TIMEGRAPH: TEMPORAL GRAPH MEMORY FOR LONG-HORIZON VIDEO QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon video question answering requires reasoning over information distributed across distant observations, where entities may persist while their states and relations change over time. Retrieving individual frames does not explicitly preserve these temporal dependencies, leaving entity continuity, temporal extent, and state changes to be reconstructed during answering. We propose *TimeGraph*, a temporal graph memory that organizes frame-level scene-graph observations into persistent entities, temporally bounded state and relation episodes, and explicit transition boundaries on a shared time axis. Given a question, TimeGraph uses entity and temporal constraints to retrieve a relevant temporal subgraph and convert its structure into answer-relevant evidence. We evaluate TimeGraph on NAVQA, NExT-QA, and OpenEQA, where it improves over previously reported results by 6.2, 3.1, and 1.9 points, respectively. Compared with frame-local lexical retrieval, temporal graph memory further improves performance by 35.25, 13.39, and 21.14 points across the three benchmarks. Further analysis shows that organizing visual evidence according to question demand can provide additional gains, although the effect varies across datasets and question types. These results demonstrate the value of establishing explicit temporal structure before question-time retrieval for long-horizon VideoQA.

1 INTRODUCTION

Video question answering over long videos requires reasoning over observations distributed across time. An entity may appear in multiple distant observations, an object may change its state, and a relation may hold only during part of a video. Consequently, answering a question may require determining whether observations refer to the same entity, when a state or relation holds, or how events change and unfold over time. These temporal dependencies arise across navigation, temporal and causal reasoning, and embodied understanding settings (Anwar et al., 2024; Xiao et al., 2021; Majumdar et al., 2024).

A common strategy for long-video question answering is to retrieve frames or short video segments relevant to the question and provide them to a vision-language model. While retrieval reduces the amount of visual input, it does not explicitly preserve the temporal structure connecting the retrieved observations. For example, a person may be observed holding a cup at one time and putting it on a table much later. The retrieved observations may contain both pieces of evidence, but they do not explicitly indicate whether they involve the same person and cup, when the holding relation starts or ends, or where the relation changes. The model must reconstruct these dependencies from independently retrieved local observations at answer time. As illustrated in Figure 1 (a), frame-level observations therefore provide local facts, but leave the temporal organization of these facts implicit.

A useful temporal memory should consequently preserve three complementary forms of structure. First, observations referring to the same physical entity should be associated across time, providing *persistent entity identity*. Second, states and relations should be represented with their temporal extent, indicating *when* they hold rather than only whether they occur. Third, changes between states or relations should be represented as explicit *transitions*, making temporal ordering and evolution directly accessible. These structures support different forms of question answering: entity continuity is important for counting and cross-observation reasoning, temporal extent for duration and interval questions, and transitions for temporal-order and state-change reasoning. The key challenge is

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

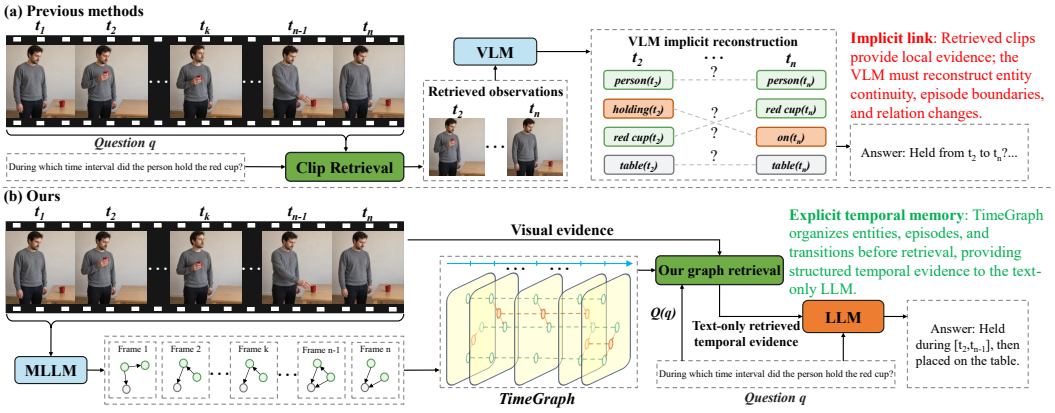


Figure 1: From frame-level observations to temporal graph memory for long-horizon video question answering. (a) Independent observations provide local visual evidence but do not explicitly preserve entity continuity or temporal changes. (b) TimeGraph associates observations along a shared time axis and represents persistent entities, temporally bounded episodes, and transitions, from which question-relevant temporal evidence is retrieved.

