Agentic Very Long Video Understanding

Aniket Rege[1,2] **Arka Sadhu**[1] **Yuliang Li**[1] **Kejie Li**[1] **Ramya Korlakai Vinayak**[2]
Yuning Chai[1] **Yong Jae Lee**[2] **Hyo Jin Kim**[1]
[1]Meta Reality Labs Research [2]University of Wisconsin-Madison
aniketr@cs.wisc.edu, kimhyojin@meta.com

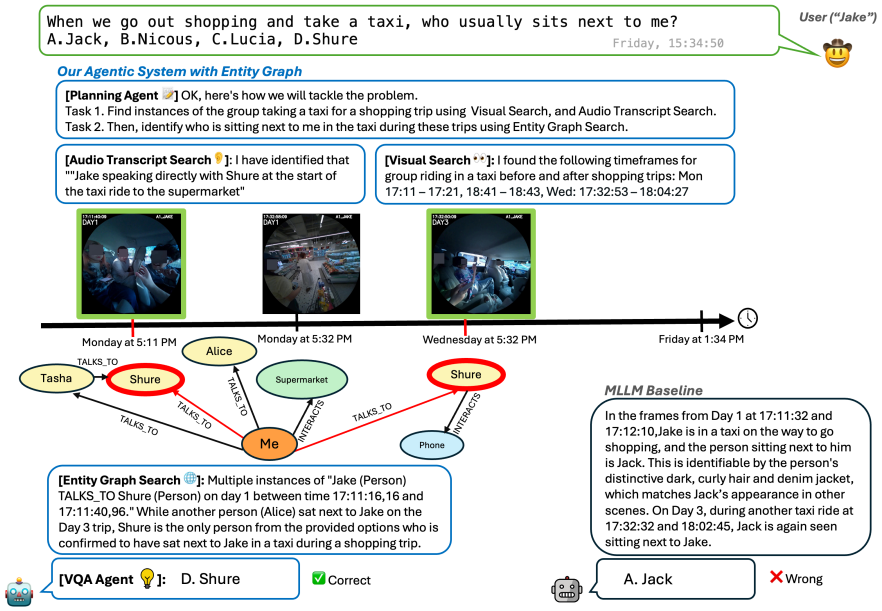

Figure 1: Given a natural-language query, our agentic entity-graph framework decomposes the task into subtasks and leverages visual search, audio transcript search, and entity scene graph search to identify relevant events spanning multiple days. This qualitative example highlights the framework's ability to perform multi-hop, cross-modal reasoning by first performing temporal localization using audio and visual cues, and then using the entity graph to infer the answer. The entity graph consists of nodes for *person*, *object*, or *location*, and edges capturing relations such as *talks-to* and *interacts-with*, each annotated with temporal intervals on when the relation holds.

Abstract

The advent of always-on personal AI assistants, enabled by all-day wearable devices such as smart glasses, demands a new level of contextual understanding; one that goes beyond short, isolated events to encompass the continuous, longitudinal stream of egocentric video. Achieving this vision requires advances in long-horizon video understanding, where systems must interpret and recall visual and audio information spanning days or even weeks. Existing methods, including large language models and retrieval-augmented generation, are constrained by limited context windows and lack the ability to perform compositional, multi-hop reasoning over very long video streams. In this work, we address these challenges through EGAgent, an enhanced agentic framework centered on entity scene graphs, which represent people, places, objects, and their relationships over time. Our system equips a planning agent with tools for structured search and reasoning over these graphs, as well as hybrid visual and audio search capabilities, enabling detailed, cross-modal, and temporally coherent reasoning. Experiments on the EgoLifeQA and Video-MME-long datasets show that our method achieves state-of-the-art performance on EgoLifeQA (57.5%) and competitive performance on Video-MME-long (74.1%) for complex longitudinal video understanding tasks. Code is available at https://github.com/facebookresearch/egagent.

1 Introduction

Unlocking always-on personal AI assistants requires understanding not just isolated events, but a continuous stream of evolving user experiences. The recent emergence of AI-equipped wearable consumer devices such as the Ray-Ban Meta glasses, Amazon Echo Frames and Snapchat Spectacles as well as various prototypes (Engel et al., 2023; Xu, 2025) creates an opportunity for AI

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 46575–46602

agents to maintain persistent access to what users see and do over time. For such assistants to provide helpful, personalized, and context-aware assistance, they need to possess *longitudinal video understanding*, *i.e.*, the ability to recall and interpret a user's lived experience over extremely long periods of time (days and months).

In this work, we address the challenge of "very long video understanding". In prior literature, the definition of "long" has been continuously evolving. Popular benchmarks like MSR-VTT (Xu et al., 2016) and DiDeMo (Anne Hendricks et al., 2017) where videos are up to a minute in length were once considered long, but recent works have further pushed this frontier to several minutes (Wu et al., 2024; Grauman et al., 2022) and up to an hour (Fu et al., 2025; Zhou et al., 2024a; Wang et al., 2025a). The recent EgoLife (Yang et al., 2025b) pushes this frontier to **beyond 50 hours** of Egocentric video over the course of a week, which is the length we define as **very long**. Unlike previous benchmarks that focus on large numbers of short, independent videos, EgoLife offers continuous, *longitudinal* first-person video from six individuals. This week-long horizon enables new research directions, such as tracking entities and their interactions across multiple days, analyzing repeated behaviors and habits, and handling extended periods of inactivity or "lulls" in the video stream. Agentic approaches, which equip agents with tools to search, retrieve, and reason over large corpora, have shown potential in addressing some of these limitations (Fan et al., 2024; Wang et al., 2024; Ma et al., 2025; Chu et al., 2025). Existing agentic approaches often struggle to maintain coherent reasoning about entities and their relationships over extended temporal horizons, and have difficulty with fine-grained temporal localization such as tracking repeated actions or habits across days (*e.g.*, "how often did I drink water this week?"). Importantly, effective linkage between information from different modalities is needed to support richer and more accurate reasoning.

To address these challenges, we propose **EGAgent**, an enhanced agentic approach that centers on the extraction and use of an **entity scene graph** from long videos, where nodes represent people, places, and objects, and edges capture their relationships (*e.g.*, uses, interacts with, mentions, talks to). Each edge is annotated with temporal intervals indicating when the relation holds. In our proposed EGAgent system, we equip a planning agent with the ability to search and reason over this entity graph, as well as utilize a visual search tool (SQL + semantic search hybrid) and an audio transcript search tool. As illustrated in Fig. 1, the system uses this graph in combination with audio and visual search to locate all shopping-related taxi rides across multiple days and infer who consistently sits next to the user. By leveraging structured representations like entity graphs, EGAgent preserves complex relationships and supports detailed, compositional reasoning over extended timeframes, overcoming the limitations of existing methods.

We evaluate our EGAgent pipeline on the EgoLifeQA benchmark and demonstrate state-of-the-art performance. Notably, EGAgent surpasses the previous state-of-the-art by 32% and 39.7% on the RelationMap and TaskMaster categories respectively, both of which require multi-hop relational reasoning. EGAgent also achieves competitive results on the Video-MME (Long) benchmark.

To summarize, our contributions are as follows:

- We introduce an entity graph representation (Sec. 3.2) for long video understanding (Sec. 3.1), enabling structured, cross-modal reasoning over very long time horizons.
- We present an agentic framework (Sec. 3.3) that queries the entity graph along with visual and audio search tools, exceeding previous state-of-the-art performance on EgoLifeQA by 20.6% (Sec. 4.3).
- We perform a detailed ablation study on entity graph construction and agentic tool usage for very long video understanding on EgoLife (Sec. 4.5 and App. D).

2 Related Work

Long Video Understanding with LLMs. The primary challenge in long-video understanding arises from the limited context window of large language models (LLMs), which restricts the amount of visual information processed at once. To address this, prior work focuses on condensing video inputs before LLM inference (Tang et al., 2025; Lu et al., 2025b; Liu et al., 2025a). Frame selection methods reduce input length by retaining only salient frames while preserving key content (Wang et al., 2025b; Buch et al., 2025; Ye et al., 2025), whereas visual token compression techniques distill videos into compact token representations that better fit within context limits (Shen et al., 2025; Shu et al.,

2025). These approaches can be query-dependent, selecting frames or tokens based on the input query (Liu et al., 2025b; Hu et al., 2025; Man et al., 2025; Diko et al., 2025), or query-independent, producing general summaries irrespective of downstream tasks (Yang et al., 2025a; Zhao et al., 2025). Other methods adopt sliding-window or hierarchical summarization strategies to maintain long-range context under fixed token budgets (Lu et al., 2025a; Zhou et al., 2024b), or to directly extend the context capacity of LLMs themselves (Ding et al., 2024; Liu et al., 2023; Jin et al., 2024).

Video Understanding with Graph-based RAG. Retrieval-augmented generation (RAG) mitigates the context limitations of LLMs by retrieving relevant information from external sources (Lewis et al., 2020; Gao et al., 2023), which has also been extended to multimodal documents and long-video understanding (Yu et al., 2025; Faysse et al., 2025). Traditional RAG operates over isolated text chunks, often losing relational context. To address this, Graph-based RAG methods such as GraphRAG (Edge et al., 2024) and LightRAG (Guo et al., 2024) leverage knowledge graphs built from extracted entities and relations from the text corpus. More recently, researchers have begun to explore multi-modal RAG approaches, such as retrieving image frames directly instead of retrieving pre-generated video captions (Reddy et al., 2025; Wan et al., 2025). This approach preserves visual details that may be lost in textual abstraction, enabling more precise and comprehensive responses to complex queries. For instance, Video-RAG (Luo et al., 2025) performs multi-modal RAG on video frames, automatic speech recognition (ASR) results, optical character recognition (OCR) results, and object-detection results. However, directly retrieving frames also introduces new challenges, including the need for efficient and accurate indexing, retrieval mechanisms, and effective data representations (Reddy et al., 2025). VideoRAG (Ren et al., 2025) combines text-, visual-, and graph-based clip retrieval, matching queries to entity descriptions within a graph. AdaVideoRAG (Xue et al., 2025) adaptively selects between no retrieval, naive retrieval, and graph-based retrieval based on question difficulty. RAVU (Malik et al., 2026) uses VLMs to detect entities, generate frame descriptions, build spatio-temporal graphs, and infer answers. GraphVideoAgent (Chu et al., 2025) iteratively retrieves relevant frames via caption-derived graphs. VideoMindPalace (Huang et al., 2025) constructs layered spatio-temporal graphs encoding indoor layouts and activity zones, though its reliance on room-level structure limits robustness in open-ended scenes.

Many of these methods either overlook temporal relationships or construct graphs for the entire video at once. In contrast, we introduce an entity graph where each node is annotated with temporal information, making the graph time-aware and allowing it to be incrementally constructed as new data arrives. Experimentally, our method matches the performance of AdaVideoRAG (Xue et al., 2025) on Video-MME (Long) while processing over ten times fewer frames.

Agentic Video Understanding. Recent advances in agentic video understanding have focused on developing systems that can autonomously perceive, reason, and act based on video content (Chen et al., 2025). VideoAgent (Wang et al., 2024) introduces an agent-based framework where the agent is tasked with iteratively finding the relevant frames in the video for VQA if the information in the initial frames is not sufficient to answer the question. VideoAgent (Fan et al., 2024) iteratively employs tools such as object memory search and video-segment search based on video captions and visual embeddings to reach an answer. DrVideo (Ma et al., 2025) reframes long-video understanding as long-document understanding by converting videos into text documents, iteratively augmenting them with key frame information and agent-based searches until enough information is gathered for chain-of-thought prediction. Similarly SiLVR (Zhang et al., 2026) operates in the text domain by compressing dense visual captions and using a downstream reasoning LLM for video understanding.

Our proposed EGAgent advances agentic video understanding by integrating a temporally-annotated entity scene graph into the tool-calling loop. Unlike prior systems that rely on unstructured captions or repeated frame retrieval, our approach enables efficient cross-modal search and compositional reasoning for complex, longitudinal queries.

