CAM: Question Answering on Entity-Centric Videos with Continuous Extraction and Adaptive Querying

Yizhou Tian [1] **Zizhe Chen** [1] **Shiyuan Deng** [2] ✉ **Garry Yang** [1] **Zijie Dai** [1]
Luohao Pan [1] **Hao Lin** [1] **Peiqi Yin** [1] **Xiao Yan** [3] **James Cheng** [1]

[1] Department of Computer Science and Engineering, The Chinese University of Hong Kong
[2] Knowin AI [3] Institute for Math and AI, Wuhan University
yztian25@cse.cuhk.edu.hk jeremydeng@knowin.ai

Abstract

Memory facilitates question answering over long videos by extracting and retrieving facts to fit within the limited context windows of multimodal LLMs (MLLMs). Existing solutions typically extract independent memory entries from fixed-length video clips and thus cannot capture high-level semantics that need to be summarized over extended time periods, such as character traits and relations. Moreover, they rely solely on similarity-based retrieval and may fail to retrieve the fine-grained details required for question answering. To tackle these problems, we propose CAM, featuring *continuous extraction* for high-level semantics and *adaptive querying* for fine-grained details. In particular, CAM stores the entities and relations extracted from video clips in a knowledge graph. To capture the high-level semantics of each entity or relation, CAM summarizes the local subgraph of the target entity or relation once the subgraph reaches a predefined size. To retrieve the fine-grained details required for question answering, CAM supports multiple search methods, including knowledge graph traversal, video re-watching, and audio listening. It utilizes a planner-executor-verifier pipeline to adaptively compose these search methods according to question intent. Evaluations on three benchmarks show that CAM outperforms SOTA baselines and improves their accuracy by up to 23 percentage points. Code is available at `https://github.com/Jake-Tian/CAM`.

1 Introduction

The latest multimodal LLMs (MLLMs) can perceive and reason over complex visual and linguistic signals [1–3]. However, directly using MLLMs to process long videos is challenging since the context windows of MLLMs are limited, and it is costly to process many input tokens. Real-world videos, such as observing continuous robotic operations [4], streaming first-person footage from AI glasses [5], or following a plot in a cinematic movie [6], are usually *entity-centric* and record the attributes, events and relations for a persistent set of entities (e.g., characters and objects). As such, video memory systems [5, 7, 8] first extract facts about these entities from the videos and then retrieve only the related facts during question answering to reduce input tokens.

We illustrate the designs of three representative video memory systems in Figure 1. In particular, EgoRAG [5] segments a video into short clips and generates hierarchical text captions on rigid temporal boundaries. To answer questions, these captions are searched based on text similarity. M3-Agent [8] explicitly models entities and organizes the facts according to their related entities. While it constructs an entity-centric memory, its retrieval mechanism fundamentally underutilizes the graph topology, relying primarily on flat textual similarity matching that struggles to navigate

✉Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

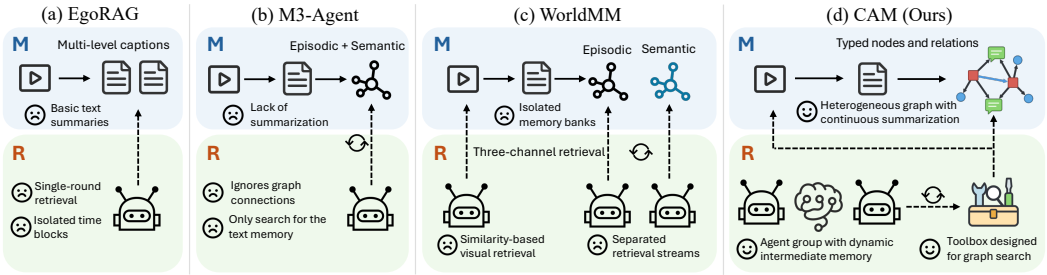


Figure 1: Comparing existing video memory systems and our CAM. The blue "M" denotes memory construction, while the orange "R" denotes reasoning. (a) EgoRAG [5] generates a text caption for each video clip and searches these captions. (b) M3-Agent [8] extracts facts related to each entity. (c) WorldMM [7] builds multiple memory modules but completely decouples visual features from textual knowledge, leading to fragmented retrieval. (d) CAM (Ours) organizes the entities and relations into a temporally grounded, heterogeneous knowledge graph, continuously summarizes high-level semantics about each entity and relation, and searches the knowledge graph adaptively using multiple methods.

complex relational chains. WorldMM [7] attempts to capture both temporal events and semantics by maintaining separate episodic, semantic, and visual memory modules. Despite this, WorldMM physically segregates its knowledge into decoupled memory banks. Because visual features are isolated from the textual knowledge graph, the agent is forced to conduct independent, fragmented retrieval streams, failing to achieve a unified multimodal understanding of complex interactions.

Building effective memory for entity-centric videos presents two challenges. First, high-level semantics, such as character traits and persistent relationships, often require consolidating evidence across multiple clips. However, evidence accumulates at different rates for different characters and character pairs. Fixed temporal boundaries therefore do not necessarily indicate when sufficient evidence is available for a reliable abstraction. Summarizing too early may produce conclusions based on limited observations, whereas delaying summarization increases the amount of raw information that must be retrieved and processed. A memory system therefore needs to update abstractions according to the evidence accumulated for each entity or relation.

Second, answering a question requires access to evidence that may be scattered across the memory or omitted during its construction. Some questions require following relations across entities and combining observations from different times, which similarity matching over individual descriptions may not retrieve together. Others depend on specific visual or acoustic details that were lost when clips were compressed into text. Since the required evidence depends on the question, retrieval needs to adapt its search strategy and, when necessary, revisit the original video or audio to recover missing details.

To tackle the limitations of existing video memory systems, we propose Continuous Adaptive Memory (CAM), which is illustrated in Figure 1(d). In particular, CAM uses a temporally grounded, heterogeneous knowledge graph to store the extracted entities and relations along with their associated facts. To capture high-level semantics, CAM introduces evidence-driven summarization. Rather than operating at fixed time intervals, summarization is treated as an asynchronous operation dictated by subgraph maturity. A localized history is summarized into high-level traits or relational edges only when a specific entity or interaction pair accumulates a sufficient density of incident edges or attributes. This ensures that the summaries are grounded on stable, diverse observations rather than transient anomalies, which improves memory quality.

Besides similarity search, CAM allows multiple additional search methods to retrieve the facts required by the target question, such as traversing the knowledge graph, re-watching a segment of the video, listening to the audio, and counting the frequencies of entities and events. These search methods are more flexible and can retrieve the fine-grained details that are lost during memory construction. To jointly utilize these search methods, CAM proposes an adaptive retrieval workflow, where a planner configures the queries for all search methods, an executor runs each search method, and a verifier checks if the retrieved contents can answer the target question. The plan-execute-verify workflow can also run iteratively for multiple rounds to pinpoint the contents to retrieve.

To evaluate CAM, we conduct experiments on three long video benchmarks and compare with four SOTA video memory systems along with strong base models and online services. The results show that CAM consistently outperforms all baselines in question answering accuracy, and compared with the best-performing baseline, CAM improves accuracy by 12.93 percentage points on average and up to 23.1 percentage points. The ablation study shows that our designs are effective in improving accuracy, and profiling shows that CAM uses 46.2% fewer reasoning tokens per question than WorldMM and 49.7% fewer than M3-Agent.

To summarize, we make the following contributions in this paper.

- To capture high-level semantics, we model the entities or relations extracted from video clips using a heterogeneous knowledge graph and summarize each entity and relation once the local subgraph of the target entity and relation reaches a sufficient size.
- To retrieve the fine-grained details required by each question, we propose a rich set of search methods and utilize these search methods adaptively with an iterative workflow.
- Experiments on three long-video benchmarks demonstrate accuracy gains of up to 23.1 percentage points over the strongest evaluated baseline, with ablation and sensitivity studies assessing the effects of key design choices.

2 Preliminary and Related Work

We study complex question answering over entity-centric long videos, where recurring entities, such as humans, robots, objects, or conversational participants, persist and interact across clips. This setting requires maintaining entity continuity, tracking evolving relations, and answering questions from fine-grained episodic evidence and emergent high-level semantics. Because an entire long video cannot fit into the context window of an MLLM, the key challenge is to build a persistent memory that preserves multimodal details while supporting long-horizon memorization and retrieval.

Long video understanding methods commonly use memory-centric designs or retrieval-augmented generation to overcome context limits [9, 10]. Prior systems rely on flat summaries [8, 5], sequential episodic memories [11], or fixed-interval updates, which are less suitable for entity-centric videos whose semantics emerge asynchronously from accumulated interactions. Graph-based methods introduce more structured representations [12–14], but often use graph topology mainly as an auxiliary retrieval index. CAM instead builds a temporally grounded heterogeneous semantic graph, abstracts mature local subgraphs, and uses a tool-augmented agent for graph traversal, targeted retrieval, and video rewatching. A comprehensive review is provided in Appendix C.

3 Continuous Adaptive Memory

CAM consists of two stages: memorization and adaptive querying. During memorization, CAM processes video clips sequentially, associates entities across clips, and stores their actions, conversations, and interactions in a heterogeneous graph. Each observation retains its temporal context. As new evidence accumulates, CAM updates character- and relationship-level abstractions using the previous summary and newly collected observations, while preserving the underlying episodic evidence.

At query time, an agent gathers evidence through an iterative planner–executor–verifier workflow. It searches the graph for relevant abstractions, episodic facts, and conversations, and invokes temporal search, graph traversal, or counting as needed. When the stored memory lacks sufficient detail, the agent can revisit the corresponding video clips or audio. We first describe memory construction and abstraction, followed by the querying workflow and its tools.

3.1 Memorization

3.1.1 Memory Graph Construction

Clip-level observation. The input video is segmented into fixed-length clips $\{c_1, \ldots, c_n\}$, each treated as an episodic observation unit. To preserve temporal continuity and enable consistent entity tracking, each clip c_i is processed together with appearance descriptors from the preceding clip A_{i-1}, providing visual cues for cross-clip entity association. Given these inputs, an MLLM extracts complementary information from each clip:

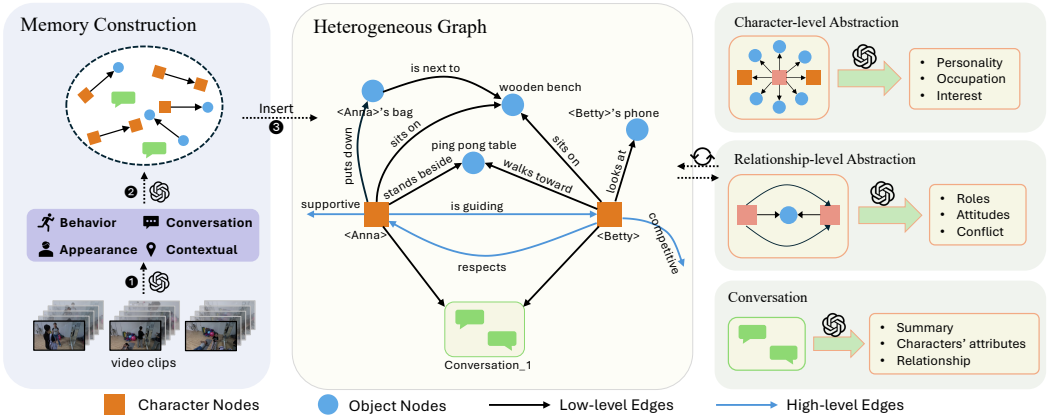


Figure 2: Memory construction in CAM. Clip-level observations are aligned across time and stored in a heterogeneous graph. Character and relationship abstractions are updated as new evidence accumulates, while low-level observations remain available for retrieval.

$$\big(B_i,\ S_i,\ A_i,\ K_i\big) \ \leftarrow\ \text{MLLM}\big(c_i,\ A_{i-1}\big).$$

Here, B_i captures observable character behaviors, S_i records dialogue transcripts, A_i encodes appearance-based descriptors, and K_i stores auxiliary context for later reasoning. Using A_i, characters are aligned across clips to ensure consistent identities over time, even under viewpoint changes, partial occlusions, or temporal gaps. After alignment, the behavior descriptions B_i are transformed by an LLM into structured relational triples $(\textit{source},\ \textit{relation},\ \textit{target})$. Each triple represents an observed interaction and serves as a minimal, interpretable unit of episodic evidence.