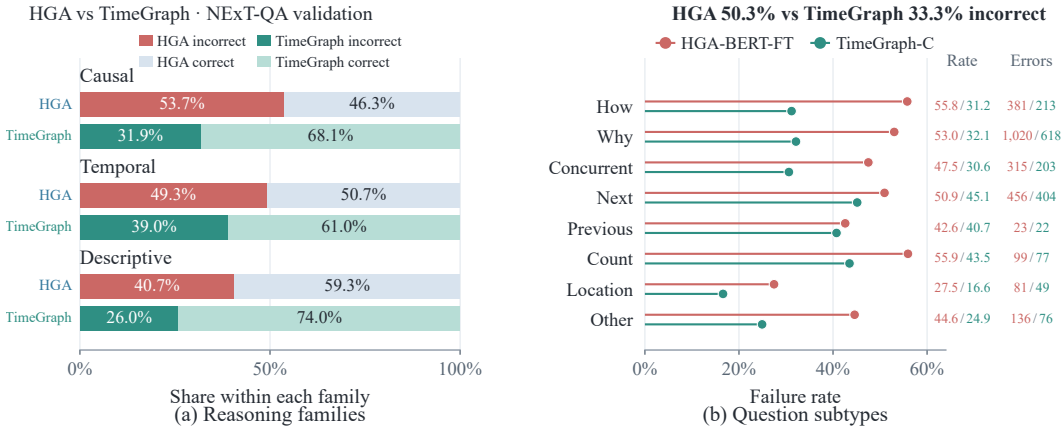


Figure 2: Failure-pattern comparison between the HGA-BERT-FT (Xiao et al., 2021) baseline and TimeGraph on the NExT-QA (Xiao et al., 2021) validation set. TimeGraph reduces the overall failure rate from 50.3% to 33.3%. (a) Correct and incorrect answer proportions across reasoning families. (b) Failure rates and error counts across question subtypes. TimeGraph consistently exhibits lower failure rates, with the largest reduction in causal reasoning.

therefore not simply to retrieve more observations, but to organize local observations into a memory that preserves *what* persists, *when* it holds, and *how* it changes.

This challenge is reflected in existing video question answering systems. Figure 2 analyzes the official HGA-BERT-FT (Xiao et al., 2021) baseline on the NExT-QA (Xiao et al., 2021) validation set and shows substantially different failure rates across reasoning types. Causal questions have a 53.7% error rate and temporal questions have a 49.3% error rate, compared with 40.7% for descriptive questions. Several subtypes that require reasoning over multiple observations, including *How*, *Why*, and *Count*, also exhibit relatively high failure rates. These statistics do not identify the cause of individual errors, but they indicate that reasoning over relationships between observations remains challenging, motivating representations that make such temporal dependencies explicit.

Scene graphs provide structured representations of objects, attributes, and relations within individual images or frames (Krishna et al., 2017; Ji et al., 2020), and graph-based methods have also been explored for video question answering (Xiao et al., 2022; 2023). Persistent visual memory further demonstrates the value of retaining structured information beyond individual observations (Anwar et al., 2024). However, frame-level scene graphs and stored observations do not by themselves establish a canonical temporal structure across observations. Observations referring to the same entity

must first be associated before identity continuity can be maintained; states and relations require explicit temporal extent; and changes require explicit transition boundaries. Existing representations therefore leave an important gap between *local visual observations* and a *queryable temporal memory* that explicitly captures entity continuity and temporal evolution.