3 Method

In this section, we formalize the task of very long video understanding (Sec. 3.1) and describe extracting entity graph representations of such long videos (Sec. 3.2). Lastly, we discuss the design of the proposed agentic framework EGAgent which utilizes these entity graph representations for very

long video understanding (Sec. 3.3).

3.1 Task Setup

We focus on the task of very long video understanding, specifically on video question-answering over videos that potentially span an entire week. Let $\mathcal{V} = \{v_t\}_{t=1}^{T}$ denote the video sampled at 1 FPS (frame per second). Similarly, let $\mathcal{AT} = \{u_i, t_{start_i}, t_{end_i}\}_{i=1}^{N}$ denote the set of transcribed speech u_i with associated time-stamps (t_{start_i}, t_{end_i}). At test time, the system receives a complex query Q in natural language, and must produce a textual answer A. Formally, the task is to obtain a mapping $H : (\mathcal{V}, \mathcal{AT}, Q) \rightarrow A$.

Naively feeding all frames and transcripts into a multimodal LLM or VLM for such very long videos is infeasible due to context window limitations. The prevailing approach, Video Retrieval Augmented Generation (RAG) (Luo et al., 2025), first selectively retrieves a small subset of frames and audio transcripts deemed relevant to the user query Q and conditions the VLM on this retrieved set to generate the answer A. However, a naive RAG approach over very long egocentric videos is insufficient to answer egocentric queries which are often entity-centric and require multi-hop reasoning across days. These include tracking repeated behaviors, or interactions between specific people, objects, and locations. Direct embedding-based retrieval over unstructured clips or captions struggles to maintain coherent entity identities over time to support compositional constraints such as "all times I talked to person X this week".

We address this in two steps. First, to support queries over entity relations over time, we construct an entity-centric scene graph that explicitly encodes people, objects, locations, temporally localized relations, and provide a structured index to allow narrowing down to the relevant regions of the video (Sec. 3.2). Second, we propose an agentic framework EGAgent which involves a planning agent that iteratively decomposes Q into sub-tasks and invokes specialized retrieval tools including the above constructed entity graph (Sec. 3.3).

3.2 Entity Graph Representations

From our observations, baseline methods often struggle with questions that require understanding a person's habits or repeated behaviors over time (*e.g.*, "What do I often check on my phone in the morning?"), as well as those that involve reasoning about interactions and relationships between different entities, such as people, objects, or places across extended periods (*e.g.*, "Before we went to see the dog, who went with me to the second floor to find Tasha?"). Because these methods do not explicitly model entity relationships or track long-term behavioral patterns, their performance on such questions, especially over long time horizons, is limited.

To address this, we construct an entity graph $G = (V, E)$ to capture relationships and interactions, enabling the planning agent to query this graph during inference.

- *Nodes (V)*: entities (*i.e.*, individuals, objects, places)
- *Edges (E)*: relationships (*i.e.*, interacts with, mentions, talks to, uses), and temporal information

Each edge is annotated with temporal information, allowing us to track the existence, sequence and duration of the corresponding relationships. Such temporal structure is crucial for reasoning about events and interactions that unfold or repeat across long horizons.

Entity Graph Creation. We construct an entity graph $G{=}(V, E)$ from a given collection of text documents $\mathcal{D}$ which includes audio transcripts, scene descriptions, predicted scene locations (illustrated in Fig. 3). We discuss details of extracting scene data to generate these documents $\mathcal{D}$ in App. F. For each document $d \in \mathcal{D}$, we apply an LLM-based extractor $\mathcal{F}$ to jointly identify entities and their relationships:

$$(V_d, E_d) = \mathcal{F}(d) \tag{1}$$

Here, V_d is the set of entities and E_d is the set of relationships extracted from d. The overall entities and relationships are aggregated as:

$$(V, E) = \left(\bigcup_{d \in \mathcal{D}} V_d, \bigcup_{d \in \mathcal{D}} E_d \right) \tag{2}$$

We assign each node $v{\in}V$ a type $\tau(v)$ to be one of "person", "object", "location". We initially represent each edge e as a tuple (v_s, v_t, r), where v_s and v_t are the source and target nodes, and $r \in \mathcal{R}$ is the relationship type. The set of relationship types is:

$$\mathcal{R} = \{\text{talks}-\text{to}, \text{interacts}-\text{with}, \text{mentions}, \text{uses}\} \tag{3}$$

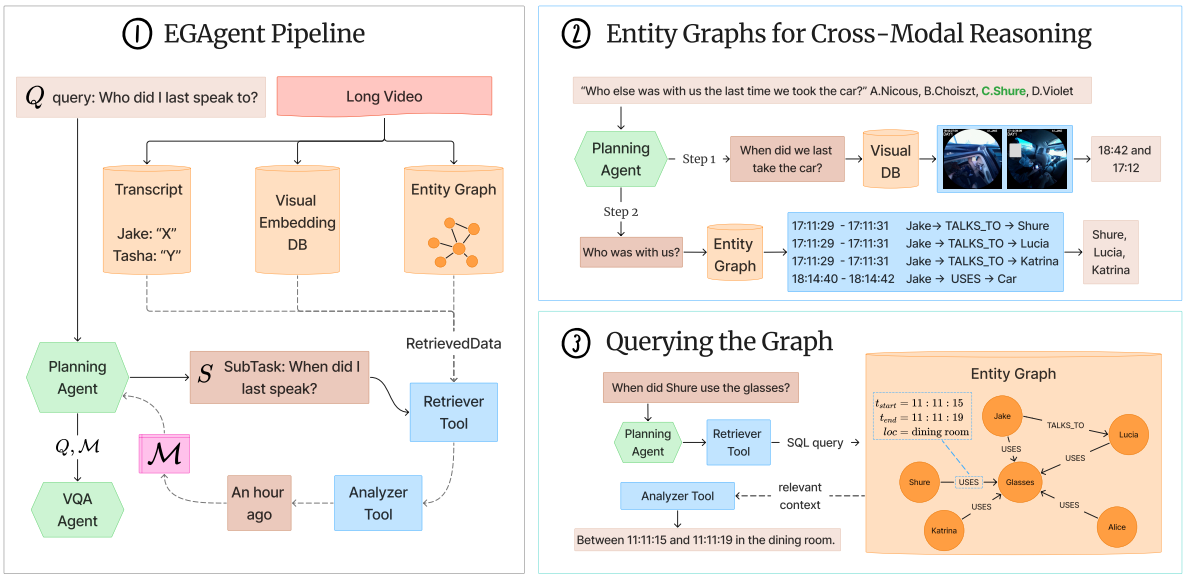

Figure 2: We show an overview of our EGAgent pipeline for very long video understanding using cross-modal reasoning in ①. Given a very long video and a query, a planning agent devises a multi-step plan of sub-tasks required to answer the query. The planning agent uses a retriever tool to probe three data sources extracted from the long video: audio transcripts, visual frame embeddings, and an entity scene graph, which is the focus of EGAgent. We show an example of how the planning agent composes cross-modal information retrieved from the visual database and entity graph to answer an EgoLife query in ②. We visualize the entity graph query mechanism in ③, where the retriever tool designs a SQL query to retrieve relevant relationships for the planning agent to reason over.

Each edge e is subsequently annotated with temporal information $(t_{\text{start}}, t_{\text{end}})$ derived from the source document d. After temporal annotation, each edge is represented as:

$$e = (v_s, v_t, r, t_{\text{start}}, t_{\text{end}}) \tag{4}$$

The resulting graph is stored as a set of tuples:

$$(v_s, \tau(v_s), v_t, \tau(v_t), r, t_{\text{start}}, t_{\text{end}}, d^*) \tag{5}$$

d^* is the supporting text snippet from which the edge was extracted. The graph is stored in memory as a SQLite3 database, with each row corresponding to one tuple. The graph construction process supports incremental updates as new documents d arrive, allowing G to grow and refine over time.

3.3 Agentic Framework EGAgent

Given the very long video and entity graph representation described above, we propose an agentic framework EGAgent for multi-modal reasoning, summarized in Algorithm 1 and illustrated in Fig. 2. EGAgent consists of six main components: a **Planning Agent**, three **Retriever Tools** (Visual Search, Audio Transcript Search, and Entity Graph Search), an **Analyzer Tool**, and a **VQA Agent** (see ① in Fig. 2). We discuss more details of our agent design and provide qualitative examples in App. B.

Each component operates over a specific data modality or reasoning step. The Planning Agent decomposes a complex user query Q into sub-tasks, selects appropriate tools, and maintains a working memory $\mathcal{M}$ that accumulates cross-modal evidence. Retriever Tools (Visual Search, Audio Transcript Search, Entity Graph Search) access different data sources to find relevant information for each sub-task, the Analyzer Tool filters and distills retrieved information, and the VQA Agent produces the final answer A from the accumulated evidence.

Planning Agent orchestrates the entire reasoning process. Given a user query Q along with natural language definitions for each tool, Planning Agent performs a joint decomposition of Q into a sequence of N sub-tasks $\{S_1, S_2, \ldots, S_N\}$ each sub-task with an associated $Tool_i$ and with appropriate query arguments q_i (Lines 2-3 in Algorithm 1).

Each sub-task S_i targets a specific aspect of the information needed such as object localization, checking diarized speech, or confirming past interaction. For each (S_i, T_i, q_i), the Planning Agent selects a retriever tool T_i from one of the following: (i) **Visual Search Tool** ($Tool_{\text{vis}}$) retrieves visual content. (ii) **Audio Transcript Search Tool** ($Tool_{\text{aud}}$) retrieves transcribed speech. (iii) **En-**

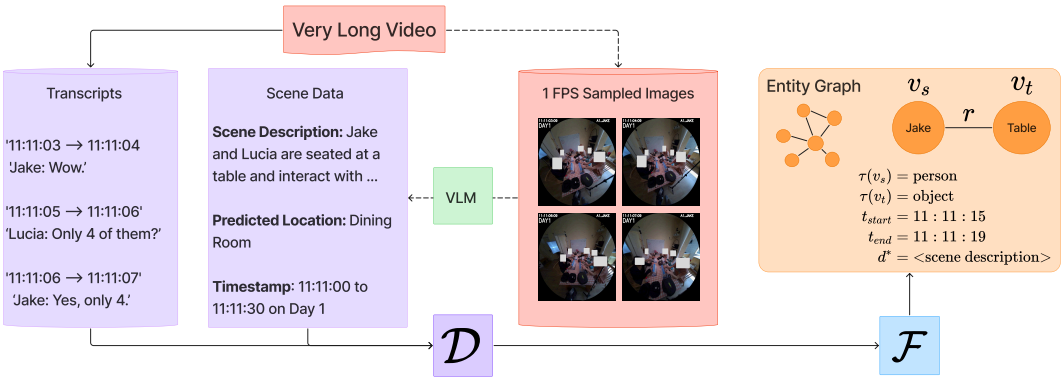

Figure 3: We use an LLM $\mathcal{F}$ to extract an entity graph from text documents $\mathcal{D}$ that represent a very long video, *i.e.*, audio transcripts $\mathcal{AT}$ and scene descriptions and locations extracted from sampled image frames $\mathcal{V}$ (see details in App. F). Each graph relationship r connects a source vertex v_s and target vertex v_t between time $(t_{\text{start}}, t_{\text{end}})$. Each vertex has an entity type $\tau(v)$ and the raw text document d^* used to extract the relationship (Sec. 3.3).