Heterogeneous memory graph integration. All extracted signals are integrated into a heterogeneous memory graph

$$G = (V, E), \qquad V = V_{\text{char}} \cup V_{\text{obj}} \cup V_{\text{conv}},$$

which serves as the core long-term video representation. Character and object nodes correspond to aligned entities across clips, while each transcript S_i is represented as a conversation node linked to its speakers, enabling joint modeling of physical actions and dialogue with distinct semantic roles.

Relational triples extracted from the i-th clip are inserted into the graph as temporally annotated edges:

$$e_{ij} = \big(\textit{source},\ \textit{relation}_j,\ \textit{target},\ t = i,\ K_i\big),$$

where $\textit{relation}_j$ is the j-th relation extracted from c_i. Temporal indices and contextual metadata preserve the video's chronological structure, supporting time-aware retrieval, causal reasoning, and episodic reconstruction. Overall, the memorization stage converts a long video into a structured, time-anchored semantic graph that underpins later abstraction and reasoning.

3.1.2 Abstract Information Extraction

Clip-level extraction records local actions and conversations, whereas character traits and social relationships emerge across multiple observations. Evidence also accumulates unevenly: a principal character may appear in many consecutive clips, while a secondary character appears only occasionally. CAM therefore updates high-level abstractions according to the evidence accumulated by each character or character pair, rather than at a shared temporal interval.

Character-level evidence. For each character $v \in V_{\text{char}}$, let $\mathcal{E}_t(v)$ denote its incident episodic edges after processing clip c_t:

$$\mathcal{E}_t(v) = \{e \in E_t \mid e.\text{source} = v \text{ or } e.\text{target} = v\}.$$

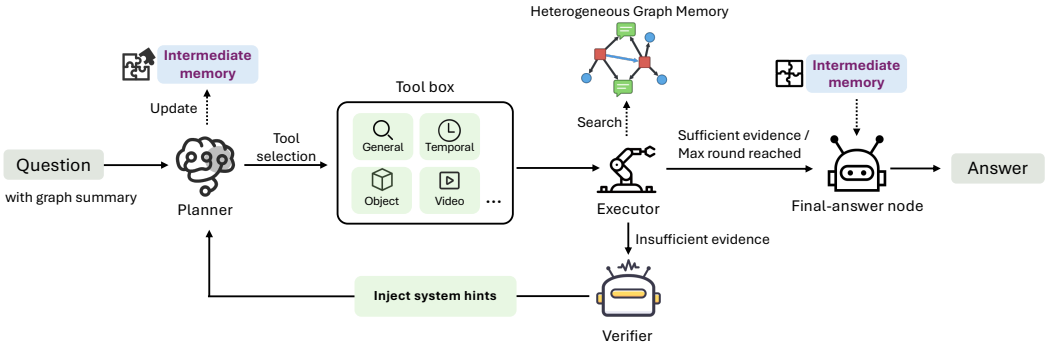


Figure 3: The adaptive reasoning workflow of CAM. This iterative feedback loop enables the system to dynamically refine its search strategy based on intermediate findings, ensuring that complex multi-step queries are resolved with high precision and minimal retrieval overhead.

These edges collect the character's actions, interactions, and contextual states. Their abstraction describes behavioral tendencies across observations.

Relationship-level evidence. For a character pair (u, v), the evidence set combines direct interactions and indirect co-occurrences:

$$\mathcal{E}_t(u, v) = \mathcal{E}_{t,\text{direct}}(u, v) \cup \mathcal{E}_{t,\text{indirect}}(u, v).$$

Indirect evidence includes involvement with the same object, location, or third entity within a clip or short temporal window. The resulting abstraction describes persistent relations between the two characters.

Update scheduling and recurrent abstraction. Both levels follow the same update rule. Let x denote a character or character pair, and let $b(x)$ be its evidence count immediately after the most recent abstraction, initialized to zero. An update is triggered when

$$|\mathcal{E}_t(x)| - b(x) \geq \tau_x,$$

where $\tau_x = \tau_{\text{char}}$ for characters and $\tau_x = \tau_{\text{rel}}$ for pairs. Thus, the threshold applies to newly accumulated edges, and each character or pair follows its own update schedule.

When triggered, the LLM receives the previous summary $h_{\text{prev}}(x)$ and the new evidence $\Delta\mathcal{E}_t(x)$, including temporal and contextual annotations:

$$h_{\text{new}}(x) = \text{LLM}\big(h_{\text{prev}}(x), \Delta\mathcal{E}_t(x)\big).$$

For the first update, the previous summary is empty. The new summary updates the character-level abstract node or relationship-level edge, and the counter advances to $b(x) \leftarrow |\mathcal{E}_t(x)|$. Original episodic edges are retained, allowing later queries to access the observations underlying an abstraction. Edge counts provide a lightweight scheduling proxy, while the abstraction model infers semantic content from the accumulated evidence and previous summary. We use $\tau_{\text{char}} = 50$ and $\tau_{\text{rel}} = 30$ across all three benchmarks without dataset-specific tuning. Further details appear in Appendix B.2.

3.2 Adaptive Reasoning

3.2.1 Reasoning Workflow

To effectively navigate the structured memory graph $\mathcal{G}$ and answer complex, multi-hop queries, we formulate the reasoning process as an iterative, tool-augmented state machine implemented in LangGraph, as depicted in Figure 3. Let the agent's state at turn t be denoted as $\mathcal{S}_t = (\mathcal{Q}, \mathcal{C}_{meta}, \mathcal{M}_t, \mathcal{H}_t)$. Here, $\mathcal{Q}$ represents the user query, and $\mathcal{H}_t = \{o_1, o_2, \ldots, o_{t-1}\}$ is the history of raw evidence returned by tool calls up to turn t.

Crucially, $\mathcal{C}_{meta}$ represents the global graph metadata provided to the agent at initialization ($t = 0$). This gives the agent a macroscopic overview of the context before execution, encompassing the video timespan, a list of main characters, the total number of extracted knowledge triples, and

available conversations. Furthermore, $\mathcal{M}_t$ denotes a dynamically updated intermediate memory that synthesizes findings across turns.

The agent systematically gathers evidence through a controlled loop of planning, execution, and self-correction:

1. **Planner Node (LLM):** Acting as the central controller, the LLM analyzes the comprehensive current state $\mathcal{S}_t$, including the macro-level $\mathcal{C}_{meta}$ and the progressive deductions in $\mathcal{M}_t$, to determine the optimal next action $a_t \in \mathcal{A}$ and its corresponding arguments:
$$a_t, \text{args}_t = \text{LLM}(\mathcal{S}_t)$$
2. **Executor Node:** This module invokes the selected tool a_t to interact with the environment (the memory graph $\mathcal{G}$ or raw video). The execution yields a tool output o_t containing new empirical evidence:
$$o_t = \text{Execute}(a_t, \text{args}_t, \mathcal{G})$$
Following execution, the history is updated ($\mathcal{H}_{t+1} = \mathcal{H}_t \cup \{o_t\}$), and the intermediate memory $\mathcal{M}_{t+1}$ updates to logically assimilate o_t, accumulating a structured chain of thought that gradually approaches the final answer.
3. **Task Completion & Verifier Node:** A verification function $\mathcal{V}$ evaluates the intermediate memory $\mathcal{M}_{t+1}$ to determine if the accumulated context resolves the query:
$$\mathcal{V}(\mathcal{S}_{t+1}) \rightarrow \{\text{Complete}, \text{Incomplete}, \text{Stuck}\}$$
If incomplete and the max round is not reached, the loop iterates. If the state is classified as "Stuck" (e.g., detecting cyclic tool calls where $a_t = a_{t-1}$), the Verifier injects system-level heuristic hints h into the state before routing back to the Planner, dynamically regularizing the LLM's search trajectory.
4. **Final Answer Node:** The iterative process terminates when $\mathcal{V}(\mathcal{S}_{t+1}) = \text{Complete}$ or the budget is exhausted. The final state $\mathcal{S}_T$ transitions to the final answer node, where the collected evidence is organized to deduce the final answer $\hat{y}$. For multiple-choice questions, the output is constrained to a single categorical letter. For open-ended generative questions, the node structures the final output as a concise, single-sentence short answer.

3.2.2 Graph and Multimodal Retrieval Tools

To empower the agent to actively navigate the heterogeneous memory graph and ground its reasoning in raw multimodal signals, we define a comprehensive suite of specialized tools.

The foundational mechanism is the **General Search** tool. To handle queries requiring both global relations and fine-grained details, its logic is divided into three stages:

1) Weighted Query Decomposition. The Planner Node parses the query q into a semantic triple $\mathcal{T}_q = \langle s_q, r_q, t_q \rangle$ with a normalized weight vector $\mathbf{W}_q$. It extracts auxiliary context I_q and determines a channel-specific retrieval budget $\mathbf{K}_q$.

2) Triple Similarity Computation. To evaluate relevance within the graph $\mathcal{G}$, we compute similarities in a dense embedding space. To account for reversed semantic roles (e.g., subject vs. object), we compute a bidirectional alignment score for the entities and integrate it with the direct semantic matching of the relation to calculate the overall structural similarity:
$$\begin{aligned} S_{\text{struct}} = &\max\left(w_s \cdot \text{sim}(s_q, s_c) + w_t \cdot \text{sim}(t_q, t_c),\ w_s \cdot \text{sim}(s_q, t_c) + w_t \cdot \text{sim}(t_q, s_c)\right) \\ &+ w_r \cdot \text{sim}(r_q, r_c). \end{aligned}$$

3) Multi-Channel Retrieval. Retrieval is partitioned by semantic roles. High-level abstractions are retrieved via S_{struct}. Low-level episodic relations require spatio-temporal grounding, scoring candidates via $S_{\text{struct}} + \lambda \cdot sim(I_q, K_c)$, where K_c denotes the contextual metadata attached to the candidate relation and λ weights the context similarity. Conversational content relies on direct semantic matching against the query, expanding retrieved utterances to include chronological neighbors (d_{i-1}, d_{i+1}) to preserve discursive coherence. All retrieved evidence is merged to flexibly integrate abstract, episodic, and dialogue contexts.

Beyond this primary search function, the agent is equipped with six additional specialized tools to facilitate explicit topological traversal, targeted adaptive querying, and raw sensory verification:

- **Temporal Search:** constrains retrieval to a window (t_begin, t_end).
- **Graph Traversal:** retrieves semantically filtered neighborhoods around an anchor entity for multi-hop deduction.
- **Search Object:** embedding-based entity alignment to handle vocabulary mismatch.
- **Frequency Count:** aggregates occurrence counts of actions/objects/events.
- **Video Rewatch:** Acts as a critical multimodal fallback by directly rewatching the video to recover subtle visual nuances and continuous motion dynamics lost during text compression.
- **Audio Listening:** samples raw audio for non-speech acoustic cues.

A more detailed explanation of each tool is provided in Appendix B.3.

3.3 Discussion

CAM is designed for entity-centric long videos, such as robotic operations and character-driven narratives, where recurring entities and evolving relations provide natural units for memory organization and abstraction. Our evaluation focuses on this setting. For less entity-centric videos, such as lectures or advertisements, the benefits may be more limited and have not been established by our experiments.

Memory quality depends on the underlying models' extraction and entity-association accuracy. The noise-injection study evaluates robustness to randomly mismatched evidence, but does not cover systematic or correlated extraction errors. Since useful tools can vary across domains and question types, future work may explore dynamic tool selection to better balance robustness, accuracy, and inference efficiency.

4 Experimental Evaluation

4.1 Experiment Settings

Datasets and Metrics. We evaluate on three entity-centric long-video benchmarks. **M3-Bench-robot** (averaging 34 minutes, robotic views) and **M3-Bench-web** (27 minutes, YouTube) assess GPT-5.2-judged accuracy across five reasoning categories: multi-evidence (ME), multi-hop (MH), cross-modal (CM), person understanding (PU), and general knowledge (GK). On all 1,276 M3-Bench-robot predictions, the GPT-5.2 judge agrees with GPT-4o on 96.7% and with human annotations on 97.7% of judgments. The evaluation protocol and validation details appear in Appendix A. Finally, **HippoVlog** features 25 videos (averaging 27.2 minutes) and is evaluated using 1,000 multiple-choice questions.