We propose *TimeGraph*, a temporal graph memory for long-horizon video question answering. As illustrated in Figure 1 (b), TimeGraph first constructs a question-independent memory from frame-level hierarchical scene-graph observations (Liu et al., 2026). Observations referring to the same entity are associated along a shared time axis and consolidated into persistent entities. Consistent states and relations are represented as temporally bounded episodes, while changes between episodes are represented as explicit transitions. The resulting memory therefore provides a unified representation of *what persists, when it holds, and how it changes*. Given a question, TimeGraph converts it into a structured query containing target entities, temporal constraints, and the required temporal operation. These constraints are used to retrieve a relevant temporal subgraph from the shared memory, which subsequently constrains visual evidence retrieval. The retrieved graph structure and visual evidence are then projected into organized textual evidence for downstream question answering. Importantly, different questions can access different structures from the same memory: episode boundaries can support duration reasoning, associated entities can support counting, and ordered transitions can support temporal reasoning. Figure 2 shows the resulting difference in failure patterns on NExT-QA: the complete TimeGraph system reduces the overall failure rate from 50.3% to 33.3% and lowers the failure rate across the examined reasoning families and question subtypes.

We evaluate TimeGraph on NAVQA (Anwar et al., 2024), NExT-QA (Xiao et al., 2021), and OpenEQA (Majumdar et al., 2024) using a common answering protocol. TimeGraph improves over previously reported results by 6.2, 3.1, and 1.9 points on the three benchmarks, respectively. More importantly, controlled comparisons against frame-local retrieval show substantially larger gains when observations are organized through temporal graph memory, while further analysis examines how visual evidence organization interacts with different question demands. These results provide evidence that explicitly organizing temporal structure before question-time retrieval can make long-horizon video evidence more useful for downstream question answering. Our contributions are summarized as follows:

- We propose *TimeGraph*, a temporal graph memory that organizes frame-level scene-graph observations into three complementary forms of temporal structure: persistent entity identity, temporally bounded state and relation episodes, and explicit transition boundaries.
- We develop structure-aware temporal retrieval that uses question targets and temporal constraints to retrieve relevant subgraphs from the shared memory and convert them into answer-relevant evidence.
- We provide controlled evidence analysis across NAVQA, NExT-QA, and OpenEQA, showing that explicit temporal organization improves over frame-local retrieval and further examining how different question demands benefit from different forms of temporal evidence.

2 Related Work

Long-Horizon Video Question Answering. Video question answering increasingly requires reasoning over information distributed across long temporal contexts. NExT-QA targets causal, temporal, and descriptive reasoning (Xiao et al., 2021), while NAVQA and OpenEQA evaluate question answering over long-horizon navigation and embodied experiences (Anwar et al., 2024; Majumdar et al., 2024). Existing methods typically address long contexts by selecting relevant frames or segments, or by constructing intermediate representations for extended observations. These settings highlight that relevant evidence may be distributed across observations and depend on the continuity and temporal evolution of entities. TimeGraph addresses this challenge by consolidating frame-level observations into a persistent temporal structure before question-time retrieval.

Persistent Visual Memory and Video Graphs. Persistent visual memory and graph-based representations provide structured alternatives to frame-level processing. ReMEmbR stores and retrieves observation histories for long-horizon navigation (Anwar et al., 2024), while ConceptGraphs, VL-KnG, and DAAAM organize visual observations into object-centric or spatiotemporal memories (Gu

et al., 2024; Al Mdfaa et al., 2025; Gorlo et al., 2025). Visual Genome and Action Genome, together with VGT and CoVGT, further demonstrate the use of graph-structured object and relation representations for visual understanding and VideoQA (Krishna et al., 2017; Ji et al., 2020; Xiao et al., 2022; 2023). HSGG provides hierarchical frame-level scene-graph observations for open-world video understanding (Liu et al., 2026). However, these representations do not explicitly establish a shared temporal structure that associates observations of persistent entities and represents their changing states and relations. TimeGraph addresses this gap by organizing observations along a common time axis into persistent entities, temporally bounded episodes, and transition boundaries.

Retrieval-Augmented Video Understanding. Retrieval-augmented methods reduce the context required for downstream reasoning by selecting relevant information from a larger observation space (Lewis et al., 2020). In video understanding, similarity-based retrieval can efficiently identify relevant frames or segments, but independently retrieved observations do not explicitly encode entity continuity or temporal evolution. TimeGraph instead uses persistent entities and temporal relations as retrieval anchors, constraining evidence retrieval to temporally connected observations. The resulting temporal structure can then be projected into question-dependent evidence while the underlying memory remains unchanged.

