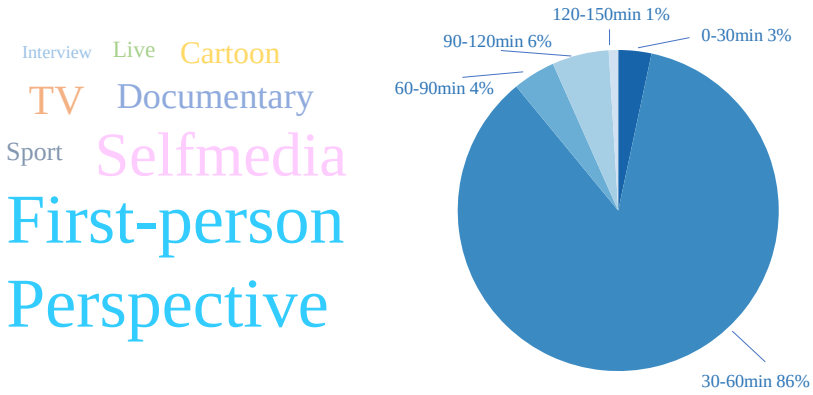


Figure 9: Video category distribution (left, word cloud) and duration distribution (right, pie chart) of StreaMem. Videos span 8 diverse content categories, with 86% falling in the 30–60 minute range.

The diversity in both content category and video duration is intentional. Different content types impose distinct challenges on streaming models: TV programs and interviews involve structured dialogue and information exchange, sports videos require tracking fast-moving entities and events, selfmedia and live content are less scripted and more temporally unpredictable, and first-person perspective videos demand egocentric scene understanding. Together, these categories provide a testbed for evaluating the streaming memory capabilities defined in StreaMem.

H LLM Judge Prompt

Figure 10 shows the prompt used by the LLM judge to score model responses. The judge is instructed to act as an objective scorer and output a single JSON object without any additional text or explanations. Given the question, ground-truth answer, and model prediction, the judge assigns an integer score from 1 to 5 based on factual consistency and coverage of key points: 5 for completely correct, 4 for mostly correct, 3 for partially correct, 2 for mostly incorrect, and 1 for completely incorrect. Empty model answers are automatically assigned a score of 1. Responses with a score of 3 or above are considered correct in our evaluation protocol.

I Model Processor Prompt

Figure 11 shows the prompt used by the Model Processor to guide the MLLM in processing streaming inputs. The system prompt instructs the model to act as a professional video content understanding assistant, receiving video frames at 1 frame per second and audio segments at 1 segment per

LLM Evaluator prompt

You are an objective scorer. Please output only one JSON object (strict), without any extra text or explanations.

Given:

Question : '{question}'

Ground_truth : '{ground_truth}'

Model Answer : '{prediction} '

Please score the model answer on an integer from 1 to 5 based on its consistency with the Ground_truth in terms of facts and key points:

5 = Completely correct;

4 = Mostly correct;

3 = Partially correct;

2 = Mostly incorrect;

1 = Completely incorrect.

If the Model Answer is empty, the score is 1.

Output only the following JSON (strict): for example {{\"score\": 4}}

Figure 10: LLM judge prompt used to score model responses. The judge assigns an integer score from 1 to 5 based on factual consistency with the ground-truth answer, with scores of 3 or above considered correct.

second. The model is required to carefully observe characters, scenes, actions, and dialogues, extract relevant information to answer questions accurately and concisely, and honestly acknowledge when information is insufficient.

The input prompt distinguishes between two input types. For question-type inputs, the model is instructed to answer briefly based on the provided video and audio content, with explicit clarification that timestamps such as *00:12:00* refer to video timestamps rather than wall-clock time, and that responses should prioritize keyword sequences or short sentences without irrelevant content. For wrong knowledge-type inputs, the model is asked to memorize the injected information and proactively update its memory if conflicts arise with subsequent observations, which directly supports the RMR subtask evaluation.

J Model Processor Pseudocode

We present simplified pseudocode for four representative model processors evaluated in StreaMem. All processors share a unified interface: ProcessUnit ingests one second of frames x_t and audio a_t, ProcessWk injects wrong knowledge text, and ProcessQuestion generates a reply $\hat{y}$. The processors differ in their memory management strategies, as detailed below.

Flash-VStream maintains a dual-GPU architecture where a dedicated MemManager continuously compresses incoming frames into a STAR memory bank $\mathcal{M}$, decoupling frame embedding from inference to reduce response latency. **VST** constructs a hierarchical textual memory: each segment of frames is compressed into a summary s_i, and the model performs iterative reasoning over accumulated summaries before producing a final answer. **MiniCPM-o-2.6** accumulates KV cache via a stateful streaming prefill interface, with periodic session resets at interval T_r to prevent context overflow and

Model Processor Prompt

System_prompt:

You are a professional video content understanding assistant. Your tasks are:

1. **Input Content**: You will receive video frame images (1 frame per second) and audio segments (1 segment per second)

2. **Task Requirements**: Carefully observe the video content, including characters, scenes, actions, dialogues, etc.