Baselines and Implementation. We compare CAM against a comprehensive suite of base video LLMs (GPT-5.2, Gemini-3-Pro, Qwen3-VL-flash), streaming models (MovieChat, MA-LMM, Flash-VStream), and memory-based frameworks (EgoRAG, HippoMM, M3-Agent, WorldMM). For fair comparison, CAM and all memory-based baselines, except HippoMM, which exclusively uses Qwen, utilize GPT-5.2 as the core backbone, while CAM additionally evaluates Qwen3-VL-flash for video and Qwen3.5-flash for reasoning variants. HippoMM is restricted to Qwen because its extensive preprocessing steps, such as dynamic segmentation, would make GPT-based processing computationally prohibitive and prevent the system from processing videos faster than real time. Videos are sampled at 1 fps and processed in 30-second fixed intervals (dynamically segmented 5-10-second clips for HippoMM), with audio transcribed into subtitles for unified multimodal reasoning. Comprehensive implementation details are provided in Appendix A.

4.2 Main Results

Table 1 reports the performance of CAM against a wide range of base models and state-of-the-art long-video understanding methods on three benchmarks: M3-Bench-robot, M3-Bench-web, and HippoVlog. Overall, CAM consistently achieves the best results across reasoning categories and datasets, demonstrating the effectiveness of hierarchical, graph-structured video memory for long-horizon reasoning.

Table 1: Question answering accuracy. All video memory methods use GPT as the MLLM except HippoMM due to efficiency considerations. Besides GPT, we also report the accuracy of our CAM when using Qwen as the MLLM to demonstrate generality.

Method	**M3-Bench-robot**						**M3-Bench-web**						**HippoVlog**
	ME	MH	CM	PU	GK	**All**	ME	MH	CM	PU	GK	**All**	
Base Models													
Qwen3-VL-flash	9.3	9.4	8.4	10.7	7.7	9.0	11.9	10.5	13.4	14.0	20.9	14.9	24.2
Gemini-3-Pro	15.5	17.6	14.1	24.4	7.3	15.3	18.0	17.9	23.8	23.1	28.7	23.2	38.0
GPT-5.2	18.3	16.5	18.7	29.5	11.0	18.0	21.3	21.9	30.9	27.1	39.6	28.7	38.8
Online Video Understanding Methods													
MovieChat	13.3	9.8	12.2	15.7	7.0	11.2	12.2	6.6	12.5	17.4	11.1	12.6	32.1
MA-LMM	25.6	23.4	22.7	39.1	14.4	24.4	26.8	10.5	22.4	39.3	15.8	24.3	40.4
Flash-VStream	21.6	19.4	19.3	24.3	14.1	19.4	24.5	10.3	24.6	32.5	20.2	23.6	38.2
Video Memory-based Methods													
EgoRAG	22.6	28.2	21.0	34.0	11.4	21.9	23.1	15.2	34.6	31.1	48.3	32.8	57.5
HippoMM	41.2	51.8	40.3	56.5	24.8	40.4	38.8	30.8	21.3	47.3	48.6	40.7	71.9
M3-Agent	42.4	56.5	43.9	60.4	26.9	41.4	45.9	28.4	44.3	59.3	53.9	48.9	65.5
WorldMM	34.4	55.3	36.6	43.2	21.7	34.5	59.2	42.1	51.7	69.1	58.4	58.1	71.5
CAM (GPT)	**65.5**	**81.2**	**66.8**	**83.8**	**36.7**	**64.5**	**64.6**	**54.7**	**52.1**	**75.6**	**75.5**	**69.3**	**76.4**
CAM (Qwen)	51.2	63.5	51.5	70.3	33.4	51.3	57.2	44.4	51.0	58.3	41.8	55.6	73.8

M3-Bench-robot is the most challenging benchmark evaluated, featuring a high density of complex multi-hop questions and tasks that require deep character understanding. On tasks requiring such extended long-term dependencies, purely streaming models like MovieChat, MA-LMM, and Flash-VStream trail significantly behind memory-based methods. Accordingly, CAM obtains the largest gains over baselines on this dataset. CAM (GPT) achieves an overall accuracy of 64.5%, outperforming the strongest prior memory-based baseline, M3-Agent, by 23.1 percentage points. Notably, even with the smaller Qwen backbone, CAM achieves a highly competitive accuracy of 51.3%, still surpassing all baselines. Although the Qwen variant does not match the peak performance of the GPT model, it offers an efficient alternative, operating at substantially lower inference cost compared with GPT. CAM (GPT) achieves 81.2% on multi-hop reasoning (MH) and 83.8% on person understanding (PU), improving over M3-Agent by 24.7 and 23.4 percentage points, respectively. While stronger memory agents such as M3-Agent and WorldMM attempt to address these areas, their reliance on premature semantic abstraction or disjointed memory retrieval restricts their ability to support complex relational and multi-hop reasoning. In fact, WorldMM struggles noticeably on M3-Bench-robot, suggesting its segregated memory design is poorly suited to benchmarks requiring strict character consistency and identity tracking.

On M3-Bench-web, which contains more diverse and unconstrained YouTube videos, CAM again establishes a clear lead with an overall score of 69.3%, surpassing the strongest baseline, WorldMM, by 11.2 percentage points. Specifically, CAM shows strong improvements in person understanding (PU) and general knowledge extraction (GK), indicating its ability to integrate dispersed factual cues and identity-related information across long temporal spans. Although the overall performance gap is narrower than in the egocentric setting, CAM's consistent advantage shows that its structured memory remains robust under higher visual and semantic variability.

Furthermore, on HippoVlog, which tests continuous audiovisual understanding, CAM achieves a state-of-the-art accuracy of 76.4%. While baselines like M3-Agent and HippoMM tend to overfit to their native benchmarks, and flat-memory methods (EgoRAG, HippoMM) suffer from lossy text compression, CAM generalizes effectively. It outperforms domain-specific models like HippoMM (71.9%) and WorldMM (71.5%) without any format tailoring, suggesting that its temporally grounded graph robustly captures complex vlog narratives. Detailed reasoning statistics are in Appendix D.

Table 2: Ablation study for the designs of CAM. Structured graph memory and adaptive querying tools both contribute critically to overall reasoning performance.

	M3-Bench-robot					
Method	ME	MH	CM	PU	GK	All
Ablation on memory construction						
Fixed-interval summarization	61.4	77.6	62.6	82.9	35.8	60.3
CAM w/o graph	43.4	56.8	47.9	57.9	27.3	44.8
CAM w/o high-level	54.6	66.9	51.5	79.9	29.1	55.7
Ablation on question answering						
similarity search only	56.6	73.1	57.2	80.2	34.3	56.5
general search + video rewatch	59.5	75.3	58.8	82.3	29.2	57.8
decreasing top-k to 30	63.4	76.5	64.3	83.0	34.9	62.9
CAM-full	**65.5**	**81.2**	**66.8**	**83.8**	**36.7**	**64.5**

4.3 Ablation Study

We conduct ablations to examine CAM's memory construction and question-answering designs. For memory construction, we compare CAM with three variants: (1) **CAM w/o graph**, which removes the heterogeneous graph and reduces retrieval to unstructured semantic matching; (2) **CAM w/o high-level**, which keeps only low-level episodic edges and disables entity- and relation-level summaries; and (3) **fixed-interval abstraction**, which generates high-level summaries every 10 minutes instead of using evidence-driven subgraph maturity. For question answering, we evaluate three constrained reasoning settings: (1) similarity search only, (2) general search with video rewatch, and (3) a reduced search budget that decreases top-k from 50 to 30.

As shown in Table 2, removing the graph structure causes the largest drop, reducing the overall score from 64.5% to 44.8%. The degradation is especially severe on Multi-Hop (81.2% to 56.8%) and Multi-Evidence reasoning (65.5% to 43.4%), showing that unstructured memory and similarity matching cannot effectively capture the spatio-temporal dependencies in long videos. Removing high-level abstraction also hurts performance, dropping the overall score to 55.7%. Cross-Modal accuracy decreases from 66.8% to 51.5%, and Multi-Hop accuracy decreases from 81.2% to 66.9%, drops of 15.3 and 14.3 percentage points, respectively. This indicates that episodic observations alone are insufficient, and entity- or relation-level summaries are necessary for connecting scattered evidence into coherent semantics. Moreover, fixed-interval abstraction reaches only 60.3%, confirming that summarizing at arbitrary temporal boundaries can introduce premature or unreliable abstractions, whereas CAM's evidence-driven abstraction yields more mature and consistent high-level memory.

For question answering, restricting the agent to similarity search yields 56.5%, and adding video rewatch only slightly increases it to 57.8% while significantly raising token costs. These results confirm that naive retrieval cannot replace CAM's specialized graph-aware tools, such as object search and graph traversal. Finally, reducing the search budget to top-$k = 30$ causes a minor drop to 62.9%, showing robustness under tighter retrieval constraints while benefiting from wider coverage for optimal reasoning.

4.4 Abstraction Threshold Sensitivity

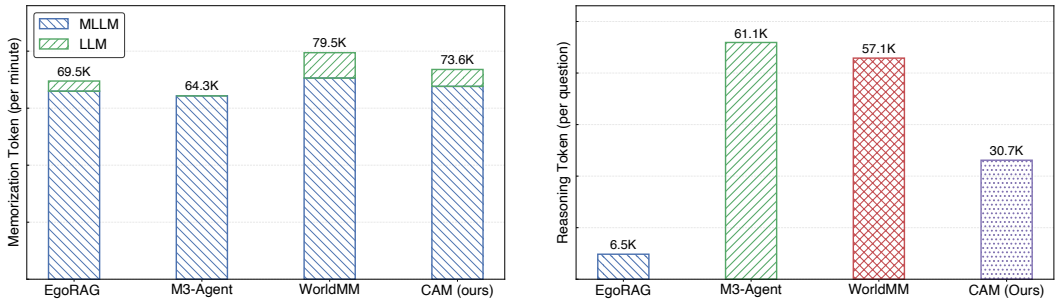
We jointly vary the character-level threshold τ_{char} and the relationship-level threshold τ_{rel} to examine the effect of abstraction frequency. Table 3 compares three threshold configurations with fixed-interval abstraction on both M3-Bench domains. The default configuration, $\tau_{\text{char}} = 50$ and $\tau_{\text{rel}} = 30$, gives the highest overall accuracy in both domains. Reducing the thresholds to $\tau_{\text{char}} = 30$ and $\tau_{\text{rel}} = 10$ lowers accuracy by 2.6 percentage points on Robot and 1.8 points on Web. Increasing them to $\tau_{\text{char}} = 100$ and $\tau_{\text{rel}} = 60$ lowers accuracy by 1.4 and 1.2 points, respectively.

All three evidence-driven configurations outperform fixed-interval abstraction, supporting the use of entity-specific update schedules across the tested range. On Robot, the default configuration also gives the highest accuracy in all five reasoning categories. Category-level optima vary on Web, but the default configuration achieves the highest overall accuracy on both domains.

Table 3: Accuracy (%) under different abstraction settings on M3-Bench-robot and M3-Bench-web. Threshold pairs denote $\tau_{\text{char}}/\tau_{\text{rel}}$. The fixed-interval variant does not use evidence thresholds.

Setting ($\tau_{\text{char}}/\tau_{\text{rel}}$)	**M3-Bench-robot**						**M3-Bench-web**					
	ME	MH	CM	PU	GK	**All**	ME	MH	CM	PU	GK	**All**
Fixed-interval	61.4	77.6	62.6	82.9	35.8	60.3	58.6	50.0	**56.3**	75.6	71.7	66.9
30/10	63.6	75.3	63.8	83.0	35.5	61.9	63.6	**57.8**	47.9	74.0	67.9	67.5
50/30 (default)	**65.5**	**81.2**	**66.8**	**83.8**	**36.7**	**64.5**	**64.6**	54.7	52.1	75.6	**75.5**	**69.3**
100/60	63.7	76.5	64.7	83.2	35.8	63.1	61.6	51.6	50.0	**76.3**	73.6	68.1

4.5 Efficiency Analysis



(a) Memorization token consumption

(b) Reasoning token consumption

Figure 4: Token consumption of CAM and the evaluated video memory baselines. (a) Memorization cost, measured in tokens per minute of video. (b) Reasoning cost, measured in tokens per question.