3 METHOD

3.1 OVERVIEW

We propose *TimeGraph*, a temporal graph memory for long-horizon video question answering. The central idea is to transform frame-level visual observations into a persistent temporal structure before question answering. Rather than treating retrieved frames as independent evidence units, TimeGraph explicitly represents entity continuity, temporally bounded states and relations, and changes between them. Given a video, we first obtain frame-wise hierarchical scene graphs and associate observations referring to the same entity along a shared time axis. The associated observations are then consolidated into persistent entities, temporal episodes, and transition boundaries, producing a question-independent temporal memory. At inference time, a question is converted into a structured query containing its target, constraints, and temporal operation. The query first retrieves relevant structures from the temporal memory, which subsequently constrain visual evidence retrieval. Finally, the retrieved graph and visual evidence are projected into a question-appropriate textual representation and passed to a downstream question answering model. Figure 3 summarizes the complete pipeline. The key distinction from frame-local retrieval is that TimeGraph retrieves evidence through a persistent temporal structure rather than directly selecting isolated frame-level observations. Entity identity, temporal extent, and state or relation changes are established during memory construction and can therefore be reused across different questions.

3.2 PROBLEM FORMULATION

Let a video be represented by a sequence of frame-level observations $\mathcal{V} = \{(\tau_t, \mathcal{H}_t)\}_{t=1}^{N}$, where N is the number of sampled observations, τ_t is the temporal coordinate of the t-th observation, and $\mathcal{H}_t$ is the corresponding hierarchical scene graph: $\mathcal{H}_t = (\mathcal{O}_t, \mathcal{A}_t, \mathcal{R}_t)$. Here, $\mathcal{O}_t$ denotes the set of observed objects, $\mathcal{A}_t$ denotes their attributes or states, and $\mathcal{R}_t$ denotes the relations between objects. We obtain $\mathcal{H}_t$ using an off-the-shelf hierarchical scene graph generator (Liu et al., 2026).

Given a task-formatted question q, the goal is to predict its answer y. We separate the process into two stages. The first stage constructs a question-independent temporal memory: $\mathcal{M} = \mathcal{F}(\mathcal{V})$, where $\mathcal{F}$ denotes the temporal memory construction process. The memory $\mathcal{M}$ is constructed once for each video and is independent of the question. The second stage performs question-conditioned retrieval and evidence projection. Let $\mathcal{G}_q = \mathcal{R}_G(\mathcal{M}, q)$ denote the temporal subgraph retrieved for question q, and let $\mathcal{B}_q = \mathcal{R}_V(\mathcal{V}, \mathcal{G}_q, q)$ denote the corresponding set of visual evidence candidates. The retrieved graph and visual evidence are then projected into textual evidence: $\mathcal{Z}(q) = \mathcal{P}(\mathcal{G}_q, \mathcal{B}_q, q)$, and the final answer is generated as

$$\hat{y} = f_{\mathrm{QA}}\left(s_{\mathrm{sys}}, q, \mathcal{Z}(q)\right), \tag{1}$$

where s_{sys} is the system prompt and f_{QA} denotes the downstream question answering model.