Algorithm 1 EGAgent Framework

Require: User query Q, Multimodal data sources (Video, Audio, Entity Graph)
Ensure: Final answer A
Initialize working memory $\mathcal{M} \leftarrow \emptyset$
// **Step 1: Joint Decomposition and Tool Selection**
SubtaskList $\leftarrow$ `PlanningAgent.decompose_and_select`(Q)
{SubtaskList = $\{(S_1, T_1, q_1), (S_2, T_2, q_2), \ldots, (S_N, T_N, q_N)\}$}
for each (S, T, q) in SubtaskList **do**
    // **Step 2a: Retrieve relevant data for the subtask**
    RetrievedData $\leftarrow T(q)$ {Visual: hybrid semantic/attribute search; Audio: transcript search; Entity Graph: SQL queries}
    // **Step 2b: Analyze retrieved data for relevance and evidence**
    Analysis $\leftarrow$ `AnalyzerTool.analyze`(RetrievedData, S) {LLM-based reasoning, evidence extraction, or filtering}
    // **Step 2c: Update working memory**
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{\text{Analysis}\}$
end for
// **Step 3: Final Synthesis**
$A \leftarrow$ `VQAAgent.answer`$(Q, \mathcal{M})$ {VQAAgent uses accumulated, cross-modal evidence in $\mathcal{M}$ to answer Q}
return A

tity Graph Search Tool ($Tool_{\text{eg}}$) queries an entity-centric scene graph. The retrieved content is passed to the **Analyzer Tool** and the corresponding analysis is updated to the working memory, $\mathcal{M}$. Such an iterative process allows EGAgent to progressively refine its understanding of the query Q while keeping per-sub-task context size manageable. Finally, the **VQA Agent** consumes the working memory and original query to provide a final answer. See ② in Fig. 2 for an example of the planning agent reasons over cross-modal information retrieved with retriever tools.

Visual Search Tool samples video frames at 1FPS and embeds each frame v_t as $\phi_I(v_t) \in \mathbb{R}^d$ using a vision-encoder (Tschannen et al., 2025). The generated embeddings along with attributes such as timestamp, location are stored in a vector database which supports efficient retrieval. At inference, the Planning Agent provides a text sub-query q_i (embedded as $\phi_T(q)$) and optional attribute filters f (*e.g.*, "kitchen", "morning"). The tool computes cosine similarity $\cos(\phi_T(q), \phi_I(x_t))$ for filtered rows in the vector database returning the k-nearest neighbors for further analysis.

Audio Transcript Search Tool operates over text transcripts. We consider two variants (i) LLM-based search where we feed entire transcripts to an LLM for a relevant time range (parallelized over days due to context limits) (ii) BM25-based lexical search. The former provides significantly better quality results at the cost of higher latency.

Entity Graph Search Tool queries the entity-centric scene graph G introduced in Sec. 3.2 and stored tuples in a SQLite database (Eq. (5)). During inference, the Planning Agent issues SQL queries q over the following fields: (i) time filter (ii) keyword text search (iii) entity source and/or target nodes (v_s, v_t) and (iv) relationship type r. In practice, real-world data is often incomplete or noisy, so the Planning Agent adopts a "strict-to-relaxed" query

strategy: it first issues an exact match query on all specified fields, and if no results are found, incrementally relaxes constraints by broadening the time window, allowing partial text matches, and finally relaxing the relationship type filter. This strategy maximizes precision when possible while increasing recall when exact matches are unavailable (see ③ in Fig. 2 for an example query trace and App. C for qualitative examples of SQL querying).

Analyzer Tool determines the relevance of the retrieved context for each sub-task S_i via an LLM to perform lightweight reasoning, evidence extraction, and optional de-duplication.

VQA Agent is a multi-modal LLM that conditions on Q and the compact evidence in $\mathcal{M}$ to generate the final answer A (Algorithm 1, Line 13), enabling detailed, temporally coherent reasoning over week-long egocentric videos.

4 Experiments

4.1 Evaluation Benchmarks

We evaluate EGAgent against baselines on two benchmarks, EgoLifeQA and Video-MME (Long), which focus on **very long** video understanding.

EgoLifeQA: EgoLifeQA consists of 500 long-context Multiple-Choice Questions (MCQs) derived from the EgoLife (Yang et al., 2025b) dataset, in which six participants lived together for one week, continuously recording their daily activities using Project Aria glasses (Engel et al., 2023). The benchmark focuses on the 50 hours of videos taken from the perspective of Jake, one of the six participants. The MCQs cover practical questions such as locating items, recalling past events, tracking habits, and analyzing social interactions. Each question has four candidate answers with a single correct option. Each question is associated with *query time* (*e.g.* 11:34 AM on day 4) and a manually verified *target time*, indicating the specific portion of the video that contains the information needed to answer the MCQ correctly.

Video-MME (Long): Video-MME (Fu et al., 2025) comprises 900 videos, with 2700 MCQs. The benchmark is divided into *Short*, *Medium*, and *Long* subsets based on video length. We focus on the *Long* subset that consists of 300 videos that range from 30 to 60 minutes. (Sec. 4.4).

4.2 Implementation Details

To prepare the entity graphs for our experiments, we extract a separate graph for each video in the Video-MME dataset. For EgoLifeQA, due to the increased likelihood of LLM invocation failures with longer input transcripts, we instead extract one graph per hour of video. In both datasets, audio is represented by text transcripts. For Video-MME, transcripts are generated using an ASR foundation model such as Whisper. In contrast, EgoLife provides manually diarized transcripts, which include both speaker identities and the corresponding speech content.

4.3 Analysis on EgoLifeQA Benchmark

We compare our approach against various strong baselines in three categories: 1) MLLM with uniform sampling; 2) MLLM with RAG; and 3) existing agentic approaches.

Baselines. To handle extremely long videos in EgoLifeQA, frame sampling in MLLM baselines varies based on their respective context window size. GPT-4.1 takes video captions that were generated for every 30-second video snippet sampled at 1 FPS. We sample 3000 frames uniformly along with the audio transcripts for Gemini 2.5 Pro. The results of LLaVa-Video-7B (Zhang et al., 2024) and LLaVA-Video-7B combined with Video-RAG (Luo et al., 2025) are reported in Yang et al. (2025b).

We compare our approach with the following existing agentic methods: EgoButler (Yang et al., 2025b), a hierarchical text-based Retrieval-Augmented Generation (RAG) approach, and Ego-R1 (Tian et al., 2025), a lightweight 3B-parameter agent trained on egocentric data, including portions of EgoLife for tool calling. We report results of all RAG and existing agentic approaches in Tab. 1 directly from these works.

Performance Analysis. Tab. 1 presents a comprehensive comparison of methods on the EgoLifeQA benchmark. Our EGAgent, which incorporates entity graph reasoning, achieves strong performance across all evaluation categories and establishes a new state-of-the-art. Notably, while Gemini 2.5 Pro with uniform sampling already outperforms the previous best results (EgoButler), our agentic system based on Gemini 2.5 Pro delivers an additional improvement of 10.7%, highlighting the significant value of entity graph reasoning.

Furthermore, the benefits of entity graph reasoning are not limited to a single MLLM backbone. Applying the same agentic framework to the GPT-4.1 backbone also yields notable gains over its uniform sampling counterpart. These results demonstrate that integrating entity graph reasoning within

Table 1: MCQ Accuracy on EgoLifeQA (Yang et al., 2025b). The previously reported state-of-the-art is underlined, the current state-of-the-art is bolded, and the current second-best italicized. Agentic approaches are given frames or captions sampled at 1FPS and then choose a subset X for analysis, which is denoted by 1FPS→X under # Frames. F = raw video frames, C = video captions, A = raw audio, T = audio transcript. "–" in results of individual categories denotes missing data as they were not reported in the original papers. We estimate token usage for these baselines, which are marked with an asterisk* (see F for details on estimation). The following are question type categories from EgoLifeQA, on whom we report MCQ Accuracy below: EL (EntityLog), ER (EventRecall), HI (HabitInsight), RM (RelationMap), TM (TaskMaster).

Category	Method	# Frames	Modality	MCQ Accuracy (%)						Average Gain (%)	Average # Tokens
				EL	ER	HI	RM	TM	Average		
MLLMs (Uniform Sampling)	LLaVA-Video-7B	64	F	–	–	–	–	–	36.4		32K*
	GPT-4.1	1FPS	C	32.0	39.7	39.3	32.8	39.7	36.0		285K
	Gemini 2.5 Pro	3000	F, T	45.6	48.4	51.7	41.6	52.4	46.8	+9.9	807K
RAG	LLaVA-Video-7B + Video-RAG	64	F	–	–	–	–	–	30.0		18K*
Agentic Baselines	EgoButler Gemini 1.5 Pro	0	C, T	36.0	37.3	45.9	30.4	34.9	36.9	+0	26K*
	EgoButler GPT-4o	0	C, T	34.4	42.1	29.5	30.4	44.4	36.2		19K*
	VideoAgent	1FPS→8	F	–	–	–	–	–	29.2		128K*
	LLaVA-OneVision-7B + T*	1FPS→8	F, T	–	–	–	–	–	35.4		32K*
	Ego-R1 Qwen-2.5-3B-Instruct	1FPS	F, C, T	–	–	–	–	–	36.0		128K*
Ours	EGAgent GPT-4.1 (F + T)	1FPS→50	F, T	48.0	48.4	55.7	40.0	61.9	48.6	+11.7	551K
	EGAgent GPT-4.1 (EG + F + T)	1FPS→50	F, C, T	44.0	49.2	55.7	53.6	66.7	50.7	+13.8	571K
	EGAgent GPT-4o (EG + F + T)	1FPS→50	F, C, T	44.8	*54.8*	*59.0*	44.0	61.9	44.6	+7.7	652K
	EGAgent Gemini 2.5 Pro (EG + F + T)	1FPS→50	F, C, T	**54.4**	**57.1**	**60.3**	**62.4**	**74.6**	**57.5**	**+20.6**	880K

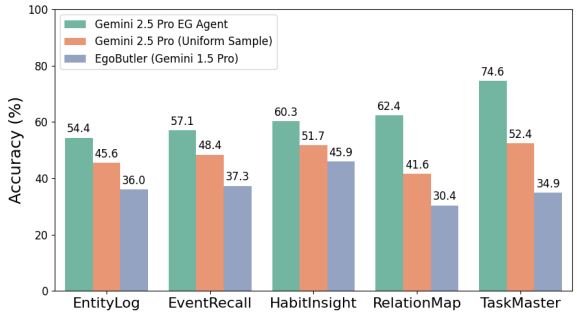

Figure 4: The performance comparison against Gemini 2.5 Pro and EgoButler in each question category in EgoLifeQA. Our approach significantly outperforms baselines on RelationMap (+20.8%) and TaskMaster (+22.2%), where entity understanding and complex reasoning is required to provide a correct answer.

agentic systems consistently enhances performance on very long video understanding tasks.

To compare with existing agentic systems, we run our EGAgent on the same LLM backbone (GPT-4o) with other agentic systems. Our EGAgent consistently surpasses other agentic approaches utilizing the same model, including EgoButler (+8.4%), VideoAgent (+15.4%), and Ego-R1 (+8.6%).

Fig. 4 further contrasts the performance gap between our proposed EGAgent and the Gemini 2.5 Pro with uniform sampling. It is evident that the agentic system benefits the most on RelationMap and TaskMaster categories that require multi-hop relational reasoning. Specifically, our approach surpasses the previous state-of-the-art and Gemini 2.5 Pro by 32% and 20.8%, respectively, on RelationMap QAs, and achieves impressive gains of 39.7% and 17.5% in the TaskMaster category. We discuss more examples and benchmark analyses in App. D.

4.4 Analysis on Video-MME Benchmark

We also evaluate our entity graph agent on the *long* subset of Video-MME in Tab. 2. Because Gemini 2.5 Pro can process native video (frames + audio) without the need for transcripts, it remains the state-of-the-art in this sub-hour length regime. Using an identical LLM backbone (Qwen2.5-VL-7B), EGAgent surpasses Video-RAG (+4.5%), and matches the performance of AdaVideoRAG while processing over 10× fewer frames.

Compared with recent agentic approaches (Guo et al., 2025; Yuan et al., 2025) that use frontier models as their LLM backbone, our EGAgent with a Gemini 2.5 Pro backbone demonstrates strong performance, second only to native Gemini 2.5 Pro that processes 256 frames. In contrast, EGAgent uses only a fifth of the image frames compared to the baseline. More importantly, uniformly sampling with MLLMs like Gemini 2.5 Pro does not scale well to extremely long videos, as demonstrated in the EgoLifeQA benchmark Sec. 4.3.

4.5 Ablations

Value of Entity Graph. Tab. 1 demonstrates that, in an apples-to-apples comparison using the same

Table 2: MCQ Accuracy on Video-MME (Long). The current state-of-the-art is bolded and the second-highest is underlined. F = raw video frames, C = video captions, A = raw audio, T = audio transcript, O = object detection bounding boxes. "1 FPS → 50" denotes retrieving 50 frames sampled at 1 FPS which are used for MLLM analysis. We estimate token usage wherever unreported, which are marked with an asterisk* (see App. F for details).