Token consumption. Figure 4 compares token consumption during memorization and reasoning. CAM uses 73.6K tokens per video minute for memorization, compared with WorldMM's 79.5K, a reduction of 7.4%. During reasoning, CAM uses 30,711 tokens per question, compared with 57,064 for WorldMM and 61,100 for M3-Agent, reductions of 46.2% and 49.7%, respectively. EgoRAG uses fewer reasoning tokens (6,455 per question), with lower QA accuracy. CAM thus achieves higher accuracy than the evaluated baselines while using fewer reasoning tokens than WorldMM and M3-Agent.

Reasoning latency. On M3-Bench-robot, CAM-full averages 47.3 seconds per question, compared with 81.4 seconds for M3-Agent and 66.7 seconds for WorldMM. Reducing the retrieval budget to top-$k = 30$ lowers latency to 40.1 seconds and reasoning consumption to 24,002 tokens per question, reductions of 15.2% and 21.8% relative to CAM-full, respectively. Accuracy decreases from 64.5% to 62.9%. Further details appear in Appendix D.

5 Conclusion

In this paper, we study reliable memory construction for complex question answering over long-form videos. We propose CAM, a structured video memory framework that organizes clip-level observations into a temporally grounded heterogeneous graph. By continuously abstracting mature subgraphs into high-level semantics and adaptively invoking graph-based and multimodal retrieval tools, CAM bridges the gap between low-level evidence and high-level reasoning, achieving strong performance on complex multi-hop entity-centric video QA benchmarks.

Acknowledgments

We thank the reviewers for their constructive comments and suggestions. This project was partially supported by a CRF (No. C2005-24Y) from the RGC of Hong Kong. This work was also sponsored by National Natural Science Foundation of China (62672365).

References

[1] OpenAI. Gpt-5 system card, August 2025. URL https://openai.com/index/gpt-5-system-card/.

[2] Sundar Pichai, Demis Hassabis, and Koray Kavukcuoglu. A new era of intelligence with Gemini 3, November 2025. URL https://blog.google/products-and-platforms/products/gemini/gemini-3/. Accessed: 2026-05-07.

[3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

[4] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

[5] Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, Bei Ouyang, Zhengyu Lin, Marco Cominelli, Zhongang Cai, Bo Li, Yuanhan Zhang, Peiyuan Zhang, Fangzhou Hong, Joerg Widmer, Francesco Gringoli, Lei Yang, and Ziwei Liu. Egolife: Towards egocentric life assistant. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28885–28900, June 2025.

[6] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. Lvbench: An extreme long video understanding benchmark, 2024. URL https://arxiv.org/abs/2406.08035.

[7] Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. WorldMM: Dynamic multimodal memory agent for long video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25599–25609, June 2026.

[8] 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. URL https://openreview.net/forum?id=PMz29A7Muq.

[9] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18221–18232, June 2024.

[10] Yongdong Luo, Xiawu Zheng, Guilin Li, Shukang Yin, Haojia Lin, Chaoyou Fu, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, and Rongrong Ji. Video-rag: Visually-aligned retrieval-augmented long video comprehension. In *Advances in Neural Information Processing Systems*, 2025.

[11] Yueqian Lin, Qinsi Wang, Hancheng Ye, Yuzhe Fu, Hai "Helen" Li, and Yiran Chen. Hippomm: Hippocampal-inspired multimodal memory for long audiovisual event understanding, 2025. URL https://arxiv.org/abs/2504.10739.

[12] Meng Chu, Yicong Li, and Tat-Seng Chua. Understanding long videos via llm-powered entity relation graphs, 2025. URL https://arxiv.org/abs/2501.15953.

[13] Xubin Ren, Lingrui Xu, Long Xia, Shuaiqiang Wang, Dawei Yin, and Chao Huang. Videorag: Retrieval-augmented generation with extreme long-context videos, 2025. URL https://arxiv.org/abs/2502.01549.

[14] Sameer Malik, Moyuru Yamada, Ayush Singh, and Dishank Aggarwal. Ravu: Retrieval augmented video understanding with compositional reasoning over graph, 2025. URL https://arxiv.org/abs/2505.03173.

[15] Garry Yang, Zizhe Chen, Man Hon Wong, Haoyu Lei, Yongqiang Chen, Zhenguo Li, Kaiwen Zhou, and James Cheng. Mesh - understanding videos like human: Measuring hallucinations in large video models. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pages 4827–4836, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3755626. URL https://doi.org/10.1145/3746027.3755626.

[16] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13504–13514, June 2024.

[17] Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. In *Advances in Neural Information Processing Systems*, 2024.

[18] Jinmeng Wu, Pengcheng Shu, Hanyu Hong, Lei Ma, Ying Zhu, and Lei Wang. Pre-trained Bidirectional Dynamic Memory Network for Long Video Question Answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 5550–5557, 2024.

[19] Ivana Balazevic, Yuge Shi, Pinelopi Papalampidi, Rahma Chaabouni, Skanda Koppula, and Olivier J. Henaff. Memory consolidation enables long-context video understanding. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 2527–2542. PMLR, 2024. URL https://proceedings.mlr.press/v235/balazevic24a.html.

[20] Anxhelo Diko, Tinghuai Wang, Wassim Swaileh, Shiyan Sun, and Ioannis Patras. ReWind: Understanding long videos with instructed learnable memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13734–13743, June 2025.

[21] 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)*, pages 21059–21069, October 2025.

[22] Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 3272–3283, June 2025.

[23] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. TimeChat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14313–14323, June 2024.

[24] Bao Tran Gia, Khiem Le, Tien Do, Tien-Dung Mai, Thanh Duc Ngo, Duy-Dinh Le, and Shin'ichi Satoh. Vrag: Retrieval-augmented video question answering for long-form videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3728–3737, 2025.

[25] Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Mingchen Zhuge, Jian Ding, Deyao Zhu, Jürgen Schmidhuber, and Mohamed Elhoseiny. Goldfish: Vision-language understanding of arbitrarily long videos. In *European Conference on Computer Vision (ECCV)*, pages 251–267. Springer, 2024. doi: 10.1007/978-3-031-73397-0_15.

[26] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. VideoAgent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision (ECCV)*, pages 58–76. Springer, 2024. doi: 10.1007/978-3-031-72989-8_4.

[27] Noriyuki Kugo, Xiang Li, Zixin Li, Ashish Gupta, Arpandeep Khatua, Nidhish Jain, Chaitanya Patel, Yuta Kyuragi, Yasunori Ishii, Masamoto Tanabiki, Kazuki Kozuka, and Ehsan Adeli. Videomultiagents: A multi-agent framework for video question answering, 2025. URL https://arxiv.org/abs/2504.20091.

[28] Runtao Liu, Ziyi Liu, Jiaqi Tang, Yue Ma, Renjie Pi, Jipeng Zhang, and Qifeng Chen. Longvideoagent: Multi-agent reasoning with long videos, 2025. URL https://arxiv.org/abs/2512.20618.

[29] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization, 2024. URL https://arxiv.org/abs/2404.16130.

[30] Qihui Zhou, Peiqi Yin, Xiao Yan, Changji Li, Guanxian Jiang, and James Cheng. Atom: An efficient query serving system for embedding-based knowledge graph reasoning with operator-level batching. *Proceedings of the ACM on Management of Data*, 2(4):193:1–193:29, 2024. doi: 10.1145/3677129. URL https://doi.org/10.1145/3677129.

[31] Petr Anokhin, Nikita Semenov, Artyom Sorokin, Dmitry Evseev, Andrey Kravchenko, Mikhail Burtsev, and Evgeny Burnaev. Arigraph: Learning knowledge graph world models with episodic memory for llm agents. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 12–20, 2025. doi: 10.24963/ijcai.2025/2. URL https://www.ijcai.org/proceedings/2025/0002.

[32] Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory, 2025. URL https://arxiv.org/abs/2501.13956.

[33] Zijie Dai, Shiyuan Deng, Sheng Guan, Yizhou Tian, Xin Yao, Xiao Yan, and James Cheng. Recmem: Recurrence-based memory consolidation for efficient and effective long-running llm agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 32353–32376, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1619. URL https://aclanthology.org/2026.findings-acl.1619/.

[34] Ziyi Bai, Ruiping Wang, and Xilin Chen. Glance and focus: Memory prompting for multi-event video question answering. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[35] Xiaoqian Shen, Wenxuan Zhang, Jun Chen, and Mohamed Elhoseiny. Vgent: Graph-based retrieval-reasoning-augmented generation for long video understanding. In *Advances in Neural Information Processing Systems*, 2025.

[36] Shulin Tian, Ruiqi Wang, Hongming Guo, Penghao Wu, Yuhao Dong, Xiuying Wang, Jingkang Yang, Hao Zhang, Hongyuan Zhu, and Ziwei Liu. Ego-r1: Chain-of-tool-thought for ultra-long egocentric video reasoning, 2025. URL https://arxiv.org/abs/2506.13654.

Appendix Contents

A Experimental Setup and Implementation Details **14**

A.1 Datasets and Metrics 14

A.2 Baselines 15

A.3 Implementation Details 16

B Additional Method Details **16**

B.1 Natural-Language Entity Tracking 16

B.2 Evidence-Driven Abstraction 17

B.3 Agent Toolbox 18

C Additional Related Work **19**

D Additional Experiments and Analysis **20**

D.1 Abstraction Threshold Sensitivity 20

D.2 Reasoning Efficiency 20

D.3 Tool-Use Statistics 21

D.4 Robustness to Graph Noise 21

D.5 Streaming and Reasoning Efficiency 22

D.6 Graph Visualization 23

E Prompt Templates **23**

E.1 Memorization Prompts 23

E.2 Reasoning Prompts 28

E.3 Answer Verification 29

F Broader Impacts and Ethical Considerations **30**

F.1 Broader Impacts 30

F.2 Safeguards and Code of Ethics 30

A Experimental Setup and Implementation Details

A.1 Datasets and Metrics

We evaluate our method on three long-video understanding benchmarks: **M3-Bench-robot**, **M3-Bench-web**, and **HippoVlog**. M3-Bench-robot contains 100 long-form videos and 1,276 questions. The videos have an average duration of 34 minutes, are captured from robotic viewpoints, and cover five reasoning categories: multi-evidence reasoning (ME), multi-hop reasoning (MH), cross-modal reasoning (CM), person understanding (PU), and general knowledge extraction (GK). Performance on M3-Bench-robot is evaluated using accuracy, with predictions judged by GPT-5.2. M3-Bench-web follows the same reasoning categories and evaluation protocol as M3-Bench-robot, but is composed

of YouTube videos with an average length of 27 minutes. We select a subset of 100 videos from M3-Bench-web for evaluation. Finally, HippoVlog consists of 25 videos with an average duration of 27.2 minutes and is evaluated using 1,000 multiple-choice questions across four distinct categories: cross-modal (V+A), auditory (A), visual (V), and semantic (S).

Assets and Licenses. The M3-Bench dataset is released under CC BY-NC-SA 4.0, while HippoVlog is distributed under the MIT License. Original videos remain subject to their respective copyright and usage terms. We cite the original publications of all baseline methods and use pretrained models subject to their respective licenses and API terms.

LLM judge protocol and validation. For M3-Bench, we reuse the official M3-Agent evaluator prompt [8], which determines whether the reference answer can be inferred from a prediction and returns only `"Yes"` or `"No"`. Predictions follow the benchmark's concise one-sentence answer format. Reported scores use GPT-5.2 as the judge. GPT-5.2 and the GPT-4o cross-judge receive the same prompt in separate post-generation calls, and the full prompt is provided in Appendix E.3. The answering agent never receives the evaluator prompt, reference answer, or evaluator output during reasoning.

To validate the judge, we independently evaluate all 1,276 M3-Bench-robot predictions with GPT-4o and human annotations:

- **GPT-5.2 vs. GPT-4o:** 96.7% agreement.
- **GPT-5.2 vs. human:** 97.7% agreement.

The resulting accuracies under GPT-5.2, GPT-4o, and human evaluation are 64.5%, 67.2%, and 65.0%, respectively. The human-evaluated accuracy is only 0.5 percentage points above the reported GPT-5.2-judged result, while GPT-4o yields a higher score. These results provide no evidence that using GPT-5.2 to judge answers generated by the same model family inflates the reported accuracy. M3-Agent also reports 96% agreement between GPT-4o and human majority vote in Section 3.3 and Table 18 of its paper [8]. HippoVlog uses exact matching for its multiple-choice answers and therefore requires no LLM judge.