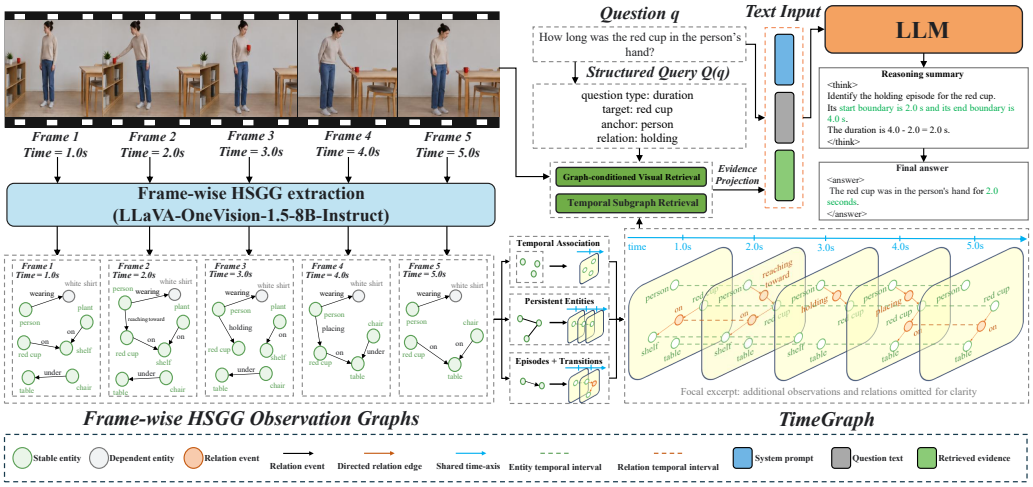


Figure 3: Overview of TimeGraph. Frame-wise hierarchical scene graphs are first associated along a shared time axis and consolidated into persistent entities, temporally bounded state and relation episodes, and transition boundaries. Given a question, a structured query identifies relevant entities and temporal constraints, which are used to retrieve a temporal subgraph and constrain visual evidence retrieval. The retrieved evidence is then projected into a question-appropriate textual representation for downstream question answering.

This formulation separates question-independent temporal memory construction from question-dependent evidence retrieval and organization. The same temporal memory can therefore support different questions while exposing different temporal structures according to their information requirements.

3.3 Temporal Graph Memory Construction

Temporal Association. Frame-level scene graphs provide local observations of objects and their relations, but the same physical entity may be observed repeatedly at different temporal coordinates. To preserve such entity continuity, we associate object observations along the temporal axis and use the resulting links to construct persistent entities. Let o and o' denote two object observations at temporal coordinates τ and τ', respectively. We measure their temporal consistency using an association score

$$S_{\text{assoc}}(o, o') = \sum_{k=1}^{K} \lambda_k \phi_k(o, o'), \tag{2}$$

where $\phi_k(o, o')$ denotes the k-th consistency cue and λ_k is its corresponding weight. The cues capture appearance, geometric, motion, area, and local detection-order consistency. All cues are computed solely from frame-level observations and are independent of the question; their detailed definitions are provided in Appendix Sec. .4. We then establish a temporal link between two observations when they are sufficiently close in time and exhibit sufficient consistency:

$$\text{link}(o, o') = \mathbf{1}\left[|\tau - \tau'| \le w \ \wedge \ S_{\text{assoc}}(o, o') \ge \eta\right], \tag{3}$$

where w denotes the temporal association window, η is the association threshold, and $\mathbf{1}[\cdot]$ is the indicator function. The temporal window limits associations between observations that are too far apart to be reliably matched, while the consistency score determines whether observations within the window are sufficiently compatible. Applying this linking process over the temporal axis yields chains of associated observations, which are treated as candidate persistent entities for subsequent temporal state and relation modeling.

Persistent Entities and Temporal Episodes. Let $\mathcal{C}(o)$ denote the temporal chain containing observation o. Observations belonging to the same chain are consolidated into a persistent entity: $u = (\mathcal{I}_u, \Omega_u)$, where $\mathcal{I}_u$ denotes the identity representation of entity u, and Ω_u denotes its supporting observations. A persistent entity can therefore be referenced across multiple temporal locations

rather than being tied to an individual frame. For each persistent entity or entity pair, observations with a consistent attribute, state, or relation are further consolidated into a temporal episode:

$$h = (u, \kappa, [\tau_s, \tau_e], \Omega_h)\,, \tag{4}$$

where u denotes either a persistent entity or an entity pair, κ denotes an attribute, state, or relation, $[\tau_s, \tau_e]$ denotes its temporal extent, and Ω_h denotes the supporting observations. This representation explicitly separates *what* is observed from *when* it holds. For example, multiple observations that a person is holding a cup are consolidated into one holding episode associated with the person and cup, together with its temporal extent, rather than being stored as independent frame-level facts.

Transition Representation. Temporal episodes describe states or relations that persist over an interval, but long-horizon questions may also depend on changes between them. We therefore explicitly represent a change between two consecutive episodes as a transition: $\delta = (\kappa^- \to \kappa^+, \tau_b, \Omega_\delta)$, where κ^- and κ^+ denote the states or relations before and after the change, τ_b denotes the transition boundary, and Ω_δ denotes the observations supporting the change. For a persistent entity or entity pair u, a sequence of episodes and transitions can therefore be represented as

$$u : h_1 \xrightarrow{\delta_1} h_2 \xrightarrow{\delta_2} \cdots \xrightarrow{\delta_m} h_{m+1}. \tag{5}$$

This temporal organization makes entity continuity, state duration, relation changes, and event ordering explicit. We define the resulting TimeGraph memory as

$$\mathcal{M} = (\mathbb{U}, \mathbb{H}, \mathbb{D}, \mathbb{T})\,, \tag{6}$$

where $\mathbb{U}$ denotes the set of persistent entities, $\mathbb{H}$ denotes the set of temporal episodes, $\mathbb{D}$ denotes the set of transitions, and $\mathbb{T}$ denotes the shared temporal axis. Unlike a collection of frame-level scene graphs, $\mathcal{M}$ provides a persistent temporal representation in which observations referring to the same entity, state, or relation can be accessed through their shared structural context.