Category	Method	Context	# Frames	Modality	Accuracy (%)	# Tokens
MLLMs (Uniform Sampling)	Gemini 2.5 Pro	1M	256	F, A	**82.0**	100K*
	GPT-4.1	1M	384	F	72.0	60K*
RAG	Video-RAG (Qwen2.5-VL-7B)	32K	32	F, O, T	43.3	10K*
	AdaVideoRAG (Qwen2.5-VL-7B)	128K	768	F, C, T	47.7	128K*
Agentic Baselines	DrVideo (DeepSeek V2.5)	128K	0.2 FPS	F, T	71.7	128K*
	VideoDeepResearch (DeepSeek-r1-0528 + Qwen2.5VL-7B)	32K	32	F, T	72.4	32K*
Ours	EGAgent (Qwen2.5-VL-7B)	32K	1FPS→50	F, C, T	47.8	172K
	EGAgent (Gemini 2.5 Pro)	1M	1FPS→50	F, C, T	<u>74.1</u>	134K

Table 3: A comparison on EgoLifeQA of Entity Graph Extraction (EGX) using only transcript (T) vs a transcript-fused caption (C+T), and swapping out the transcript search tool from an LLM search to BM25 lexical search. All EGAgent methods reason over the entity graph, frames and audio transcripts (EG + F + T). EgoButler uses transcript-fused captions (C +T). All gains (%) are with respect to EgoButler GPT-4o.

Method	VLM	EGX	# F	T Search	Accuracy (%)	Gain (%)
EgoButler	GPT-4o	–	0	LLM	36.2	-
EGAgent	GPT-4o	T	50	BM25	36.6	+0.4
		T+C	50	BM25	39.4	+3.2
		T+C	50	LLM	44.6	+8.4
	GPT-4.1	T	0	-	36.8	+0.6
		T	50	BM25	42.2	+6.0
		T	50	LLM	49.2	+13.0
		T+C	50	BM25	43.9	+7.7
		T+C	50	LLM	50.7	+14.5
	Gemini 2.5 Pro	T	50	BM25	48.6	+12.4
		T+C	50	BM25	51.8	+15.6
		T+C	50	LLM	57.5	+21.3

Table 4: Wall-clock runtime of EGAgent that reasons over the entity graph, frames and audio transcripts (EG + F + T) on EgoLifeQA.

Method	T Search	Accuracy (%)	Runtime (sec)	#Tokens
EGAgent GPT-4.1	BM25	43.9	125	172K
	LLM	50.7	169	571K

backbone, the proposed method incorporating an entity graph substantially improves performance on EgoLifeQA. It outperforms the baseline (without entity graph) in 4 out of 5 categories, with particularly notable gains in the RelationMap and TaskMaster categories. This improvement can be attributed to the entity graph's ability to enable cross-modal reasoning. Furthermore, as demonstrated in App. D.3 (Tab. 6), EGAgent's entity graph substantially improves temporal localization, improving recall@10s from 0.232 to 0.884.

Extraction of Entity Graph. We compare two variants of Entity Graph Extraction (EGX) in EgoLifeQA in Tab. 3. The additional information from visual captions increases MCQ accuracy by ∼2.6% on average across all three MLLM backbones (GPT-4o, GPT-4.1 and Gemini 2.5 Pro).

Agent Wall-Clock Latency. We tabulate the wall-clock latency of EGAgent pipeline in Tab. 4. EGAgent takes 2-3 minutes to answer an MCQ, depending on the number of sub-tasks required by the planning agent. We also evaluate the latency impact of the transcript search and replace the default LLM search with BM25 (Robertson et al., 2009), which drops token usage by 3.3× at the cost of a ∼6.8% MCQ accuracy drop on average.

We discuss more ablations on tool usage and retrieval accuracy in App. D.

5 Conclusion

We introduce a novel EGAgent framework (Sec. 3.3) for longitudinal video understanding, addressing the unique challenges posed by always-on personal AI assistants processing very long egocentric videos. By leveraging entity scene graphs (Sec. 3.2) and specialized tools for structured, cross-modal reasoning, our approach enables detailed and temporally coherent analysis. Experiments on EgoLifeQA (Sec. 4.3) and Video-MME long (Sec. 4.4) demonstrate state-of-the-art performance on tasks requiring the tracking of entities, behaviors, and relationships over extended periods. As video lengths continue to grow, we believe our results highlight the potential of agentic planning over structured representations of inter-entity relationships for very long video understanding moving forward.

6 Limitations

While our EGAgent achieves strong performance on longitudinal video understanding tasks, it is important to note that the construction of entity scene graphs depends on the accuracy of upstream perception and language models, which may occasionally introduce errors in extracting entities and relationships. Additionally, our experiments relied on transcripts and for EgoLife, manually annotated speaker diarization. In scenarios where off-the-shelf diarization models are used, downstream performance is likely to be adversely affected by prediction errors. For specific examples of failure cases, please refer to App. E.

7 Ethical Considerations

Our work uses the publicly available EgoLife dataset, which was released under an MIT license. We adhere to all terms of use associated with this dataset. The EgoLife dataset automatically detects and blurs faces and other personally identifiable information (PII) such as sensitive audio content. We also use the Video-MME dataset, which was released under a custom license[1]. We have adhered to all terms of use associated with this dataset, using an unmodified version strictly for academic research. In addition to these pre-existing safeguards, we have taken extra care to protect individual privacy in our reporting: all faces appearing in the figures throughout this paper have been blurred.

References

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *ICCV*.

Shyamal Buch, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. 2025. Flexible frame selection for efficient video reasoning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29071–29082.

Boyu Chen, Zhengrong Yue, Siran Chen, Zikang Wang, Yang Liu, Peng Li, and Yali Wang. 2025. Lvagent: Long video understanding by multi-round dynamical collaboration of mllm agents. *arXiv preprint arXiv:2503.10200*.

Meng Chu, Yicong Li, and Tat-Seng Chua. 2025. Graphvideoagent: Enhancing long-form video understanding with entity relation graphs. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 4639–4648.

Anxhelo Diko, Tinghuai Wang, Wassim Swaileh, Shiyan Sun, and Ioannis Patras. 2025. Rewind: Understanding long videos with instructed learnable memory. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13734–13743.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, and 1 others. 2023. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561*.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. 2024. Videoagent: A memory-augmented multimodal agent for video understanding. In *ECCV*, pages 75–92. Springer.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2025. Colpali: Efficient document retrieval with vision language models. In *ICLR*.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2025. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *CVPR*, pages 24108–24118.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, and 1 others. 2022. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, pages 18995–19012.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

[1]License: https://github.com/MME-Benchmarks/Video-MME

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*.

Kai Hu, Feng Gao, Xiaohan Nie, Peng Zhou, Son Tran, Tal Neiman, Lingyun Wang, Mubarak Shah, Raffay Hamid, Bing Yin, and 1 others. 2025. M-llm based video frame selection for efficient video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13702–13712.

Zeyi Huang, Yuyang Ji, Xiaofang Wang, Nikhil Mehta, Tong Xiao, Donghyun Lee, Sigmund Vanvalkenburgh, Shengxin Zha, Bolin Lai, Licheng Yu, and 1 others. 2025. Building a mind palace: Structuring environment-grounded semantic graphs for effective long video analysis with llms. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24169–24179.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. Llm maybe longlm: Self-extend llm context window without tuning. *arXiv preprint arXiv:2401.01325*.

AI LangChain. 2025. LangGraph: Building language agents as graphs. https://github.com/langchain-ai/langgraph. Accessed: 2025-11-01.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeurIPS*, 33:9459–9474.

Junyi Liu, Liangzhi Li, Tong Xiang, Bowen Wang, and Yiming Qian. 2023. Tcra-llm: Token compression retrieval augmented large language model for inference cost reduction. *arXiv preprint arXiv:2310.15556*.

Shuming Liu, Chen Zhao, Tianqi Xu, and Bernard Ghanem. 2025a. Bolt: Boost large vision-language model without training for long-form video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3318–3327.

Zhihang Liu, Chen-Wei Xie, Pandeng Li, Liming Zhao, Longxiang Tang, Yun Zheng, Chuanbin Liu, and Hongtao Xie. 2025b. Hybrid-level instruction injection for video token compression in multi-modal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8568–8578.

Yujie Lu, Yale Song, William Wang, Lorenzo Torresani, and Tushar Nagarajan. 2025a. Vited: Video temporal evidence distillation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8501–8511.

Zijia Lu, ASM Iftekhar, Gaurav Mittal, Tianjian Meng, Xiawei Wang, Cheng Zhao, Rohith Kukkala, Ehsan Elhamifar, and Mei Chen. 2025b. Decafnet: Delegate and conquer for efficient temporal grounding in long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24066–24076.

Yongdong Luo, Xiawu Zheng, Guilin Li, Shukang Yin, Haojia Lin, Chaoyou Fu, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, and Rongrong Ji. 2025. Video-RAG: Visually-aligned retrieval-augmented long video comprehension. In *NeurIPS*.

Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. 2025. Drvideo: Document retrieval based long video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18936–18946.

Sameer Malik, Ayush Singh, Moyuru Yamada, and Dishank Aggarwal. 2026. Ravu: Retrieval augmented video understanding with compositional reasoning over graph. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2869–2878.

Yuanbin Man, Ying Huang, Chengming Zhang, Bingzhe Li, Wei Niu, and Miao Yin. 2025. Adacmˆ 2: On understanding extremely long-term video with adaptive cross-modality memory reduction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8534–8544.

Arun Reddy, Alexander Martin, Eugene Yang, Andrew Yates, Kate Sanders, Kenton Murray, Reno Kriz, Celso M de Melo, Benjamin Van Durme, and Rama Chellappa. 2025. Video-colbert: Contextualized late interaction for text-to-video retrieval. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19691–19701.

Xubin Ren, Lingrui Xu, Long Xia, Shuaiqiang Wang, Dawei Yin, and Chao Huang. 2025. Videorag: Retrieval-augmented generation with extreme long-context videos. *arXiv preprint arXiv:2502.01549*.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, and 1 others. 2025. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *ICML*.

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

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. 2025. Adaptive keyframe sampling for long video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29118–29128.

Shulin Tian, Ruiqi Wang, Hongming Guo, Penghao Wu, Yuhao Dong, Xiuying Wang, Jingkang Yang, Hao Zhang, Hongyuan Zhu, and Ziwei Liu. 2025. Ego-r1: Chain-of-tool-thought for ultra-long egocentric video reasoning. *arXiv preprint arXiv:2506.13654*.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, and 1 others. 2025. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*.

David Wan, Han Wang, Elias Stengel-Eskin, Jaemin Cho, and Mohit Bansal. 2025. Clamr: Contextualized late-interaction for multimodal content retrieval. *arXiv preprint arXiv:2506.06144*.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, and 1 others. 2025a. Lvbench: An extreme long video understanding benchmark. In *CVPR*, pages 22958–22967.

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

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

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

Chi Xu. 2025. Designing xreal one pro: the next generation of ost glasses. In *SPIE AR, VR, MR Invited Talks 2025*, volume 13415, page 1341504. SPIE.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, pages 5288–5296.

Zhucun Xue, Jiangning Zhang, Xurong Xie, Yong Liu, Xiangtai Li, Dacheng Tao, and 1 others. 2025. Adavideorag: Omni-contextual adaptive retrieval-augmented efficient long video understanding. In *NeurIPS*.

Chenyu Yang, Xuan Dong, Xizhou Zhu, Weijie Su, Jiahao Wang, Hao Tian, Zhe Chen, Wenhai Wang, Lewei Lu, and Jifeng Dai. 2025a. Pvc: Progressive visual token compression for unified image and video processing in large vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24939–24949.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, and 1 others. 2025b. Egolife: Towards egocentric life assistant. In *CVPR*, pages 28885–28900.