A.2 Baselines

We compare CAM against a comprehensive set of baselines spanning base video LLMs, long video understanding models, RAG systems, and memory-based models. Base video LLMs include GPT-5.2, Gemini-3-Pro, and Qwen3-VL-flash, while long video understanding models include MovieChat, MA-LMM, Flash-VStream. Finally, we compare with memory-based frameworks for long video reasoning, including EgoRAG, HippoMM, M3-Agent, and WorldMM:

- **EgoRAG:** EgoRAG [5] segments videos into short clips and generates hierarchical text captions based on rigid temporal boundaries. To answer questions, it searches these captions using text similarity. While EgoRAG is designed for lifelong egocentric understanding and continuous memory accumulation, its memory structure primarily serves basic retrieval and summarization rather than complex, explicit entity-relation reasoning.
- **HippoMM:** HippoMM utilizes a biologically inspired episodic-semantic memory structure to improve the long-term retention of video content. However, because its memory organization is largely sequential, the model is limited in its ability to effectively handle multi-hop and relational reasoning tasks. It is treated as a special case and exclusively uses Qwen models instead of GPT. This is because HippoMM involves extensive preprocessing steps, including dynamic segmentation and consolidation. Using GPT combined with the required GPU processing for these steps would be highly computationally expensive. Furthermore, it is important that the processing time remains shorter than the length of the video itself, which GPT's overhead would prevent. Consequently, its original experimental setting is maintained.
- **M3-Agent:** This framework explicitly models entities and organizes extracted facts around them to construct an entity-centric memory. It works by storing clip-level textual summaries within an external memory. The original M3-Agent framework was trained using reinforcement learning with a relatively small base model (Qwen2.5-omni-7B). Because this small base model lacks both generality and the capability of handling complex visual tasks across other benchmarks, we evaluate M3-Agent using GPT-5.2 to ensure a fair comparison with the other methods. Its main drawback

is that its retrieval mechanism underutilizes graph topology; it relies on flat textual similarity matching, which struggles to navigate complex relational chains.
- **WorldMM:** WorldMM [7] attempts to capture both temporal events and higher-level semantics by maintaining three separate memory modules: episodic, semantic, and visual. Despite this comprehensive approach, it physically segregates its knowledge into decoupled memory banks. Because the visual features are isolated from the textual knowledge graph, the agent is forced to conduct independent, fragmented retrieval streams instead of achieving a unified multimodal understanding.

A.3 Implementation Details

CAM uses hosted APIs for model inference and requires no local GPU. CAM and all baseline methods (except the three base models and HippoMM) adopt GPT-5.2 as the backbone for both text-only and multimodal processing to ensure a fair comparison. To evaluate the effect of different LLMs on the video memory frameworks, CAM additionally uses Qwen3-VL-flash (MLLM) and Qwen3.5-flash (LLM). HippoMM is treated as a special case and exclusively uses the Qwen models. Because HippoMM involves extensive preprocessing steps, including dynamic segmentation and consolidation, using the GPT model would make the process prohibitively slow, so its original experimental setting is maintained. For video input, EgoRAG, M3-Agent, WorldMM, and CAM all use 30-second fixed-interval clips, whereas HippoMM relies on dynamic video segmentation with clips ranging from 5 to 10 seconds long. Audio streams are transcribed and incorporated into the video representation as subtitles, enabling unified multimodal reasoning, and videos are uniformly sampled at 1 fps. During dialogue retrieval, the sentences immediately preceding and following the matched conversation are also included in the context window, as answers often rely on surrounding conversational context rather than the retrieved utterance alone.

B Additional Method Details

B.1 Natural-Language Entity Tracking

While prior methods such as M3-Agent [8] use InsightFace embeddings for face association, CAM extracts and passes appearance descriptors (A_{i-1}) as natural language. Visual-embedding trackers rely primarily on local appearance similarity and can become unreliable under occlusion, illumination changes, extreme poses, or blurred faces. CAM's streaming MLLM can instead combine clothing, surrounding people and objects, neighboring frames, and previously processed context. When the visual evidence in one clip is ambiguous, later clips can also resolve an earlier provisional identifier through equivalence updates.

Association Quality. We manually annotated 13 videos spanning 6.98 hours and 846 clips, establishing 47 distinct characters as ground-truth identities. We ran the released M3-Agent implementation on the same videos using its provided InsightFace parameters without retuning, then aligned its predicted identity clusters with the ground truth. Table 4 summarizes the results.

Table 4: Character-association errors on the manually annotated subset. "Merge/miss" combines false merges and missed identities. Added latency is measured per 30-second clip on an RTX 4090.

Association method	Total errors	Splits	Merge/miss	Added latency
CAM natural-language association	**14**	14	**0**	**0 s**
InsightFace	18	1	17	8.5 s

Relative to the ground truth, CAM splits 14 of the 47 identities once each, creating 14 excess nodes, but produces no false merges or missed identities. Five of the 13 videos are fully correct, and the resulting fragments contain only 5.2% of all graph edges. InsightFace produces 17 false-merge or missed-identity errors and one split, for a total of 18 association errors. Thus, CAM yields fewer errors overall, and its errors are limited to splits. This distinction matters for graph memory: false merges contaminate an identity with another character's actions, dialogue, and relationships, while

missed identities discard evidence entirely. Splits fragment and delay evidence accumulation but do not transfer evidence across identities.

Computational Cost and Threshold Sensitivity. On an RTX 4090, InsightFace embedding and clustering add 8.5 seconds per 30-second clip, corresponding to a 39.2% overhead relative to CAM's 21.7-second construction time. On CPU, the same association stage takes more than one hour for a 30-minute video. Its similarity threshold also controls the split–merge trade-off. Because M3-Agent's provided setting is validated only on its native benchmark, its transferability to other visual conditions is unclear. In contrast, CAM generates appearance descriptors during the existing MLLM extraction step, requiring no additional GPU model, inference pass, or similarity-threshold tuning.

B.2 Evidence-Driven Abstraction

Semantic graphs from long videos differ sharply from text-derived graphs. In video, a few persistent entities—especially characters—recur across many clips, producing high-degree nodes, dense temporal links, and strict ordering constraints. Unlike text, where facts are discrete, video evidence is incremental, noisy, and unevenly distributed. This uneven accumulation means abstraction cannot simply follow fixed time intervals. Some entities stabilize early, while others remain sparse until much later. Premature abstraction risks inconsistency, while delayed abstraction wastes evidence already sufficient for reasoning. Both mismatches undermine memory coherence and downstream analysis.

Consequently, fixed-interval abstraction misaligns with evidential support, yielding premature or delayed summaries that undermine memory consistency and downstream reasoning. To address this imbalance, CAM employs *evidence-driven abstraction*, where abstractions are triggered by the accumulation of clip-level relations in the memory graph rather than periodic time steps.

Concretely, CAM applies two complementary abstractions: *character-level* abstraction operates on individual nodes, triggering summaries only after sufficient incident evidence accumulates to support stable inference of long-term behaviors, whereas *relationship-level* abstraction operates on node pairs, triggering only after sufficient interaction evidence accrues to capture persistent social dynamics beyond isolated characters. Both abstractions are conditioned on graph structure and relation density via predefined evidence thresholds, ensuring summaries are grounded in adequate observations.

Character-Level Abstraction. For each character node $v \in V_{\text{char}}$, we maintain the set of incident edges that aggregate all episodic evidence involving that character, including actions, interactions with other entities, and associated contextual states. At processing step t, this set is

$$\mathcal{E}_t(v) = \{e \in E_t \mid e.\text{source} = v \text{ or } e.\text{target} = v\}.$$

This set grows incrementally as new clips are processed. To support repeated refinement without retriggering after every subsequent edge, we record $b_{\text{char}}(v)$, the number of incident edges present immediately after the most recent character-level abstraction, initialized to zero. Character-level abstraction is triggered only after at least τ_{char} new incident edges have accumulated:

$$\text{Trigger}^{(t)}_{\text{char}}(v) = \mathbb{I}\big(|\mathcal{E}_t(v)| - b_{\text{char}}(v) \geq \tau_{\text{char}}\big),$$

where $\mathbb{I}(\cdot)$ is the indicator function.

This thresholding reflects the intuition that stable character attributes can only be inferred from *sufficiently diverse observations*. When the trigger fires, the newly accumulated evidence, including its temporal and contextual annotations, is summarized together with the existing high-level characterization, if one is available. The LLM output updates the character-level abstract node or attribute edge linked to v, and the counter is advanced to $b_{\text{char}}(v) \leftarrow |\mathcal{E}_t(v)|$. All low-level episodic edges are retained for evidence preservation and future refinement.

Relationship-Level Abstraction. Beyond individual traits, long-term video understanding requires modeling persistent inter-character relations (e.g., cooperation or rivalry) that emerge only through repeated interactions over time rather than within isolated clips.

For a character pair $(u, v) \in V_{\text{char}} \times V_{\text{char}}$, we define the relationship evidence available at processing step t as

$$\mathcal{E}_t(u, v) = \mathcal{E}_{t,\text{direct}}(u, v) \cup \mathcal{E}_{t,\text{indirect}}(u, v),$$

where $\mathcal{E}_{t,\text{direct}}(u,v)$ contains edges corresponding to explicit interactions between u and v, and $\mathcal{E}_{t,\text{indirect}}(u,v)$ captures co-involvement with the same object, location, or third entity within a clip or short temporal window, encoding implicit relational signals. We similarly maintain $b_{\text{rel}}(u,v)$, the relationship-evidence count recorded after the most recent abstraction, initialized to zero. Relationship abstraction is triggered after τ_{rel} new interaction edges accumulate:

$$\text{Trigger}_{\text{rel}}^{(t)}(u,v) = \mathbb{I}\big(|\mathcal{E}_t(u,v)| - b_{\text{rel}}(u,v) \geq \tau_{\text{rel}}\big).$$

When the threshold τ_{rel} is reached, the newly accumulated interaction evidence is summarized together with the existing relational abstraction, if one is available. The resulting description updates the abstract relational edge between u and v, and the counter is advanced to $b_{\text{rel}}(u,v) \leftarrow |\mathcal{E}_t(u,v)|$.

As with character-level abstraction, low-level relational edges are preserved to allow subsequent refinement as new interactions are observed. By conditioning each update on a fresh batch of accumulated relational evidence, CAM ensures that relationship summaries reflect consistent behavioral patterns rather than transient or coincidental interactions.

B.3 Agent Toolbox

To empower the agent to actively navigate the heterogeneous memory graph and ground its reasoning in raw multimodal signals, we define a comprehensive suite of seven specialized tools. Each tool is designed to overcome specific bottlenecks in long-form video understanding, ranging from structured knowledge retrieval to raw sensory verification.

- **General Search (`general_search`):** This tool serves as the agent's foundational retrieval mechanism. To address queries that require both global entity relations and fine-grained visual or conversational details, this search operation is mathematically decomposed into three distinct stages:

  1) Weighted Query Decomposition. During tool invocation, the Planner Node directly parses the natural-language query q into a semantic triple $\mathcal{T}_q = \langle s_q, r_q, t_q \rangle$. To capture the relative importance of each component, the Planner dynamically assigns a normalized weight vector $\mathbf{W}_q = \langle w_s, w_r, w_t \rangle$ (where $w_s + w_r + w_t = 1$). Simultaneously, it extracts auxiliary context I_q and determines a retrieval budget $\mathbf{K}_q = \langle k_{\text{high}}, k_{\text{low}}, k_{\text{conv}} \rangle$ to allocate how many nodes to retrieve across the three memory channels. These parameters collectively form the tool's execution arguments:

  $$\text{args}_t = \big(\mathcal{T}_q,\ \mathbf{W}_q,\ \mathbf{K}_q,\ I_q\big)$$

  2) Triple Similarity Computation. To evaluate relevance within the heterogeneous graph $\mathcal{G}$, we define a similarity function in a shared latent space mapped by a dense embedding function f_{emb}. The base cosine similarity between elements is $sim(x,y) = \frac{f_{\text{emb}}(x)^\top f_{\text{emb}}(y)}{\|f_{\text{emb}}(x)\|_2 \|f_{\text{emb}}(y)\|_2}$. When comparing the query triple $\mathcal{T}_q = \langle s_q, r_q, t_q \rangle$ to a candidate triple $\mathcal{T}_c = \langle s_c, r_c, t_c \rangle \in \mathcal{G}$, we first account for reversed semantic roles using a weighted directional entity alignment score for the nodes:

  $$\text{Align}\big((s_q,t_q),(x,y)\big) = w_s \cdot sim(s_q,x) + w_t \cdot sim(t_q,y)$$

  The total structural similarity, S_{struct}, integrates this bidirectional node alignment with the direct semantic matching of the relation, ensuring overall robustness against role reversals:

  $$S_{\text{struct}}(\mathcal{T}_q,\mathcal{T}_c) = \max\Big(\text{Align}\big((s_q,t_q),(s_c,t_c)\big),\ \text{Align}\big((s_q,t_q),(t_c,s_c)\big)\Big) + w_r \cdot sim(r_q,r_c)$$

  3) Multi-Channel Retrieval. Retrieval is executed independently across partitioned memory channels to reflect their distinct semantic roles. High-level abstractions are retrieved primarily via structural similarity to capture long-term, aggregated knowledge. Low-level episodic relations require fine-grained spatio-temporal grounding, thus incorporating the extracted context I_q alongside structural similarity ($S_{\text{struct}} + \lambda \cdot sim(I_q, K_c)$), where K_c denotes the contextual metadata attached to the candidate relation and λ weights the context similarity. Conversely, conversational content bypasses structured triples entirely, relying on direct natural-language matching against the original query ($\arg\max_i sim(q,d_i)$). To preserve discursive coherence, retrieved utterances are expanded to include their immediate chronological neighbors (d_{i-1}, d_{i+1}). Finally, all retrieved evidence is merged into a unified set $\mathcal{R} = \mathcal{R}_{\text{high}} \cup \mathcal{R}_{\text{low}} \cup \mathcal{R}_{\text{conv}}$, enabling the system to flexibly integrate abstract, episodic, and dialogue contexts for downstream reasoning.