3.4 Question-Conditioned Temporal Retrieval

Structured Question Query. Given a task-formatted question q, we convert it into a structured query

$$Q(q) = (T_q, C_q, O_q)\,, \tag{7}$$

where T_q denotes the target entity or event, C_q denotes the relevant constraints or anchors, and O_q denotes the temporal operation required by the question. The temporal operation specifies the type of temporal reasoning required by the question. Examples include identifying an interval, determining temporal order, locating a previous or subsequent event, counting entities satisfying a condition, and identifying a state or relation change. Answer choices are included in q only when required by the corresponding benchmark. The structured query serves only as an interface to the temporal memory. It does not modify the memory or introduce an additional learned memory component. In particular, the temporal structure of the video is established independently of the question during memory construction.

Temporal Subgraph Retrieval. The structured query first identifies relevant structures from the temporal memory:

$$B_G(q) = R_G\left(\mathcal{M}, Q(q)\right), \tag{8}$$

where R_G denotes the temporal graph retrieval function and $B_G(q)$ denotes the retrieved graph anchors. The retrieval process uses the target and constraint anchors to identify relevant persistent entities, episodes, transitions, and their temporal neighborhoods. The retrieved elements are then assembled into a temporal subgraph:

$$\mathcal{G}_q = \mathrm{Subgraph}\left(\mathcal{M}; B_G(q)\right). \tag{9}$$

The resulting subgraph provides two complementary types of information. First, persistent entities establish canonical identities across multiple observations, preventing independently retrieved frames from being treated as unrelated entities. Second, temporal episodes and transitions provide explicit temporal anchors, such as episode boundaries and transition times, which constrain subsequent visual retrieval. Thus, the retrieval unit is a temporally connected subgraph rather than an isolated frame.

Graph-Conditioned Visual Retrieval. The temporal graph provides structured information about persistent entities, states, relations, and their temporal organization, but some questions require visual details that are not fully represented by these graph elements. We therefore use the retrieved temporal subgraph to constrain visual evidence retrieval:

$$B_V(q) = R_V\left(\mathcal{V}, \mathcal{G}_q, Q(q)\right), \tag{10}$$

where R_V denotes the graph-conditioned visual retrieval function and $B_V(q)$ denotes the retrieved visual evidence candidates. The candidate observations are associated with entities, episodes, relations, or transition boundaries already identified in $\mathcal{G}_q$. This differs from frame-local retrieval, where visual candidates are selected directly from the frame collection without an explicit persistent temporal structure. The resulting evidence therefore combines structural anchors from the temporal graph with visual observations that provide appearance details not explicitly encoded in the graph.

Evidence Projection. The retrieved temporal subgraph and visual candidates must be converted into a compact representation that can be consumed by the downstream QA model. We refer to this process as *evidence projection*. The projection determines which structural elements of the retrieved evidence should be expressed for a given question. Different question demands require different forms of evidence. For example, a duration question requires the boundaries of a relevant episode, a counting question requires a deduplicated set of entities satisfying a condition, and a temporal-order question requires an ordered sequence of relevant events or transitions. We therefore define the general evidence projection as

$$\mathcal{Z}(q) = P\left(\mathcal{G}_q, B_V(q), Q(q)\right), \tag{11}$$

where P denotes the evidence projection function that converts the retrieved graph and visual evidence into a textual representation. We consider two evidence projection strategies. The first is a *shared* strategy, which applies the same evidence organization across question types:

$$\mathcal{Z}_{\text{shared}}(q) = P_{\text{shared}}\left(\mathcal{G}_q, B_V(q)\right). \tag{12}$$

The second is a *demand-aware* strategy, which organizes the retrieved evidence according to the temporal operation O_q in the structured query:

$$\mathcal{Z}_{\text{demand}}(q) = P_{O_q}\left(\mathcal{G}_q, B_V(q)\right). \tag{13}$$

The two strategies use the same temporal memory and the same retrieved evidence. Their difference lies only in how the evidence is organized for downstream answering. This separation allows us to independently examine the benefit of preserving temporal structure and the benefit of matching evidence organization to question demand.