Jinhui Ye, Zihan Wang, Haosen Sun, Keshigeyan Chandrasegaran, Zane Durante, Cristobal Eyzaguirre, Yonatan Bisk, Juan Carlos Niebles, Ehsan Adeli, Li Fei-Fei, and 1 others. 2025. Re-thinking temporal search for long-form video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8579–8591.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and 1 others. 2025. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. In *ICLR*.

Huaying Yuan, Zheng Liu, Junjie Zhou, Ji-Rong Wen, and Zhicheng Dou. 2025. Videodeepresearch: Long video understanding with agentic tool using. *arXiv preprint arXiv:2506.10821*.

Ce Zhang, Yan-Bo Lin, Ziyang Wang, Mohit Bansal, and Gedas Bertasius. 2026. SiLVR: A simple language-based video reasoning framework. *Transactions on Machine Learning Research*.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2024. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*.

Shiyu Zhao, Zhenting Wang, Felix Juefei-Xu, Xide Xia, Miao Liu, Xiaofang Wang, Mingfu Liang, Ning Zhang, Dimitris N Metaxas, and Licheng Yu. 2025. Accelerating multimodal large language models by searching optimal vision token reduction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29869–29879.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2024a. Mlvu: Benchmarking multi-task long video understanding. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13691–13701.

Xingyi Zhou, Anurag Arnab, Shyamal Buch, Shen Yan, Austin Myers, Xuehan Xiong, Arsha Nagrani, and Cordelia Schmid. 2024b. Streaming dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18243–18252.

Acknowledgements

This work was supported in part by NSF IIS2404180 and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration.

A Overview

Design Details and Qualitative Examples. We provide details of EGAgent design and a visual walkthrough of our entire EGAgent pipeline in App. B with a qualitative example. We demonstrate how the planning agent invokes retrieval tools to retrieve relevant context from the very long video and continuously update the working memory. We illustrate how we query our entity graph in App. C.
Ablations on EgoLifeQA. We provide additional empirical analyses on EgoLifeQA (Yang et al., 2025b) in App. D, including evaluating oracle search, the importance of each search tool, retrieval accuracy of our three search tools, and wall-clock latency and memory cost of each component of EGAgent.
Implementation Details. We provide the prompts and code snippets we use for our planning agent, to extract and temporally annotate our entity graph, to query our search tools, and other implementation details in App. F.

B Qualitative Example of EGAgent Pipeline

We illustrate an example of our entire pipeline on a query from EgoLifeQA in Fig. 5. Given the query, the planning agent identifies high-level tasks, and comes up with a sequence of N sub-tasks ($N < 6$). In this example, the planner generated 5 tasks, S_1 through S_5. Each sub-task is routed to the appropriate search tool T_i. In this example, S_1 is routed to $Tool_{\text{vis}}$ to select relevant frames from the Visual DB with query q_1 = "people dancing". These retrieved frames are then sent to the analyzer tool, which observes that people are dancing on day 2 between 15:50 and 16:07, without knowledge of their identities. Similarly, S_2 is routed to $Tool_{\text{eg}}$ to search for social relationships in Entity Graph, which we describe in more detail in Fig. 6. Given the sub-task S_2, the planning agent uses a strict-to-relaxed hierarchy to choose a SQL query q_2 to search the entity graph to answer the sub-task, *i.e.* graph entities $\tau(v_s)$ = Person, r = TALKS_TO and $(t_{\text{start}}, t_{\text{end}})$ to search between. The retrieved rows of the SQL table are sent to the analyzer tool, and the relevant inter-entity relationships $(v_s, \tau(v_s), v_t, \tau(v_t), r, t_{\text{start}}, t_{\text{end}}, d^*)$ are appended to the working memory $\mathcal{M}$. We highlight one such relationship in Fig. 6, *i.e.*, Shure saying "Got it." to Alice between 3:50:21 PM and 3:50:22 PM, which overlaps with the dancing activity (S_1 in Fig. 5), indicating that both Shure and Alice take part in dancing. The planning agent proceeds until all remaining sub-tasks are routed to their appropriate search tool T_i with query arguments q_i and analyzed by the analyzer tool. The analysis output from each subsequent tool S_3, S_4, S_5 is also appended to the working memory $\mathcal{M}$. Once all sub-tasks are complete, the original query Q and working memory $\mathcal{M}$ are sent to the VQA agent to predict the answer A.

C Entity Graph

We show a qualitative example of how we query the entity graph in our EGAgent pipeline in Fig. 6. We also discuss our temporal annotation of entity graph edges, a novel contribution that enables EGAgent to temporally localize relevant relationships for a given query. We provide the implementation details and the prompts we use to construct the entity graph and temporally annotate edges in App. F.

We also provide some statistics of the entity graph we extract from EgoLife. In total, we extract 13968 relationships over a 7 day period. We visualize the relationships extracted for each day in Fig. 7. A vast majority of relationships have source node "person" (13930 / 13968), while the target node is more balanced (1314 "location", 6449 "object" and 6167 "person"). This indicates that the graph is focused more on person-person and person-object interactions, while also capturing person-location information.

D Ablation Study on EgoLife

Here we provide some additional ablation studies on EgoLife. We focus on tool usage, upper bound performance when using oracles, retrieval accuracy of tool search, and wall clock latency of our EGAgent pipeline.

D.1 Ablation on tool usage

To evaluate the importance of each search tool T on EGAgent performance, we evaluate our EGAgent

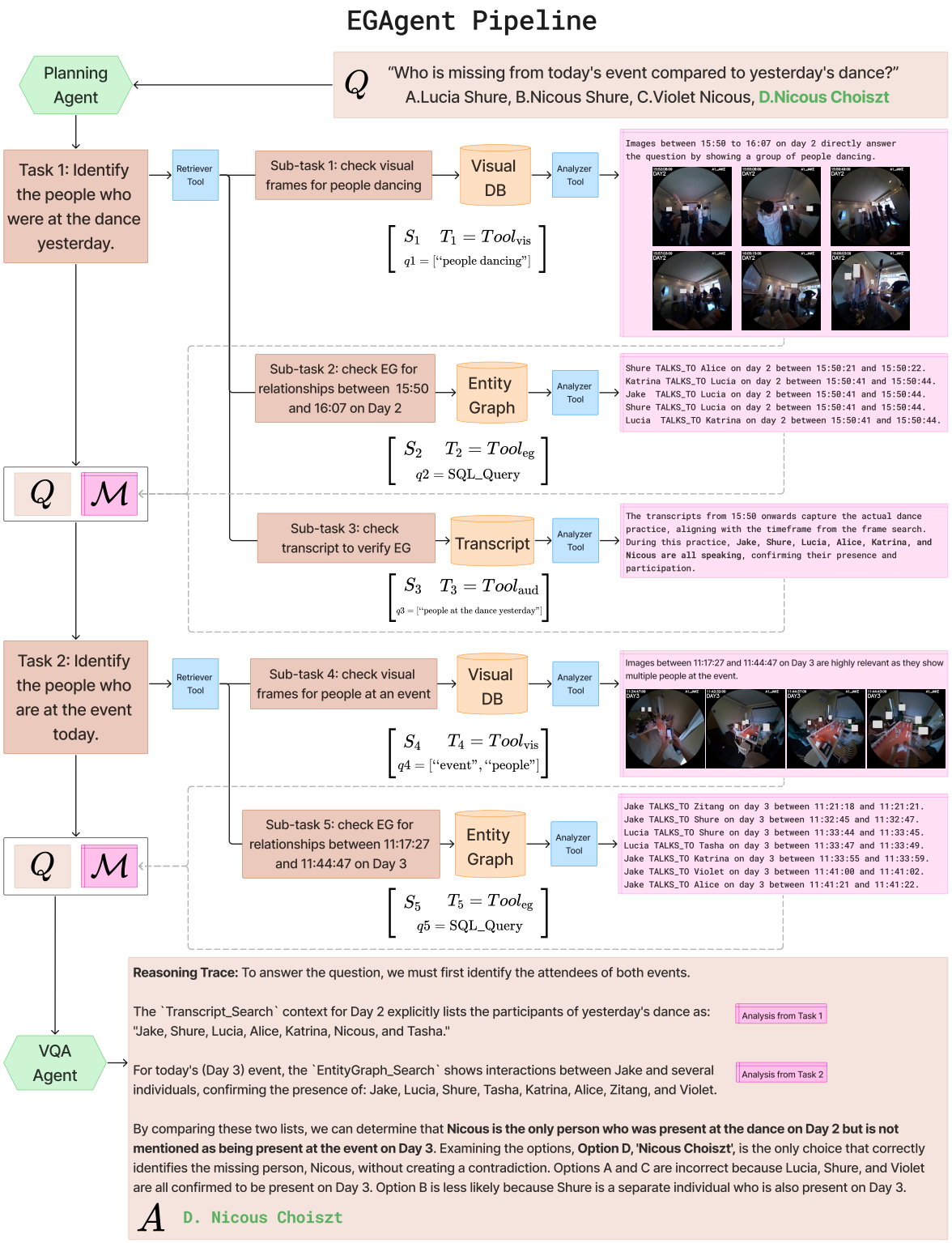

Figure 5: A walkthrough of our entire EGAgent pipeline (Sec 3.3, main paper) for an example query from EgoLifeQA, with more details in App. B. At a high-level, given the query, the planning agent comes up with a sequence of 5 sub-tasks, *i.e.*, S_1 through S_5. Each sub-task is routed to the appropriate search tool T_i followed by the analyzer tool, whose output is appended to the working memory $\mathcal{M} \leftarrow \mathcal{M} \cup$ Analysis. Once all sub-tasks are complete, the original query Q and working memory $\mathcal{M}$ are sent to the VQA agent to predict the answer A. The SQL_Query and the details about the entity graph search is illustrated in Fig. 6.

with all possible combinations of tools in Tab. 5.

We observe that using only the frame search tool performs poorly as the agent has no sense of entity identities. This reflects in its near-random perfor-

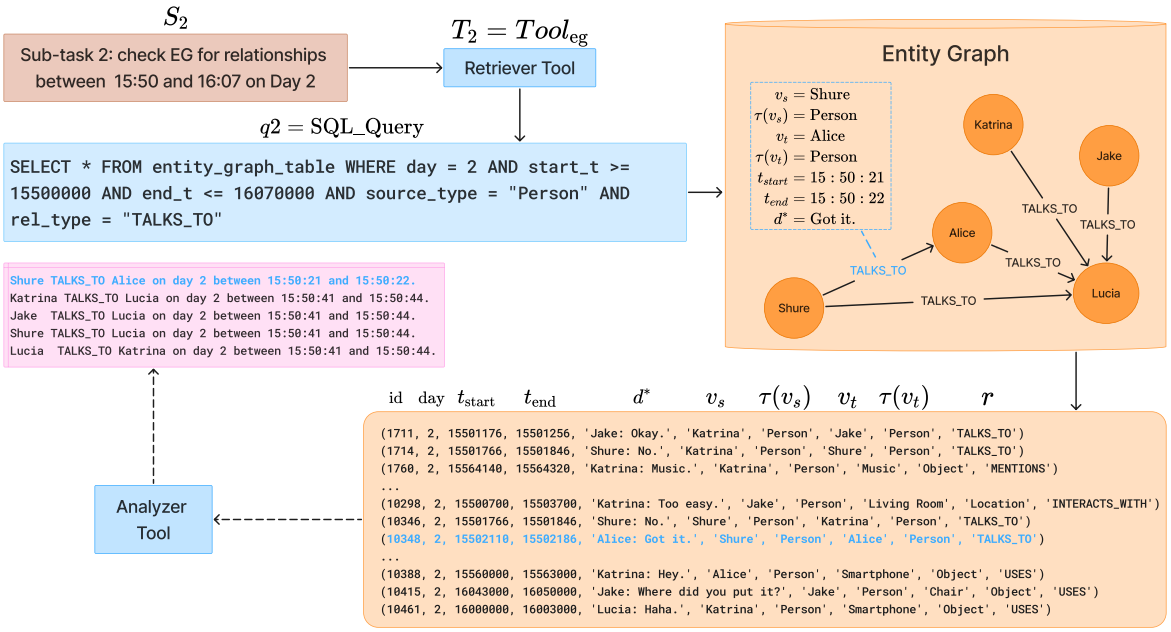

Figure 6: Here we focus on the entity graph search tool $Tool_{\text{eg}}$ in the example from Fig. 5 and discuss its role in the overall EGAgent pipeline in App. B. Given the sub-task S_2, the planning agent uses a strict-to-relaxed hierarchy to choose a SQL query q_2 to search the entity graph to answer the sub-task, *i.e.*, graph entities $\tau(v_s) = \text{Person}$, $r = \text{TALKS\_TO}$ and $(\text{day}, t_{\text{start}}, t_{\text{end}})$ to search between. The relevant rows of the SQL table are sent to the analyzer tool, and the relevant inter-entity relationships $(v_s, \tau(v_s), v_t, \tau(v_t), r, t_{\text{start}}, t_{\text{end}}, d^*)$ are appended to the working memory $\mathcal{M}$.