- **Temporal Context Search (`search_temporal_context`):** For queries requiring strict temporal localization, this tool extends the `general_search` operation by introducing two bounding parameters: t_{begin} and t_{end}. By constraining the search space to a specific temporal window, the agent effectively prunes irrelevant historical or future events, focusing the similarity computation exclusively on a localized subgraph.
- **Search Object (`search_object`):** Visual perception models frequently introduce vocabulary discrepancies during initial graph construction (e.g., misclassifying a "marker" as a "pen"). To mitigate this semantic mismatch, this tool bypasses rigid lexical matching in favor of robust, embedding-based entity alignment, ensuring accurate object identification despite variations in natural language taxonomy.
- **Frequency Statistics (`get_frequency_stats`):** A quantitative tool that aggregates occurrences across the timeline, returning the exact number of times a specific action, object, or event appears in the structured memory.
- **Graph Traversal (`graph_traversal`):** Enables explicit topological traversal of the memory structure for multi-hop deduction. Let $u \in V$ denote an anchor entity currently held in the agent's intermediate memory $\mathcal{M}_t$. Given a target relation query r_q, this tool retrieves the semantically filtered neighborhood: $\mathcal{N}(u, r_q) = \{v \in V \mid \exists e \in E \text{ s.t. } e.\text{source} = u,\ e.\text{target} = v,\ sim(r_q, e.\text{relation}) \geq \tau\}$, where τ is a similarity threshold. By sequentially applying this operation, the agent actively constructs a deductive path $\mathcal{P} = (v_0, r_1, v_1, \ldots, r_L, v_L)$ across heterogeneous edges (e.g., character relationships or causal links), where L is the number of hops. This bridges disparate temporal video segments strictly through graph topology rather than global search.
- **Video Rewatch (`watch_video_clip`):** The initial partitioning of video into textual triples via MLLMs inevitably compresses subtle visual nuances, such as background context and continuous motion dynamics. Serving as a critical multimodal fallback mechanism, this tool accepts a specific clip identifier as input, enabling the agent to bypass the structural memory and directly retrieve the raw visual frames for re-evaluation. This ensures the reasoning process remains firmly grounded in empirical visual evidence when the generated graph lacks the fine-grained details necessary to resolve semantic ambiguities.
- **Audio Listening (`listen_to_audio`):** Similarly, non-speech acoustic cues (e.g., environmental sound effects or background noise) are frequently omitted during graph construction, as transcription models predominantly focus on character dialogue. This tool allows the agent to bypass the text modality and directly sample the raw audio track within targeted clips, providing essential auditory grounding when the transcribed contextual information is insufficient.

C Additional Related Work

Long Video Understanding. Long video understanding remains challenging due to limited context windows, evolving semantics, and the need to preserve both fine-grained details and long-range temporal dependencies [15], motivating memory-centric designs that compress or store video information for long-horizon reasoning, such as compact textual summaries, fixed-size streaming memories, or hierarchical memory across abstraction levels with bidirectional or event-level updates [9, 16–23]. While effective, these methods often rely on fixed memory budgets or heuristic updates, leading to information loss when critical details appear late in the video, whereas retrieval-augmented approaches dynamically fetch relevant segments or auxiliary signals at query time to improve scalability [10, 24, 25], but remain highly dependent on retrieval accuracy and typically treat evidence as unstructured context, limiting temporal and relational modeling. More recently, agentic and multi-agent frameworks decompose long-video reasoning into iterative or collaborative subtasks, with VideoAgent [26], VideoMultiAgents [27], and LongVideoAgent [28] enabling targeted or parallel analysis through tool use and shared memory, yet overall existing methods still trade off between memory compression, retrieval efficiency, and reasoning capacity, motivating unified frameworks that integrate memory, retrieval, and structured reasoning for robust long-horizon understanding.

Graph-based Video Search. Graph structures have been widely explored in text-based retrieval-augmented generation and memory systems to model relational knowledge and support multi-hop reasoning [29–33]. In the video domain, structured representations have increasingly been adopted to address the limitations of fixed context windows and sequential processing, with early spatio-temporal graphs explicitly modeling object interactions and temporal dynamics for activity

and event understanding [12, 27]. More recent approaches leverage graph-based memories and prompting to bridge high-level semantics and low-level video content, such as Glance-Focus [34], while explicit entity tracking further supports long-form reasoning through evolving relational graphs, as in GraphVideoAgent, which achieves strong performance on long-horizon benchmarks including EgoSchema and NExT-QA [12]. Structured retrieval-augmented frameworks combine graphs with selective retrieval to scale to extreme video lengths, including VideoRAG's graph-grounded semantic indexing [13], RAVU's compositional reasoning over spatio-temporal entity graphs [14], and Vgent's graph-based retrieval–reasoning–augmented generation with intermediate verification [35]. Despite these advances, most methods rely on relatively flat relational structures, treat multimodal signals as external augmentations rather than unified graph components, and lack explicit modeling of hierarchical event abstractions or causal dependencies across clips, motivating richer and more integrated graph representations for robust long-video understanding.

Memory-Based Video LLMs. To support long-video reasoning, memory-based methods construct structured knowledge over video streams. EgoLife [5] emphasizes lifelong egocentric understanding with continuous memory accumulation, but its memory mainly serves retrieval and summarization rather than explicit entity–relation reasoning; Ego-R1 [36] extends this direction with vision-centric tools and iterative reasoning for long-horizon tasks, yet still lacks a structured relational representation. HippoMM [11] adopts a biologically inspired episodic–semantic memory to improve long-term retention, but its largely sequential organization limits multi-hop and relational reasoning, while M3-Agent [8] stores clip-level textual summaries in an external memory, leading to unstructured representations and premature abstraction that are difficult to revise as new evidence emerges.

In contrast, CAM models long-video memory as a heterogeneous, temporally grounded semantic graph, deferring high-level abstraction until sufficient evidence accumulates and separating episodic observations from semantic summaries. Its multi-channel retrieval and query-guided video rewatching further enable efficient access to both abstract knowledge and raw visual evidence, supporting robust reasoning when critical information is sparsely distributed over time.

D Additional Experiments and Analysis

D.1 Abstraction Threshold Sensitivity

Unlike temporal boundaries, which force abstraction regardless of content density, the thresholds τ_{char} and τ_{rel} are grounded in information density. In our implementation, these thresholds are empirically set to $\tau_{\text{char}} = 50$ and $\tau_{\text{rel}} = 30$. A threshold that is too low causes premature abstraction, whereas a threshold that is too high delays the emergence of high-level semantics and risks exceeding the model's context window during summarization. Table 3 in the main text reports overall and category-level accuracy under each setting.

D.2 Reasoning Efficiency

Method	M3-Bench-robot			M3-Bench-web			HippoVlog		
	Avg rounds	correct	wrong	Avg rounds	correct	wrong	Avg rounds	correct	wrong
M3-Agent	3.14	2.55	3.58	3.11	2.68	3.52	2.76	2.46	3.33
WorldMM	2.74	1.85	3.22	2.21	2.08	2.41	1.28	1.2	1.49
CAM	2.2	1.78	2.92	2.48	2.36	2.71	2.13	2.03	2.41

Table 5: Average reasoning rounds required by different methods across benchmarks, separated by correct and incorrect predictions.

Table 5 presents a detailed breakdown of the average reasoning rounds executed by the agents, separated into correct and incorrect predictions across the three benchmarks. A consistent trend across all methods is that incorrect predictions require significantly more reasoning rounds than correct ones, illustrating the agents' tendency to exhaustively search their memory when critical evidence is missing or ambiguous. M3-Agent consistently exhibits the highest average reasoning rounds across all datasets (e.g., 3.14 on M3-Bench-robot) and struggles significantly with termination, as evidenced by its disproportionately high round counts for incorrect predictions. This aligns with

its unstructured memory format, which forces the agent into repetitive, inefficient search loops when it cannot locate necessary details. In contrast, CAM demonstrates a highly efficient reasoning trajectory. By leveraging its structured, temporally grounded graph and explicit tool navigation, CAM consistently arrives at correct answers using fewer rounds than M3-Agent (e.g., requiring only 1.78 rounds versus 2.55 on M3-Bench-robot) while also avoiding endlessly repetitive loops when the required information is ultimately unavailable. Meanwhile, WorldMM shows varying round counts—dropping notably low on HippoVlog (1.28 average)—reflecting its heavy reliance on disjointed, single-pass memory bank lookups rather than sustained, multi-hop deduction.

D.3 Tool-Use Statistics

Benchmark	1st Most Common (%)	2nd Most Common (%)	3rd Most Common (%)
M3-Bench-robot	`general_search` (70.2%)	`watch_video_clip` (16.8%)	`search_temporal_context` (8.6%)
M3-Bench-web	`general_search` (78.6%)	`search_temporal_context` (13.9%)	`watch_video_clip` (3.7%)
HippoVlog	`general_search` (73.8%)	`watch_video_clip` (13.3%)	`listen_to_audio` (7.4%)

Table 6: Distribution of the top three most frequently invoked tools by the agent during the reasoning process across different benchmarks.

Table 6 details the distribution of tool calls during the agent's reasoning process across the evaluated benchmarks. The variation in tool invocation frequency directly reflects the differing question distributions and specific reasoning demands inherent to each dataset. Across all benchmarks, `general_search` consistently ranks as the most frequently invoked tool, accounting for over 70% of total calls. This dominance is expected, as the agent is strictly constrained to utilize `general_search` during its initial reasoning round to establish a foundational knowledge context. Furthermore, `watch_video_clip` emerges as a critical secondary mechanism, ranking second for both M3-Bench-robot and HippoVlog. The prominence of this multimodal fallback tool highlights its vital role in recovering fine-grained visual details, effectively mitigating the inevitable information loss that occurs when raw video segments are initially compressed into textual representations. Finally, dataset-specific requirements dictate the remaining tool preferences, such as the increased reliance on `search_temporal_context` for M3-Bench-web and the targeted use of `listen_to_audio` for the audio-rich HippoVlog benchmark, demonstrating the agent's adaptive selection capabilities.

D.4 Robustness to Graph Noise

We examine how errors in low-level graph evidence affect downstream question answering. For each video, we inject edges sampled from other video graphs at ratios of 2%, 5%, and 10% of the original low-level edges to simulate errors and hallucinations in clip-level extraction. We then rerun abstraction and question answering. Table 7 reports the results on M3-Bench-robot and M3-Bench-web.

Table 7: Robustness to graph noise. We inject low-level edges sampled from other graphs before abstraction and question answering, and report accuracy (%) on M3-Bench-robot and M3-Bench-web.