Answering with Structured Evidence. The final QA model (1) receives the task-formatted question and the textual evidence produced by the projection module, it does not directly receive the raw video frames. Visual models are used upstream to construct frame-level scene graphs and retrieve supporting visual observations, while TimeGraph provides the persistent temporal structure that connects these observations. The complete inference pipeline can therefore be summarized as

$$\mathcal{V} \rightarrow \mathcal{M} \rightarrow \mathcal{G}_q \rightarrow B_V(q) \rightarrow \mathcal{Z}(q) \rightarrow \hat{y}. \tag{14}$$

The key design principle is that entity identity, temporal episodes, and state or relation changes are established before question-time retrieval. Consequently, answering operates on a question-relevant temporal structure rather than reconstructing long-range temporal dependencies from a collection of independently retrieved frames. The overall algorithm and detailed processes are provided in Appendix .1.

4 Experimental Results

4.1 Experimental Setup

Datasets. We evaluate *TimeGraph* on three long-horizon video and embodied question answering benchmarks that require reasoning over observations distributed across time. **NAVQA** (Anwar et al., 2024). NAVQA evaluates question answering over long-horizon navigation sequences. Following the evaluation setting used in our experiments, we use the 139 non-position questions from

Table 1: Performance comparison on the NaVQA and OpenEQA datasets. Acc. denotes accuracy (%), while Score denotes the normalized LLM-Judge score.

NaVQA		OpenEQA	
Method	Acc.	Method	Score
ConceptGraphs Gu et al. (2024)	29.9	GPT-4 + LLaVA-1.5 captions Ong & Jang (2025)	43.6
ReMEmbR (NVILA-Lite-2B) Anwar et al. (2024)	48.3	GPT-4 + ConceptGraphs Majumdar et al. (2024)	36.5
DAAAM (DAM-3B) Gorlo et al. (2025)	67.2	GPT-4V Majumdar et al. (2024)	49.6
ReMEmbR (NVILA-Lite-8B) Anwar et al. (2024)	60.7	MemoryEQA + GPT-4o Zhai et al. (2025)	36.8
VL-KnG (GR) Al Mdfaa et al. (2025)	32.4	VL-KnG (GR) Al Mdfaa et al. (2025)	50.7
VL-KnG (GER-G) Al Mdfaa et al. (2025)	66.2	3D-Mem (re-eval) Yang et al. (2024)	50.8
ReMEmbR (original) Anwar et al. (2024)	65.0	SplatReasoner Yu et al. (2026)	51.6
TimeGraph (ours)	**73.4**$_{(+6.2)}$	**TimeGraph (ours)**	**53.5**$_{(+1.9)}$

seven CODa navigation sequences and report accuracy under the official NAVQA evaluation protocol. **NExT-QA** (Xiao et al., 2021). NExT-QA focuses on causal and temporal reasoning over videos. We use 4,996 validation questions from 570 videos and report five-way exact-choice accuracy. **OpenEQA**(Majumdar et al., 2024). OpenEQA evaluates question answering from episodic embodied experiences. We use 1,636 EM-EQA questions from 152 episodes and report the normalized LLM-Judge score: $S_{\mathrm{EQA}} = 25\,(\mathrm{mean}(s) - 1)$, where s denotes the evaluator score. The resulting metric is a 0–100 answer-quality measure. The evaluation protocol and evaluator are kept fixed throughout our comparisons.

Implementation Details. Frame-level visual observations are generated using an HSGG extractor based on LLaVA-OneVision-1.5-8B-Instruct (Liu et al., 2026). The same observations are used across evidence settings whenever applicable, ensuring that comparisons differ only in subsequent evidence organization. The temporal graph memory is constructed independently of the question and reused for all questions from the same video or episode. At question time, a structured query is used to retrieve a temporal subgraph and constrain visual evidence retrieval. We use GPT-5.6-luna for question representation and answer generation across all three benchmarks, and also use it as the OpenEQA evaluator with temperature 0. The answering model is fixed within each comparison, and the final QA model receives only the task-formatted question and organized textual evidence, not raw video frames.