Table 5: Ablation study on impact of each tool on MCQ accuracy across EgoLifeQA task types. We equip EGAgent with various combinations of search tools (EG for $Tool_{\text{eg}}$, F for $Tool_{\text{vis}}$, and T for $\mathcal{M}_{aud}$). EGAgent highlights the importance of cross-modal reasoning (EG, F, T) by showing strong performance on all task types, especially those requiring inter-entity relationships (RelationMap).

Method	Modality	MCQ Acc (%)						Average Gain (%)	Average # Tokens
		EntityLog	EventRecall	HabitInsight	RelationMap	TaskMaster	Average		
EgoButler Gemini 1.5 Pro	C, T	36.0	37.3	45.9	30.4	34.9	36.9	+0	–
EGAgent GPT-4.1 (EG)	C	38.4	42.9	31.1	31.2	44.4	37.6	+0.7	21K
EGAgent GPT-4.1 (F)	F	40.0	37.3	31.1	28.0	34.9	34.6	-2.3	131K
EGAgent GPT-4.1 (T)	T	32.8	42.9	**59.0**	44.0	66.6	45.6	+8.7	438K
EGAgent GPT-4.1 (F + T)	F, T	**48.0**	48.4	55.7	40.0	61.9	48.6	+11.7	560K
EGAgent GPT-4.1 (EG+ F + T)	F, C, T	44.0	**49.2**	55.7	**53.6**	**66.7**	**50.7**	+13.8	571K

mance on RelationMap (28%), while its performance on more visual-focused tasks like EntityLog (40%) and EventRecall (37.3%) remains strong compared to EgoButler. When we add the powerful audio transcript search tool to EGAgent (T), the accuracy significantly improves for HabitInsight (+13.1%), RelationMap (+13.6%), and TaskMaster (+31.7%), while dropping slightly on the more visual-focused EntityLog (-3.2%). Using only the audio transcript search tool $Tool_{\text{aud}}$, EGAgent (T) performs the best on HabitInsight and TaskMaster, as these types of questions are more dependent on repeated and time-localized utterances from the audio transcripts. When we add the visual search tool $Tool_{\text{vis}}$ to EGAgent (F+T), it slightly drops performance on audio and entity-relationship focused tasks HabitInsight (-3.3%), TaskMaster (-4%), and RelationMap (-5%) compared to EGAgent (T), but similar to EGAgent (F), improves significantly on visual-focused tasks EntityLog (+15.2%) and slightly on EventRecall (+5.5%). Finally, when we add the entity graph search tool $Tool_{\text{aud}}$, we get state-of-the-art performance on entity-focused RelationMap (+23.2% over EgoButler), TaskMaster (+31.8% over EgoButler) and EventRecall (+11.9% over EgoButler), while remaining competitive on EntityLog and HabitInsight.

In summary, equipping EGAgent with the entity

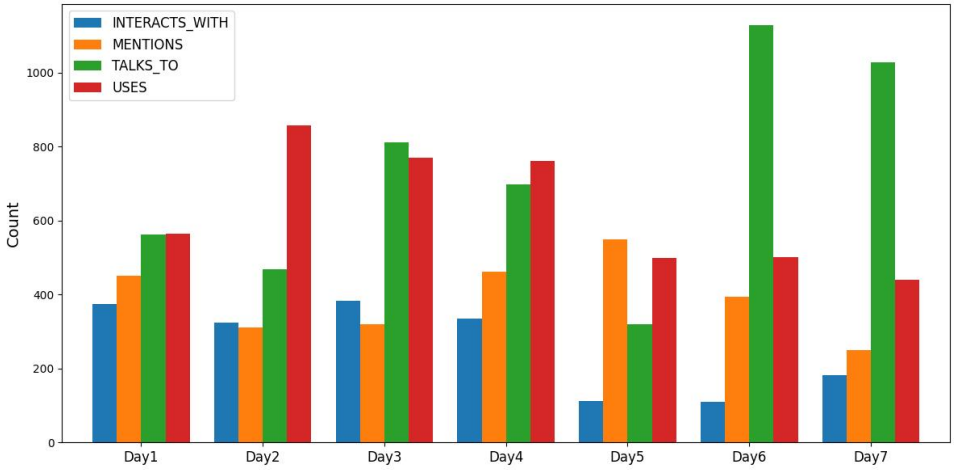

Figure 7: Entity Graph relationship types extracted from all seven days of EgoLife.

graph search tool $Tool_{\text{eg}}$ in addition to the standard visual and audio search tools $Tool_{\text{vis}}$ and $Tool_{\text{aud}}$ is crucial for robust performance on tasks requiring knowledge distributed across modalities, *e.g.*, inter-entity relationships (RelationMap, HabitInsight), audio triggers (TaskMaster), and visual-focused tasks (EntityLog, EventRecall). This result indicates that for agents to robustly understand long videos, it is important that they can search across modalities and reason over this fused context (cross-modal reasoning).

D.2 Oracles Indicate Room for Growth in Temporal Localization

To evaluate the upper bound performance, we use the ground-truth relevant moments (target_time) as oracle information for visual and audio transcript search. For visual search, we uniformly sample 50 frames at 1 FPS centered on the timestamps from (target_time), and for audio transcript search, we extract the entire transcript from the ground-truth day. As seen in Tab. 6, using both a visual and audio transcript oracle outperforms EGAgent (EG + F + T) by 6.9% with GPT 4.1 and 11.2% with Gemini 2.5 Pro. This shows that there is still room for improvements in MCQ accuracy that can be enabled by better temporal localization over very long videos.

D.3 Retrieval Accuracy

Our oracle upper bound experiments in Sec. D.2 highlight that precise temporal localization enables strong MCQ accuracy on EgoLifeQA, and that retrieval quality is an important factor in the success of agentic approaches for very long video understanding. To evaluate where the strength of our agent is coming from, we do a simple recall analysis on EgoLifeQA. We examine the working memory $\mathcal{M}$ for each multiple-choice question in the dataset, and extract the portions added by each search tool $\mathcal{M}_{eg}$ by $Tool_{\text{eg}}$, $\mathcal{M}_{vis}$ by $Tool_{\text{vis}}$, and $\mathcal{M}_{aud}$ by $Tool_{\text{aud}}$.

Each multiple-choice question in EgoLifeQA mcq_i contains total_i ground-truth timestamps in (target_time). To evaluate the quality of search of our tools, we compute a recall over these ground-truth timestamps with each of our search tools in Tab. 7. We denote the number of timestamps that each search tool searches over by "Input #ts", and the number of relevant timestamps highlighted by the analyzer tool (which is added to working memory $\mathcal{M}$) by "Selected #ts". For example, the Multimodal LLM baseline (Gemini 2.5 Pro) is provided 3000 uniformly sampled timestamps (Input #ts), from which it selects ~ 3.1 as relevant to the query (Selected #ts).

Since the provided target_time are discrete (HH:MM:SS), we use time windows centered on the target_time in our computation. For a given mcq_i and search tool, we record a $\text{hit}_{w,i}$ if any timestamp selected by the search tool (*i.e.*, one of Selected #ts) lies in the temporal window W. We define recall@windowsize (recall@W) over the $N = 500$ MCQ in EgoLifeQA as follows:

$$recall@W = \sum_{i=1}^{N} \frac{\text{hits}_{w,i}}{\text{total}_i}$$

We vary the size of these windows from 10 seconds up to one hour to measure how recall saturates as we relax the strictness of temporal localization.

Table 6: We use the ground-truth timestamps provided by EgoLifeQA to evaluate visual and audio transcript oracles, *i.e.*, search has perfect precision (1.0). F = raw video frames, C = video captions, A = raw audio, T = audio transcript. "1 FPS $\rightarrow$ 50" denotes retrieving 50 frames from those sampled at 1 FPS, with only these 50 frames used for MLLM analysis. We observe that there is still a gap between EGAgent tool search and perfect search, but perfect search still saturates at sub 70% accuracy with the state-of-the-art multimodal LLM.

Method	# Frames	Modality	MCQ Acc (%)	Average Gain (%)	Average # Tokens
EgoButler Gemini 1.5 Pro	0	C, T	36.9	+0	-
GPT 4.1 Prev4Day	0	T	45.6	+8.7	700K
GPT 4.1 Oracle	0	T	52.0	+15.1	243K
	50	F, T	57.6	+20.7	274K
Gemini 2.5 Pro Oracle	0	T	57.9	+21.0	332K
	50	F, T	**68.7**	**+31.8**	346K
EGAgent GPT-4.1 (EG+F+T)	1FPS$\rightarrow$50	F, C, T	50.7	+13.8	571K
EGAgent Gemini 2.5 Pro (EG+F+T)	1FPS$\rightarrow$50	F, C, T	57.5	+20.6	880K

Table 7: Recall@windowsize (recall@W) of agentic approaches on EgoLifeQA with respect to ground-truth timestamps provided by the dataset. We compute recall over temporal windows W centered on each ground-truth timestamp for the predicted timestamps from each tool: *i.e.*, EG for $Tool_{\text{eg}}$, F for $Tool_{\text{vis}}$, and T for $Tool_{aud}$. The number of timestamps that each search tool searches over is marked by Input #ts, and the number of timestamps highlighted by the analyzer tool as relevant to the query is marked by Selected #ts.

Category	Method	Input # ts	Selected # ts	recall@W					
				10 sec	30 sec	1 min	2 min	10 min	1 hour
MLLMs (Uniform Sampling)	Gemini 2.5 Pro	3000	3.1	0.101	0.160	0.192	0.238	0.325	0.410
Ours	EGAgent (F+T) Overall	4750	4.8	0.232	0.241	0.255	0.268	0.322	0.418
	EGAgent (EG+F+T) $\mathcal{M}_{EG}$	158	10.8	0.127	0.166	0.199	0.233	0.413	0.658
	EGAgent (EG+F+T) $\mathcal{M}_{VIS}$	50	17.6	0.857	0.868	0.873	0.875	0.900	0.930
	EGAgent (EG+F+T) $\mathcal{M}_{AUD}$	4700	2.6	0.218	0.247	0.261	0.288	0.347	0.417
	EGAgent (EG+F+T) Overall	4958	31.0	0.884	0.895	0.898	0.902	0.932	0.962

As seen in Tab. 7, the visual search tool shows strong recall even at window size of 10 seconds, indicating that it shows strong temporal localization capabilities. It is natural to question why our MCQ accuracy remains relatively low (34.6% when using only $Tool_{\text{vis}}$, Tab. 5) even with such high recall of $Tool_{\text{vis}}$; we highlight that even with perfect precision (using an audio-visual oracle, Sec. D.2), the MCQ accuracy saturates at 68.7%. This indicates that an audio-visual analysis of ground-truth timestamps alone is insufficient to push the frontier further.

The audio transcript search tool shows poor recall at small window sizes, which is surprising as an oracle with audio transcript search is 21% better than the previous state-of-the-art (Gemini 2.5 Oracle with T in Tab. 6). When examining $\mathcal{M}_{\text{aud}}$ we observe that this is because while the analyzer tool points out relevant context from audio transcripts for each task from the planning agent, it occasionally misses explicitly pointing out timestamps on which the timestamp occurs is ambiguous. This leads to missing hits even when the analyzer has analyzed the correct portion of the audio transcript (as is evident from the search tool's analysis in the working memory $\mathcal{M}$).