Injected noise ratio	M3-Bench-robot						M3-Bench-web					
	ME	MH	CM	PU	GK	**All**	ME	MH	CM	PU	GK	**All**
0% (clean)	**65.5**	**81.2**	**66.8**	**83.8**	**36.7**	**64.5**	**64.6**	54.7	**52.1**	**75.6**	75.5	**69.3**
2%	62.4	77.6	62.6	83.3	33.3	62.5	64.4	**55.1**	49.3	74.7	75.5	68.5
5%	62.6	80.0	62.4	82.4	34.9	62.5	63.9	54.3	47.7	72.8	**77.5**	67.9
10%	60.6	70.6	61.8	82.0	35.4	60.8	61.9	50.7	44.2	68.9	71.6	65.2

Accuracy decreases gradually as the noise ratio increases. At 10% noise, accuracy drops by 3.7 percentage points on M3-Bench-robot and 4.1 points on M3-Bench-web. At 10% noise on M3-Bench-robot, the largest category-level decline occurs in multi-hop reasoning, from 81.2% to 70.6% (10.6 percentage points), compared with a 4.9-point decline in multi-evidence reasoning. This suggests that multi-hop reasoning is particularly sensitive to mismatched graph evidence under this noise condition.

These results indicate that CAM tolerates a moderate amount of randomly injected noise, although this experiment does not capture systematic or correlated extraction errors.

D.5 Streaming and Reasoning Efficiency

Streaming Construction. Conditioning the extraction of clip c_i on the descriptors A_{i-1} from the preceding clip introduces a sequential dependency. However, this design fundamentally aligns with the constraints of our target domain: continuous, entity-centric video streams (e.g., live robotic operations, egocentric wearable cameras). In these online environments, future frames are fundamentally unavailable, rendering offline batch processing impossible. Therefore, the architectural priority is not parallelization, but rather real-time execution speed. Our empirical measurements demonstrate that processing a 30-second clip requires only 21.7 seconds on average (combined MLLM and LLM inference). Because the processing time is strictly less than the clip duration, CAM achieves faster-than-real-time performance, allowing it to scale continuously on a live video stream without inducing a temporal bottleneck.

Reasoning Latency. Table 8 compares average reasoning latency per question with accuracy on M3-Bench-robot.

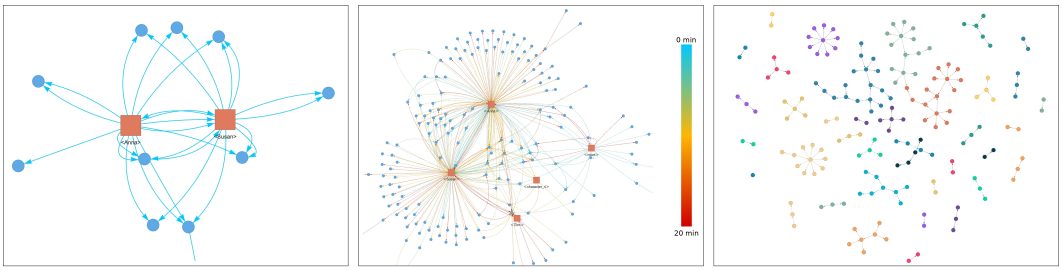
Table 8: Average reasoning latency per question and accuracy on M3-Bench-robot.

Method	**Latency per question**	**Accuracy (%)**
M3-Agent	81.4 s	41.4
WorldMM	66.7 s	34.5
CAM (top-$k = 30$)	40.1 s	62.9
CAM-full	47.3 s	**64.5**

CAM-full is 34.1 seconds faster per question than M3-Agent and 19.4 seconds faster than WorldMM, while improving accuracy by 23.1 and 30.0 percentage points, respectively. Reducing the retrieval budget to top-$k = 30$ lowers latency from 47.3 to 40.1 seconds and reasoning token consumption from 30,711 to 24,002 tokens per question, reductions of 15.2% and 21.8%, respectively. Accuracy decreases from 64.5% to 62.9%, a drop of 1.6 percentage points, demonstrating an adjustable accuracy–efficiency trade-off.

D.6 Graph Visualization

Long-video graphs and text-derived graphs exhibit fundamentally different topologies. Text-derived graphs typically contain many distinct nodes connected by relatively few edges, allowing densely connected communities to form meaningful clusters. In contrast, long-video graphs repeatedly accumulate observations around a small set of persistent entities, resulting in fewer nodes but substantially more temporally grounded edges. Their node-degree distribution is also highly imbalanced: principal characters accumulate far more incident evidence than secondary characters. Consequently, long-video graphs do not decompose cleanly into clusters. As illustrated in Figure 5, these properties make cluster-based abstraction poorly aligned with long-video memory. This motivates CAM's *evidence-driven abstraction*, which triggers character- and relationship-level summaries according to the evidence accumulated by individual entities and entity pairs rather than relying on global graph clusters.



(a) 2-min video (b) 20-min video (c) GraphRAG for a document

Figure 5: Qualitative comparison of graph topologies. (a) CAM memory graph for a 2-minute video. (b) CAM memory graph for a 20-minute video, demonstrating scalable and organized relational density over extended temporal horizons. (c) A standard text-based GraphRAG topology for a written document, highlighting the structural differences between general text abstraction and CAM's entity-centric video memory.

E Prompt Templates

In this section, we provide the specific prompts used to instruct the LLM planner and extract abstract information within the CAM framework.

E.1 Memorization Prompts

Prompt: Clip-level Observation

You are given a 30-second video represented as sequential frames (pictures in chronological order).
Your tasks:

1. **Characters' Behavior**: Describe each character's behavior in chronological order.
   - Include: (a) Interaction with objects in the scene. (b) Interaction with other characters. (c) Actions and movements.
   - Do NOT repeat the information from the conversation.
   - When the character interacts with objects in the scene, include precise location information for placement, retrieval, or movement.
   - Furniture/container names include "dressing table", "bedside table", "wardrobe", "shoe cabinet", "refrigerator", and "microwave oven".
   - Spatial modifiers include "below", "above", "left side", "right side", "beside", "in front of", "next to", "under", and "on the counter".
   - Each entry must describe exactly one event/detail. Split sentences if needed.

   Example output: `["<Alice> enters the room.", "<Alice> takes cap from the cabinet on the left side of the wardrobe.", "<Alice> sits with <Bob> side by side on the couch.", "<Bob> watches TV."]`

2. **Conversation**: Record the dialogue based on subtitles. Always use the characters' real name that appears in the subtitle.
   - *Output format:* List of two-element lists `[character, content]`.
   - *Example output:* `[["<Alice>", "Hello, my name is Alice."], ["<Bob>", "Hi, I'm Bob. Nice to meet you."]]`
3. **Characters' Appearance**: Describe or update each character's appearance, including facial features, clothing, body shape, hairstyle, or other distinctive characteristics.
   - Each characteristic should be concise, separated by commas.
   - Existing characters: Update if changes observed (hair, clothing), enhance if new details visible, otherwise keep unchanged.
   - Keep appearance info even if character leaves scene. If a character is not visible in scene, do not include them in the characters' appearance list.

   Example output: `[Appearance(name="<Alice>", appearance="female, fat, ponytail, wear glasses, short-sleeved shirt, blue jeans, white sneakers")]`
4. **Scene**: Use one word or phrase to describe the scene in the current video (e.g., "bedroom", "gym", "office"). This field is optional and should only be provided if the scene is clearly identifiable.

Character Naming Rules:
Use angle brackets to represent characters (e.g., <Alice>, <Bob>, <male_1>) in behaviors, conversation, and character appearance. There are three cases to name a character:

1. The name can be deduced from the conversation or subtitles. Refer to the character by his/her name (e.g., <Alice>, <Bob>).
2. The name is not provided, but you can deduce the character's job or identity (e.g., police, mailsman). Refer to the character by `<job_number>` (e.g., <police_1>). Only refer to a character by their job when it is clear. Do not make any unsure guesses on a character's job.
3. Neither the name nor the job can be inferred. Name the character by gender and a number (e.g., <male_1>, <male_2>, <female_1>).

For the second and third cases, if the character's name is provided in the following video clips, use the equivalence line to update the character's name.

Character Matching Rules:
For characters appearing in the video (MUST follow before creating new characters):

- First, check if the subtitle name is provided. If so, use it as the character name (e.g., <Anna>). Also compare the character name with the unknown characters (e.g., <male_1>, <police_1>) in the appearance list. If high similarity is found, add "Equivalence: <unknown_id>, <Anna>" at the start of behaviors. This indicates that the character is the same as the previously identified unknown character. If the equivalence is found, refer to this character by its character name instead of the unknown ID in the behaviors, conversation, and character appearance.
- If a character's name is not provided in the subtitle, match this character's appearance to the characters in the appearance list. If high similarity is found, refer to this character by its name in the appearance list.
- If no match is found, create a new character following the naming rules above. Our goal is to minimize the number of unique character IDs. When uncertain, match to existing rather than creating new.

Additional Rules:

- Maintain strict chronological order.
- Avoid repetition in both behavior and conversation. If no behavior or conversation is observed, return an empty list for behaviors and conversation.

Prompt: Triples Extraction

Convert each action sentence into triples of the form `[source, content, target]`.
Return **only** a valid JSON array. Do not include explanations, Markdown, or any extra text.
Output Format:

- Use the format `[[source, content, target], ...]` with double quotes and no trailing commas.
- Use `null` when no target exists.
- Preserve the original sentence order and the action order within each sentence.

Definitions:

- **Source**: The entity performing the action or having the state.
- **Content**: A verb-centered action, relation, or state.
- **Target**: The affected or related entity.

First identify actors, actions, and affected entities; then resolve references, split compound structures, normalize verbs, add state relations, and remove redundancy.
Rules:

1. **Entities**: Keep angle brackets around characters (e.g., `<Alice>`, `<robot>`) but never around objects. If an object has angle brackets in the input, remove them. Copy entity names verbatim and do not invent entities.
2. **Verbs**: Use simple present tense rather than progressive forms. Retain relevant prepositions, directions, and adverbs (e.g., "looks at," "turns left," or "runs quickly").
3. **Objects**: Use singular nouns, retain adjectives and named-object descriptions, and split compound objects into separate triples.
4. **Multiple Relations**: Create a separate triple for each subject, action, and object while preserving their order. For example, "`<Lisa>` dances and sings" produces two triples.
5. **Reference Resolution**: Never retain pronouns such as "his," "her," or "their." Replace them with explicit entities or owners; if ambiguous, use the nearest subject.
6. **Body Parts**: Merge a body part into the action instead of treating it as an object. For example, "`<Alice>` hits `<Bob>`'s head" becomes `["<Alice>", "hits head", "<Bob>"]`.
7. **Communication**: Encode communication directly between characters. Do not create abstract targets such as "question" or "message"; for example, use `["<Tom>", "asks", "<Mary>"]`.
8. **Resulting States**: Add a state triple when an action changes an object's state or location. For example, "puts coffee on table" yields `["<robot>", "puts", "coffee"]` and `["coffee", "is on", "table"]`.
9. **Spatial Relations**: Keep location information in the content field and use clean entities as the source and target. For example, use `["<Betty>", "sits on the right side of", "sofa"]`.
10. **Consistency**: Preserve meaningful adjectives, adverbs, and other modifiers. Keep only distinct actions and do not duplicate a state already expressed by a stronger relation.
11. **Fallback**: If unsure, use the minimal form `[source, verb, target]`.

Example Input:

```
["<Michael> pats <Susan>'s shoulder and smiles.",
"<robot> places the red cup on the counter.",
"<John> takes his wallet and keys from the drawer.",
"<Betty> carries a red bag in her left hand and a white bag in her right
hand."]
```

Example Output:

```
[["<Michael>", "pats shoulder", "<Susan>"],
["<Michael>", "smiles", null],
["<robot>", "places", "red cup"],
```

```
["red cup", "is on", "counter"],
["<John>", "takes", "John's wallet"],
["<John>", "takes", "John's key"],
["John's wallet", "is in", "drawer"],
["John's key", "is in", "drawer"],
["<Betty>", "carries in left hand", "red bag"],
["<Betty>", "carries in right hand", "white bag"]]
```
Now convert the following list of action sentences into triples:

Prompt: Character-level Abstraction

You are given a character's name and a list of their behaviors in chronological order.
Your task is to summarize the character's attributes:

- Personality (e.g., confident or nervous).
- Role or profession (e.g., host or newcomer).
- Interests or background, when inferable.
- Distinctive behaviors or traits (e.g., speaks formally or fidgets).