- Pay attention to important information such as speech content and background sound effects in the audio.
- Extract relevant information from the video content to answer the questions raised.
- Answers should be accurate, concise, and based on the actual content in the video
- If the information is insufficient to answer the question, please honestly state so.

3. **Answer Format**:

Answer the question directly, without additional explanations or opening remarks.

Please answer the question accurately based on the received video content.

Input_prompt:

if type = "question":

Please answer this question briefly based on the provided video and audio content. The times mentioned in the question, such as '00:12:00', refer to video timestamps, meaning 12 minutes into the video. The video provided to you is 1fps, which means each frame represents 1 second. Please prioritize providing keyword sequences or short sentences.Do not output content unrelated to the question.

question: {question}

your answer:

elif type == "wrong knowledge":

I will tell you about the events that will happen next. Please remember them. If there are conflicts with subsequent memories, please update the memory in time.

the events that will happen: {wrong_knowledge}

Figure 11: Model Processor prompt used to guide the MLLM during streaming evaluation. The system prompt defines the model's role and input format, while the input prompt handles two types of inputs: question answering and wrong knowledge injection for the RMR subtask.

maintain stable inference cost. **GPT-4o** maintains a sliding window history $\mathcal{H}$ of image-text turns, trimming old frames when payload exceeds the API size limit $B_{\max}$.

Algorithm 1 Flash-VStream Processor

Require: model, MemManager, $\mathcal{M}$ (STAR memory), texts $\leftarrow$ []

procedure PROCESSUNIT(x_t, a_t, t)
  texts $\leftarrow$ texts $\cup$ {ASR(a_t, t)}; send x_t to MemManager
end procedure
procedure PROCESSQUESTION(q) $\rightarrow \hat{y}$
  flush $\mathcal{M}$; $v \leftarrow |\mathcal{M}|$
  return Decode(model.generate(vid_token $\oplus$ texts $\oplus$ q, dummy_tokens$= v$))
end procedure

Algorithm 2 VST Processor

Require: model, $\mathcal{F}$ (frame buffer), $\mathcal{S}$ (segment summaries)

procedure PROCESSUNIT(x_t, t)
  $\mathcal{F} \leftarrow \mathcal{F} \cup \{x_t\}$; **if** segment boundary **then** $\mathcal{S} \leftarrow \mathcal{S} \cup \{$model.summarize($\mathcal{F}$)$\}$; $\mathcal{F} \leftarrow \emptyset$
end procedure
procedure PROCESSQUESTION(q) $\rightarrow \hat{y}$
  $\{C_i\}_{i=1}^{n} \leftarrow$ split(subsample($\mathcal{F}$), n)
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{$model.generate($\mathcal{S}, C_i$)$\}_{i=1}^{n-1}$
  return model.generate($\mathcal{S}, C_n, q$)
end procedure

K DATASET LICENSE

The annotations of StreaMem will be released to the community under the CC BY-NC-SA 4.0 license, consistent with the licenses of our source datasets LVBench and M3-Bench. By downloading our dataset from our website or other sources, the user agrees to adhere to the terms of CC BY-NC-SA 4.0

Algorithm 3 MiniCPM-o-2.6 Processor

Require: model, session σ, reset interval T_r
procedure PROCESSUNIT(x_t, a_t, t)
  if $t \bmod T_r = 0$ **then** reset σ; re-prefill system prompt
  end if
  model.streaming_prefill(σ, $\{x_t, a_t, t\}$)
end procedure
procedure PROCESSQUESTION(q) $\rightarrow \hat{y}$
  return model.streaming_generate(σ, q)
end procedure

Algorithm 4 GPT-4o / Gemini Processor

Require: client, $\mathcal{H} \leftarrow []$, $N_{\max}$, $B_{\max}$
procedure PROCESSUNIT(x_t, a_t, t)
  $\mathcal{H} \leftarrow \mathcal{H} \cup \{(x_t,\ \text{ASR}(a_t, t))\}$; **if** $|\mathcal{H}| > N_{\max}$ **then** drop oldest
end procedure
procedure PROCESSQUESTION(q) $\rightarrow \hat{y}$
  trim $\mathcal{H}$: replace old x_t with "[omitted]" until payload $\leq B_{\max}$
  return client.chat($\mathcal{H} \cup \{q\}$, T=0)
end procedure

and the licenses of the source datasets. StreaMem is intended for academic research only; commercial use in any form is prohibited.