The entity graph search tool shows the worst fine-grained temporal localization of all EGAgent search tools at smaller window sizes ($\leq$ 2 min) which is expected as it is a lower-dimensional projection of the audio-visual space when compared to visual embeddings in $\mathbb{R}^d$ generated by a vision encoder (SigLIP 2) or raw audio transcripts. We observe that the entity graph starts to beat the recall@W of the audio transcript search at windows $>$ 2 minutes, indicating its broader temporal coverage compared to the audio transcript search. Since searching the entity graph is 3.5$\times$ faster than audio transcript search (Tab. 8), the entity graph search provides a flexible recall-latency tradeoff and is valuable to our EGAgent for both coarse temporal shortlisting (high recall at large window size) and fine-grained cross-modal reasoning with the visual and audio transcript search tools. See Figure 1 and

Table 8: An expanded version of Table 4 showing wall-clock latency in seconds of each module within EGAgent averaged over all MCQ on EgoLifeQA. For both the Visual and Transcript searches, the wall-clock time of the analyzer tool (a multimodal LLM) dominates the retrieval time. When the transcript search backbone is an LLM, both the retrieval and analysis happen simultaneously.

Method	Acc(%)	Transcript Search Backbone	Wall-Clock Runtime (sec)								#Tokens
			Planning	Visual Search		EG Search	Transcript Search		VQA Agent	Total	
				Retriever	Analyzer		Retriever	Analyzer			
EGAgent GPT-4.1	43.9	BM25	3.1	4.6	41.1	8.4	1.7	8.2	6.9	125	172K
(EG + F + T)	50.7	LLM	3.1	4.5	41.8	10.2	–	35.4	6.9	169	571K

Fig. 5 for examples.

Lastly, when all our tools are combined to form EGAgent (EG + F + T), we observe very strong recall of 0.88 even at a window size of 10 seconds. This result provides evidence that the strong performance of EGAgent on EgoLifeQA (Table 1) can be attributed to higher quality temporal localization of context relevant to the original query about the very long video.

D.4 EGAgent Latency and Memory

Latency. We expand Tab. 4 (main paper) to show latency of each component of EGAgent in Tab. 8 in terms of wall clock time. We observe that the latency of the MLLM analyzer tool dominates for both the visual search and audio transcript search-compared to the actual retrieval time. Notably, in the case of visual search, the analyzer MLLM must process 50 retrieved frames, contributing significantly to the latency ($\sim 9.1\times$ higher than the actual retrieval time). When switching the backbone of audio transcript retrieval from a MLLM to BM25, the latency of overall audio transcript search is $3.6\times$ lower. This analysis shows that our entity graph search tool adds minimal inference overhead to standard audio-visual search setups (12.8% on average) while providing strong accuracy gains, especially in tasks requiring knowledge of inter-entity relationships (Tab. 5).

Memory. The entity graph queried by EGAgent's Entity Graph Search Tool ($Tool_{\text{eg}}$) is extremely lightweight. For $\sim$52 hours of EgoLife video recorded by participant Jake, the SQLite database occupies only $\sim$2 MB on disk, and for Video-MME (Long), $\sim$65 KB on average per video. On the other hand, the visual embedding database used by the Visual Search Tool ($Tool_{\text{vis}}$) is much larger as it stores embeddings of video frames sampled at 1 frame-per-second (FPS). On Egolife (Jake), this corresponds to 187011 frames ($\sim$51.94 hours) and 1.4 GB memory on disk. On Video-MME (Long), this corresponds to a total of 739896 frames ($\sim$205.52 hours) and 5.7 GB memory on disk, or 19 MB per video. The Audio Transcript Search Tool ($Tool_{\text{aud}}$) queries raw audio transcripts (.srt files provided by the original dataset creators), which total 5.2 MB memory on disk on EgoLife (Jake). Video-MME provides transcripts for only 744 / 900 total videos and 292 / 300 videos in the Long subset. The Long subset transcripts which EGAgent queries occupies a total of 24.2 MB or $\sim$83 KB per video.

D.5 Entity Graph Noise

To address possible ASR or extraction errors affecting the quality of our entity graph relationships, we utilize a strict-to-relaxed search strategy where the agent automatically broadens temporal windows to maximize recall if initial matches fail (Section 3.3). Additionally, we randomly sample 100 relationships from EgoLife (out of 13968) and manually audit if the relationship is correct by examining the raw video and audio transcripts. On this subset, we find a **94% accuracy rate**, where the failures generally correspond to subtle errors in visual captioning or while fusing the captions and transcripts.

E Failure Cases

We show example mistakes made by EGAgent (Gemini 2.5 Pro) on EgoLifeQA in Fig. 8.

(1) **Identity Attribution under Perceptual Ambiguity.** As shown in Fig. 8a, EGAgent successfully localizes the relevant moment but fails to reliably identify the individual in the frame. This highlights the challenges in multimodal identity grounding; the agent struggles to disambiguate individuals when visual signals are degraded and confounded by audio, misaligning identity with concurrent speech rather than visual context. Incorporating persistent persona representations (e.g., appearance cues like hair color or body shape) could improve robustness.

(2) **Conflicts Between Audio and Visual Evidence.** Fig. 8b illustrates a failure case stemming from incorrect temporal reasoning due to conflicts between audio and visual evidence. EGAgent inconsistently reconciles temporal evidence across modalities, sometimes privileging prior verbal intent over more reliable visual observations. Enforcing cross-modal consistency checks and prioritizing direct evidence could mitigate this issue.

(3) **Entity Graph Incompleteness.** In the example in Fig. 8c, social or relational information is underrepresented relative to object interactions in the graph, which EGAgent cannot reconcile. Richer representations of agents and interactions, along with more targeted query planning to explicitly target missing relational attributes, may address this limitation.

(4) **Ambiguity in the Question.** As shown in Fig. 8d, when asked to infer Katrina's food preferences, EGAgent selects pizza, supported by multiple observations of her consuming pizza over the course of the week. However, during a shared meal, she remarks "this is delicious" in the context of Tremella soup being discussed, introducing ambiguity as to whether the statement refers specifically to the soup or to another dish at the table. Both pizza (frequent consumption) and Tremella soup (positive verbal feedback) are plausible answers.

F Implementation Details

For GPT-4.1, GPT-4o, and Gemini 2.5 Pro, we use the default settings with a temperature of 0 and a maximum of 3 retries. For Qwen-2.5-VL-7B (see Table 2 in the main paper), the model is hosted locally using vLLM on 4×H200 GPUs, with temperature set to 0, `tensor-parallel-size` = 4, and `gpu-memory-utilization` = 0.85.

Agent Implementation. We use LangGraph (LangChain, 2025) for implementing our EGAgent. We use AI assistants to help write code for our agent implementation. We first convert our multiple-choice question into a StateGraph called VeryLongVideoQA which contains all necessary attributes for EGAgent inference. We show code for our agent design in App. F. Note that all accuracies reported in this work are from a single run, as running agents multiple times on each dataset is computationally prohibitive.

We construct our EGAgent as shown in Figure 3 over the VeryLongVideoQA StateGraph. Once EGAgent receives a query Q, our planning agent ("planner" node) comes up with a sequence of N sub-tasks, which are saved to `VeryLongVideoQA.plan`. A router ("route_plan") then sends sub-task S_i (`VeryLongVideoQA.current_task`) to appropriate tool T_i (visual, entity graph, or audio transcript search) along with search query arguments q_i. The retrieved content from these tools is passed to the analyzer tool ("analyzer") which updates the working memory $\mathcal{M}$ (`VeryLongVideoQA.working_memory`). We also use an early exit condition that checks if the working memory already contains answers for future sub-tasks ("grade_plan_completion"). If it does not, we return control to the planning agent and proceed with future sub-tasks. If the working memory answers all past and future sub-tasks, we jump straight to the VQA agent ("generate_answer") which predicts the final answer A (`VeryLongVideoQA.answer`).

```
from langgraph.graph import StateGraph
from typing_extensions import TypedDict
from typing import List

# defines graph attributes
class VeryLongVideoQA(TypedDict):
    """
    Attributes:
    question: multiple-choice question
    candidates: four options for MCQ
    selected_video: selected video name
    start_t: when to begin tool search
    end_t: when to end tool search
    query_time: the time (and day) the
      query is asked, if provided
    audio_transcripts: full audio
      transcripts of long video
    plan: decompose the question into
      multi-step plan
    working_memory: accumulate cross-
      modal evidence
    current_task: current planner task
      being executed
    previous_tasks: planner tasks
      previously completed
    answer: VQA agent predicted answer
    total_tokens: total tokens used
    """

    question: str
    candidates: List[str]
    selected_video: str
    start_t: int
    end_t: int
    video_duration: int
    query_time: str
    audio_transcripts: List[str]
    plan: List[str]
    working_memory: str
    current_task: str
    previous_tasks: List[str]
```

```
    answer: str
    total_tokens: List[str]
```

```
from langgraph.graph import START, END

wf = StateGraph(VeryLongVideoQA)

# Define the agent nodes
wf.add_node("planner", planner)
wf.add_node("search_eg", search_eg)
wf.add_node("search_visual",
  search_visual)
wf.add_node("search_tscripts",
  search_tscripts)
wf.add_node("generate_answer",
  generate_answer)


# Build agent graph
wf.add_edge(START, "planner")
wf.add_conditional_edges(
    "planner",
    route_plan,
    {
        "eg": "search_eg",
        "visual": "search_visual",
        "audio": "search_transcripts"
    },
)

wf.add_edge("search_eg", "analyzer")
wf.add_edge("search_visual", "analyzer")
wf.add_edge("search_transcripts",
  "analyzer")

# allow early termination if all tasks
# addressed by working memory
wf.add_conditional_edges(
    "analyzer",
    grade_plan_completion,
    {
        "complete": "generate_answer",
        "incomplete": "planner",
    },
)

# Send working memory and query to VQA
  Agent
wf.add_edge("generate_answer", END)
```

Entity Graph Extraction. To create an entity graph, we first need a good audio-visual scene representation to extract entities and relationships from. We create these scene representations by fusing (with GPT 4.1) audio transcripts and visual captions we generate via GPT-4.1 at 30 second intervals (see "System Prompt for Visual Caption - Transcript Fusion" below). These fused captions have cross-modal information, where people, objects, actions, and events are described by visual captions, and audio cues (+ speaker identities in the case of EgoLife) provide additional context to relationships that occur in the scene.

We use Langchain's LLMGraphTransformer to extract an initial candidate set of nodes and relationships from our generated fused captions. While temporal localization via search tools is very important for long video understanding (Sec. D.2), LLMGraphTransformer module does not support adding any additional metadata to graph nodes and edges. To later equip our search tool with temporal filtering capabilities, we annotate all extracted relationships (entity graph edges) with timestamps based on the audio transcripts and visual captions. See "User Prompt for Temporal Annotation of Entity Graph Edges" below for more details.