Avoid restating visual facts; focus on identity construction.
For each attribute, you should also provide a confidence score between 0 and 100. If the confidence score is less than 50, you should not include the attribute in the output.
Output a JSON dictionary in which each key is an attribute and each value is its confidence score.
Example output: `{"student": 90, "enthusiastic": 80, "likes to read": 70, "professional": 50, "likes to play games": 60}`

Prompt: Relationship-level Abstraction

You are given a list of character interactions in chronological order. Your task is to extract relationships between the characters, including:

- Roles (e.g., friends, colleagues, host–guest, teacher–student, or parent–child).
- Attitudes or emotions (e.g., respect, dislike, or friendliness).
- Power dynamics (e.g., who leads or whether the characters are equal).
- Evidence of cooperation.
- Exclusion, conflict, or competition.

Additional Rules:

- Store only abstract relationships between the characters.
- Do not include actions or summaries of actions in the output (e.g., `<Alice>` speaks with `<Bob>` or `<Alice>` plays games with `<Bob>`).
- Do not generate repetitive or symmetric information.

For each relationship, you should also provide a confidence score between 0 and 100. If the confidence score is less than 50, you should not include the relationship in the output. It is acceptable to only generate a few relationships if you don't have enough information.
Output a JSON array (a list of lists). Each inner list contains four elements: `[character1, relationship, character2, confidence score]`.
Example output:
```
[["<Alice>", "is friend with", "<Bob>", 90],
["<Alice>", "is teacher of", "<Charlie>", 80],
["<Charlie>", "respects", "<Alice>", 70]]
```

Prompt: Conversation Summary

You are given a conversation between several characters.
Output Format: Return a JSON object with the following keys:

1. `"summary"`: A string summarizing the key topics, decisions, or outcomes in two to four concise sentences.
2. `"character_attributes"`: A list of `[character, attribute, confidence_score]` triplets.
3. `"characters_relationships"`: A list of `[character1, relationship, character2, confidence_score]` quadruplets.

Example output:

```
{"summary": "Alice and Bob discussed their upcoming project.
They agreed on a timeline and assigned tasks.",
"character_attributes": [["<Alice>", "organized", 85],
["<Bob>", "cautious", 70]],
"characters_relationships":
[["<Alice>", "is friend with", "<Bob>", 90]]}
```

Detailed Instructions:

1. **Summary**
   - Summarize the key topics, decisions, or outcomes discussed in the conversation.
   - Write two to four concise sentences covering the main themes and important points.
   - Focus on what was discussed and decided, rather than on individual statements.

   Example: "Alice and Bob discussed their upcoming project. They agreed on a timeline and assigned tasks. Bob expressed concerns about the deadline, which Alice addressed by suggesting additional resources."
2. **Character Attributes**
   - Extract each character's attributes revealed through their dialogue and interaction style.
   - Focus on personality traits, role or profession, interests, and background information when mentioned.
   - Do not include physical appearance, concrete actions, temporary emotional states, or information not directly supported by the conversation.
   - Confidence scores range from 0 to 100. Include only attributes with confidence scores of at least 50.
   - Avoid redundant or overly similar attributes (e.g., do not include both "friendly" and "kind" unless they are distinctly different).
   - Use angle brackets for character names (e.g., `<Alice>` and `<Bob>`).
   - Output a list of three-element lists: `[character, attribute, confidence_score]`.

   Example output:
   `[["<Alice>", "organized", 85], ["<Alice>", "problem-solver", 75], ["<Bob>", "detail-oriented", 80], ["<Bob>", "cautious", 70]]`
   Invalid examples: `["<Alice>", "asked a question", 90]` describes an action, and `["<Bob>", "has brown hair", 80]` describes appearance.
3. **Character Relationships**
   - Extract abstract relationships between characters based on their dialogue interactions.
   - Include roles, attitudes, power dynamics, and evidence of cooperation, conflict, exclusion, or competition.
   - Do not include specific actions or events, temporary interactions, dialogue content, or discussion topics.
   - Confidence scores range from 0 to 100. Include only relationships with confidence scores of at least 50.
   - Do not generate symmetric duplicates. If `<Alice>` respects `<Bob>`, do not include the reverse unless it is explicitly different.

- It is acceptable to generate only a few relationships if there is insufficient information.
- Output a list of four-element lists: `[character1, relationship, character2, confidence_score]`.

Example output:
```
[["<Alice>", "is friend with", "<Bob>", 90],
["<Alice>", "is teacher of", "<Charlie>", 80],
["<Charlie>", "respects", "<Alice>", 70]]
```
Invalid examples: `["<Alice>", "spoke with", "<Bob>", 90]` is an action, and `["<Alice>", "discussed the project", "<Bob>", 85]` is dialogue content.

Edge Cases:

- If only one character speaks, focus on that character's attributes and skip relationships.
- If the conversation is empty or unclear, return empty arrays for attributes and relationships and provide a brief summary noting the issue.
- If character names are ambiguous, use the names provided in the conversation.

Now summarize the following conversation in JSON format:

E.2 Reasoning Prompts

Prompt: Planner Strategy

Tool usage guidance:

- `general_search`: Use this FIRST for all questions to get a temporal anchor and find relevant `clip_ids`. Allocate your budget (Total $k \leq 50$) based on the primary modality of the question:
  - **k_action (0-30)**: Primary for behavior, actions, temporal sequence, or 'where is' queries.
  - **k_conversation (0-30)**: Primary for 'why', dialogue, or causal reasoning.
  - **k_high_level (0-10)**: Secondary for character traits or relationships.
  - **k_appearance (0-15)**: Use ONLY for physical looks, hair, or clothing. Set to 0 if the question is irrelevant to appearance.

  IMPORTANT (First Round): If this is your first tool call, you MUST use `general_search` and allocate the FULL budget (Total $k = 50$) across these modalities to ensure a broad understanding of the video.
- `search_temporal_context`: Use this ONLY after finding a candidate `clip_id` via `general_search` to see events right before or after it.
- `watch_video_clip`: Use this ONLY after finding a candidate `clip_id` via `general_search`. Mandatory for visual questions requiring high detail (e.g., specific placement of objects, visual state, exact counts) when text graph is insufficient. Provide a specific `focus` based on what is missing from the text.

Strategy for Clip ID Selection:

1. Look for the `[clip_id]` next to the most relevant action or conversation in `general_search` results.
2. Pay attention to temporal context: if looking for what happened *before* an event, search the clip(s) preceding the event.
3. Use that specific `clip_id` as input for `watch_video_clip` or `search_temporal_context`.

Analyze the conversation history above. What is the most effective next step to solve the question?

Prompt: Query Parser

You are a query parser for a knowledge graph system that stores video information in a hierarchical structure.
Given a query, output the following in JSON format:
Output Fields:

1. `query_triples`: A list containing one to three query triples.
   - Each triple has the format `[source, content, target, source_weight, content_weight, target_weight]`.
   - Use `null` for missing components.
   - Normalize unnamed characters to generic IDs (e.g., `<male_1>`); retain a name when it is explicitly provided.
   - Assign each weight a value between 0.0 and 1.0.
2. `speaker_strict`: Set this field to the relevant character ID or IDs when the query asks about specific dialogue; otherwise, use `null`.
3. `spatial_constraint`: Use a location string only when the query refers to a general space; otherwise, use `null`.

Special Rules for Location Queries:

- **Hierarchical locations**: Preserve complete location phrases (e.g., "cabinet below the dressing table") without splitting them.
- **Temporal-spatial queries**: Construct triples that retrieve the most recent state or source location.

Now parse the following query in JSON format:

Prompt: Final Answer

You are the final-answer synthesizer. Use all collected evidence in the conversation history to answer the original question concisely and directly.
Requirements:

- Respond in exactly **one sentence**.
- Do not include explanations, meta-commentary, or justifications.
- Do not answer "I don't know," "The information is not sufficient," or "It is unclear."
- If the evidence is uncertain, make the most reasonable inference supported by the conversation history.
- Reuse exact terms from the question and search results when possible.

E.3 Answer Verification

Prompt: Answer Verifier (M3-Bench Only)

You are provided with a question, a ground truth answer, and an answer from an agent model. Your task is to determine whether the ground truth answer can be logically inferred from the agent's answer, in the context of the question.
Do not compare only the surface forms of the two answers. Instead, determine whether the meaning of the agent answer supports or implies the ground truth answer in the context of the question.
Decision Rule:

- Return `"Yes"` if the ground truth answer can be reasonably inferred from the agent answer.
- Return `"No"` otherwise.

Important Notes:

- Exact wording or matching sentence structure is not required.

- Semantic entailment or implication is sufficient.
- Return only `"Yes"` or `"No"`, with no explanation or additional formatting.

Input Fields:

- `question`: The question asked.
- `ground_truth_answer`: The correct answer.
- `agent_answer`: The model answer to be evaluated.

Now evaluate the following input:
Input:
`question: {question}`
`ground_truth_answer: {ground_truth_answer}`
`agent_answer: {agent_answer}`
Output: `"Yes"` or `"No"`

HippoVlog does not require an LLM-as-a-judge because its multiple-choice answers can be verified directly.

F Broader Impacts and Ethical Considerations

F.1 Broader Impacts

CAM supports question answering over long, entity-centric videos. **Positive Impacts:** Potential applications include robotic assistance, accessibility tools, and video archiving and retrieval, where connecting observations across time can help users find relevant events and interactions. **Potential Harms:** The same capabilities could facilitate unauthorized monitoring, surveillance, or non-consensual profiling. In addition, errors in entity association or the abstraction of character traits and relationships may produce misleading characterizations of individuals. Biases in the underlying models may also influence these inferences. Such descriptions should therefore be treated as model-generated interpretations of observed content rather than verified personal attributes.

F.2 Safeguards and Code of Ethics

Our evaluation uses existing public benchmarks, including M3-Bench-robot, M3-Bench-web, and HippoVlog. CAM uses existing language and multimodal models for extraction, abstraction, and reasoning. Safety mechanisms provided by these models may offer some protection, but we do not evaluate their effectiveness as system-level safeguards for long-term video memory. Retaining episodic evidence and supporting video rewatching allow inferred descriptions to be checked against source observations, but do not guarantee factual correctness or prevent misuse.

Deployment should account for the sensitivity of persistent identity information and inferred personal attributes. Appropriate measures may include obtaining consent for video analysis, restricting access to stored memories, limiting data retention, and requiring human review before using inferred traits or relationships in consequential decisions. These are deployment considerations rather than safeguards validated by our experiments. We describe the framework's scope and limitations to support responsible use and avoid treating its outputs as verified judgments about individuals.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction describe the memory construction and adaptive querying mechanisms of Continuous Adaptive Memory (CAM), with claims scoped to the evaluated entity-centric video benchmarks.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 3.3 discusses the scope of the evaluation, dependence on extraction and entity-association accuracy, the limits of the noise-injection study, and future work on dynamic tool selection.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper focuses on empirical system design and agent workflows for multimodal large language models rather than mathematical theorems or theoretical proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The experimental settings, datasets, baselines, and implementation details are fully documented in Section 4.1 and Appendix A

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The implementation is publicly available at the URL provided in the abstract. The experiments use existing benchmarks and pretrained models, with access provided by their original sources.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Implementation details such as clip intervals, sampling rates, and thresholds are provided in Section 4.1, Appendix A, and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The evaluated long-video benchmarks (more than 100-hour video in total) are extremely large in scale. Given the extensive computational resources required to process long videos, alongside the numerous baselines and ablation configurations tested, running multiple independent trials to compute statistical significance and error bars is computationally expensive.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4 and Appendix D discuss token consumption and inference efficiency.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. Potential societal impacts, privacy risks, and considerations for responsible use are discussed in Appendix F.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Appendix F discusses both the positive applications and potential negative societal impacts of long-video tracking.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This work releases the implementation of a video question-answering framework, rather than new pretrained model weights or high-risk datasets. Potential risks and responsible deployment considerations are discussed in Appendix F.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The datasets, models, and baseline methods used in our experiments are properly credited. Their sources, licenses, and applicable usage terms are described in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release the CAM implementation under the MIT License with installation and usage documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our paper does not involve study participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: Our paper does not involve study participants.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The core methodology utilizes the LLM for tool orchestration and memory updates, which is detailed in Section 3.1

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.