Entity Graph Extraction Design Choices. We initially experimented with free-form relation extraction, allowing the LLM to describe relationships from the fused transcripts and video captions without constraints. We found nearly all of these relationships collapsed into four major buckets, i.e., "interacts with" and "uses" for physical person-object interactions in specific locations and "talks to" or "mentions" for verbal interactions, which are the abstractions we used to extract the entity graph (`allowed_relationships` below). Increasing relation granularity primarily introduced synonymous or low-frequency relation types, which fragmented retrieval queries and drastically reduced entity-graph recall during agentic search (Sec. D.3). We found that coarser abstractions improved robustness by enabling broader matching across noisy ASR and caption signals while preserving the relational information needed for temporal localization and multi-hop reasoning. We also highlight that while these four abstractions are sufficient for the dense social and physical interactions in egocentric data like EgoLife, our entity graph extraction pipeline is entirely domain-agnostic. The relationship schema itself is prompt-defined (see "System Prompt for Entity Graph Extraction" below), making it straightforward and efficient to adapt to different domains (e.g., 30 minutes on ∼52 hours of EgoLife).

```
from langchain_core.documents import
  Document
from langchain_experimental.
  graph_transformers import
  LLMGraphTransformer

def generate_eg(text:str):
    llm = get_vision_llm('gpt-4.1')
    allowed_nodes = ["Person",
      "Location", "Object"]
    allowed_relationships = ["TALKS_TO",
      "INTERACTS_WITH", "MENTIONS",
      "USES"]
    docs = [Document(page_content=text)]
    eg = LLMGraphTransformer(
```

```
        llm,
        allowed_nodes,
        allowed_relationships
    )
    eg = eg.aconvert_to_graph_documents(
      docs)
    return eg

# fuse audio tscripts and visual caps
fused_caps = generate_fc(caps, tscripts)
relationships =  generate_eg(fused_caps)

eg_with_tstamp = temporal_annotator.
  invoke(
    {
        "relationships": relationships,
        "transcripts": tscripts,
        "captions": fused_caps
    }
  )
```

Token Usage Estimates. Here we provide details for estimates of total tokens used by baseline methods in Tab. 1 and Tab. 2. For GPT 4.1 and Gemini 2.5 Pro, we apply 85 and 258 tokens per image respectively as per their API documentation. For Video-RAG (Luo et al., 2025), we add 2K tokens used by auxiliary texts (reported in the original paper) to an estimated 258 tokens per image. For EgoGPT (Yang et al., 2025b), we roughly estimate the number of tokens for text summaries at intervals of 30 seconds ($\sim$100), one hour ($\sim$500), and one day ($\sim$2000). We assume one inference pass searches one day (2K tokens), 10 hours per day (5K tokens), and 120 30-second intervals per hour (12K tokens). For all other methods (Xue et al., 2025; Ma et al., 2025; Yuan et al., 2025), we assume they use the entire LLM context window.

(a) QID 18

(b) QID 231

(c) QID 353

(d) QID 470

Figure 8: Analyzing EGAgent mistakes on EgoLifeQA. The query, options are in the top left, with colors denoting the ground-truth answer and the agent's incorrect prediction, as well as a ground-truth labeled "reason" for the correct answer. The topmost row shows a 4 second window around the `target_time` for each query provided by the benchmark (ground-truth time for evidence). The two rows of images below the top show frames retrieved by the Visual Search Tool ($Tool_{\text{vis}}$). The box below the images shows the reasoning chain of the VQA Agent and the red box at the bottom summarizes why EGAgent made a mistake. QID for each example refers to the question ID from EgoLifeQA.

Planning Agent Prompt:

“““ You are an expert at answering questions about very long videos. These questions may require multi-hop reasoning. Your job is to come up with a multi-step plan of all possible information that may be needed to answer the question.

Each step of your plan will be routed to three search tools. The first search tool looks at transcripts with timestamps, the second looks at image frames sampled at 1 FPS, and the third looks at an entity scene graph extracted from the long video. All search tools will search for context relevant to the plan step.

Keep each step concisely framed, and do not add compiling information as the final step, as this will be done automatically. You may use up to five steps, but use as few as necessary to answer the question.”””

System Prompt for Temporal Annotation of Entity Graph Edges:

“You are a helpful assistant that adds timestamps to relationships between graph nodes. Output only valid JSON, no prose.”

User Prompt for Temporal Annotation of Entity Graph Edges:

“““You are given:

1) a list of relationships: each relationship has relationship_id, source_id node, target_id node, and relationship type.

2) a caption containing dialogue: the caption contains a timestamp (start_t $\rightarrow$ end_t) and visual information from the scene as well as information on spoken dialogue.

3) a list of transcripts [t1, t2, ...], each containing a ‘timestamp’ (start_t $\rightarrow$ end_t) and ‘text’ containing spoken dialogue.

For every single provided relationship, find all transcripts and captions that support it.

Relationships : {relationships} Captions: {captions} Transcripts: {transcripts}

Rules:

- First, try to use only timestamps already present in transcript utterances.

- If no supporting utterances exist, use the entire interval from the caption as start_t and end_t.”””

System Prompt for Visual Caption - Transcript Fusion:

“““You are an expert multimodal summarization model. You will be given two aligned inputs corresponding to the same short video segment (about 30 seconds):

1. Visual Caption - a detailed description of what is visible in the video.

2. Diarized Transcript - spoken dialogue transcribed with timestamps and speaker names.

Your task is to fuse these into a single, coherent, and natural paragraph that integrates both visual and spoken information. Follow these rules carefully:

* Focus on relevant spoken content (who says what and its meaning) and highlight visual content (location, people, actions, and objects).

* Preserve factual details but avoid repetition or speculation.

* Keep the fused caption written in neutral, descriptive tone.

* Output only the fused caption - no explanations or metadata.

User Prompt for Visual Caption - Transcript Fusion:

““““ Here are the inputs for this segment:

1. Visual Caption: {caption_text}

2. Diarized Transcript: {transcript_text}

Produce one fused caption that naturally combines both.”””””

System Prompt for Entity Graph Extraction:

“““ Knowledge Graph Instructions

1. Overview

You are a top-tier algorithm designed for extracting information in structured formats to build a knowledge graph. Try to capture as much information from the text as possible without sacrificing accuracy. Do not add any information that is not explicitly mentioned in the text.

- Nodes represent entities and concepts.

- The aim is to achieve simplicity and clarity in the knowledge graph, making it accessible for a vast audience.

2. Labeling Nodes

- Consistency: Ensure you use available types for node labels. Ensure you use basic or elementary types for node labels.

- For example, when you identify an entity representing a person, always label it as ’person’. Avoid using more specific terms like ’mathematician’ or ’scientist’.

- Node IDs: Never utilize integers as node IDs. Node IDs should be names or human-readable identifiers found in the text.

- Relationships represent connections between entities or concepts. Ensure consistency and generality in relationship types when constructing knowledge graphs. Instead of using specific and momentary types such as ’BECAME_PROFESSOR’, use more general and timeless relationship types like ’PROFESSOR’. Make sure to use general and timeless relationship types!

3. Coreference Resolution

- Maintain Entity Consistency: When extracting entities, it’s vital to ensure consistency. If an entity, such as John Doe, is mentioned multiple times in the text but is referred to by different names or pronouns (e.g., Joe, he), always use the most complete identifier for that entity throughout the knowledge graph. In this example, use John Doe as the entity ID. Remember, the knowledge graph should be coherent and easily understandable, so maintaining consistency in entity references is crucial.

4. Strict Compliance

Adhere to the rules strictly. Non-compliance will result in termination. """

User Prompt for Entity Graph Extraction:

“““ Based on the following example, extract entities and relations from the provided text. Use the following entity types, don’t use other entity that is not defined below:

ENTITY TYPES: {allowed_nodes}

Use the following relation types, don’t use other relation that is not defined below:

RELATION TYPES: {allowed_relationships} ”””

Entity Graph Search System Prompt:

“““ You are an expert SQL reasoning assistant working over a SQLite database ‘entity_graph_table’ with the following schema:

entity_graph_table(
id INTEGER PRIMARY KEY AUTOINCREMENT,
day INTEGER, # 1 to 7. Must be $\leq$ query time day
start_t INTEGER, # e.g., 132609 for 13:26:09. Should be earlier than end_t
end_t INTEGER, # e.g., 184016 for 18:40:16
transcript TEXT, # what was said between start_t and end_t
source_id TEXT, # name of the source entity e.g., Jake, Microwave, Yard
source_type TEXT, # (“Person”, “Location”, “Object”)
target_id TEXT, # name of the source entity e.g., Shure, Phone, Knife
target_type TEXT, # (“Person”, “Location”, “Object”)
rel_type TEXT # (“TALKS_TO”, “INTERACTS_WITH”, “MENTIONS”, “USES”)
)

This schema represents an entity graph extracted from long egocentric video (7 days, 8 hours a day). Each entry of the table represents a relationship in the entity graph: source_id (source_type) $\rightarrow$ rel_type $\rightarrow$ target_id (target_type) which occurs between time start_t and end_t on a particular day (from 1 to 7). e.g., Jake (Person) $\rightarrow$ USES $\rightarrow$ mobile phone (Object)

You are given a multiple-choice question about the long video and the time it is asked (e.g., day 6 at 15:23:41), as well as a specific goal designed by an expert planner. Your job is to construct SQL queries to query the above table to answer the specific goal given by the planner.

Rules for query generation:

1. Your goal is to find relevant rows describing relationships between entities.

2. You must construct SQL queries progressively, starting with the strictest filter and relaxing step by step if no results are found.

3. Each stage should keep only the necessary filters. The order of relaxation is:

(a) Strict: exact day, exact timestamp (start_t $\geq$x and end_t$\leq$y), exact source_id, exact target_id, exact rel_type.

(b) Relax time: same day, exact source_id/target_id, same rel_type.

(c) Relax day: all days, exact source_id/target_id, same rel_type. Day has to be $\leq$ to the query time day.

(d) Relax entity match: same rel_type but use substring (‘LIKE‘) for source/target_id. Try to use single word for both IDs here to maximize probability of substring match.

(e) Relax rel_type: search by entity only (no rel_type constraint).

4. Always return your reasoning, and the SQL for each step, in a structured format.

5. Do not hallucinate entity names; use = or LIKE matching only to suggest similar candidates.

6. Always use SELECT * FROM entity_graph_table WHERE ... Do not use SELECT transcript or any other specific element of the schema.

7. Do not search the transcript unless you have exhausted other options.

8. Keep relaxing until the last SQL query has ONLY target_id (and optionally transcript). ”””

Entity Graph Search User Prompt:

“““ User question: {question} asked at {query_time}, and relevant context gathered thus far: ‘{working_memory}’.

Your job is to create a SQL query to answer this specific goal given by an expert planner: ‘{current_task}’.

Return a JSON object with: - “reasoning”: a short summary of your search plan and why constraints are relaxed.

- “sql_queries”: an ordered list of candidate SQL strings to execute, from strictest to most relaxed.

You do not need to run SQL, only generate the statements.”””

Visual Search System Prompt:

“““You are a question re-writer that rewrites the input question into concise text queries optimized for text-image retrieval on frames from a very long video sampled at 1 FPS, and specifies when to search. You are also given relevant context from previous retrieval steps. Retrieval is carried out with SigLIP 2 embeddings, so keep the rewritten queries short (single word wherever possible), distinct, and unambiguous. Do not use generic common nouns or times of day (e.g., noon or afternoon) that are not specific to objects or actions present in the question and options, as these will likely return irrelevant search results using SigLIP 2 embeddings. Do not use specific named entities as text queries (such as names of non-famous people), as SigLIP 2 will not have seen these during training.

You are given the starting and ending time of the long video, and asked to select the day and a start time and end time to search between. If you are unsure when to search, search the entire duration (i.e., the entire day). You may search any day and time before the query time."""

Visual Search User Prompt: “““Here is the initial question: {current_task} asked at {query_time}

Relevant context from previous retrieval steps: {working_memory}

Here is a dictionary containing the start and end times of all days formatted as HHMMSS: {day_search_dict}.

Rules:

1. You may search any day between the start_t and end_t of that day.

2. Select a set of 1 to 3 concise text queries e.g., [’q1’] or [’q1’, ’q2’, ’q3’] for each day you would like to search, and optionally when during that day to search (between start_t and end_t).

3. If you only need one text query, only use one text query. Only use additional text queries if they are semantically distinct from one another.”””

Audio Transcript Search System Prompt:

“““You are a helpful assistant who analyzes how retrieved transcripts are relevant to answer a multiple-choice question about a long video. You are given a single step of a multi-step plan for answering the multiple-choice question and a list of transcripts (which may be diarized, i.e., have speaker names annotated) over the entire long video, where each list element is a dictionary of start time, end time, transcript text.”””

Audio Transcript Search System Prompt:

“““Your task is to select audio transcripts relevant to this step of the multi-step plan: {current_task}. You are also provided context from previous retrieval steps, as they may be relevant in your selection process: {working_memory}

Here are the full audio transcripts of the long video: {transcripts}.

Once you select all audio transcripts that may be relevant to answer the step, describe how the audio transcripts are relevant to the goal: {current_task}. Note that the question is from the first-person (egocentric) perspective of Jake. Any references to “me” or “I” thus refer to Jake.”””