4.2 Comparison with Existing Methods

We compare TimeGraph with previously reported results on NAVQA, OpenEQA, and NExT-QA for contextual evaluation. As shown in Tables 1 and 2, TimeGraph achieves 73.4% accuracy on NaVQA, a score of 53.5 on OpenEQA, and 66.7% overall accuracy on NExT-QA, exceeding the best reported values included in our comparison by 6.2, 1.9, and 3.1 points, respectively. These results show consistent gains across the three datasets. Since the compared methods may differ in training data, model architecture, visual inputs, frame budgets, question subsets, and evaluation protocols, we treat these results as published-result comparisons rather than strictly controlled baselines.

4.3 Evidence Pipeline Analysis

We progressively introduce increasingly structured video evidence to examine the contribution of temporal graph memory and subsequent evidence organization. We consider five settings: *question only*, *frame-local retrieval*, *temporal graph memory*, *shared visual evidence*, and *demand-aware visual evidence*. The first setting provides only the task-formatted question. Frame-local retrieval selects HSGG observations by lexical overlap, while temporal graph memory retrieves question-relevant temporal structures from TimeGraph. Shared visual evidence further adds graph-conditioned

Table 2: Performance comparison on the NExT-QA dataset. C/T/D denote causal, temporal, and descriptive accuracy (%), All denotes overall accuracy.

Method	Training	C	T	D	All
HGA Xiao et al. (2021)	Supervised	46.3	50.7	59.3	49.7
MIS Chu et al. (2025)	Supervised	54.6	56.6	66.9	57.2
GF Chu et al. (2025)	Supervised	56.9	57.1	70.5	58.8
SeViT Chu et al. (2025)	Supervised	54.0	54.1	71.3	56.7
CoVGT Xiao et al. (2023)	Supervised	58.8	57.4	69.4	60.0
HiTeA Chu et al. (2025)	Hybrid	62.4	58.3	**75.6**	63.1
GP-VQA Li et al. (2025)	Hybrid	–	–	–	61.9
VFC (zero-shot) Chu et al. (2025)	Zero-shot	51.6	45.4	64.1	51.5
InternVideo Chu et al. (2025)	Zero-shot	43.4	48.0	65.1	49.1
AssistGPT Chu et al. (2025)	Zero-shot	60.0	51.4	67.3	58.4
SeViLA Chu et al. (2025)	Zero-shot	61.3	**61.5**	**75.6**	63.6
TimeGraph (ours)	Zero-shot	**68.1**	61.0	74.0	**66.7**$_{(+3.1)}$

Table 3: Performance of progressively structured evidence configurations. NAVQA and NExT-QA report accuracy (%), OpenEQA reports normalized LLM-Judge score.

Evidence configuration	NAVQA Acc. (%) ↑	NExT-QA Acc. (%) ↑	OpenEQA Score ↑
Question only	1.44	21.58	16.08
Frame-local retrieval	17.27	47.14	27.78
Temporal graph memory	52.52	60.53	48.92
Shared visual evidence	62.59	65.19	49.39
Demand-aware visual evidence	**73.38**	**66.73**	**53.50**

Table 4: Ablation of temporal structures in TimeGraph. All variants use the same frame-level HSGG observations and downstream QA model. NAVQA and NExT-QA report accuracy (%), OpenEQA reports normalized LLM-Judge score.

Memory representation	Entity continuity	Temporal episodes	Transitions	NAVQA Acc. (%) ↑	NExT-QA Acc. (%) ↑	OpenEQA Score ↑
Frame-level graph	×	×	×	17.27	47.14	27.78
+ Entity continuity	✓	×	×	24.46	53.04	37.20
+ Temporal episodes	✓	✓	×	33.81	59.65	43.59
TimeGraph	✓	✓	✓	**52.52**	**60.53**	**48.92**

visual candidates, and demand-aware visual evidence organizes the retrieved evidence according to the question demand.

Table 3 shows consistent gains as increasingly structured evidence is introduced. Compared with frame-local retrieval, temporal graph memory improves performance by 35.25 points on NAVQA, 13.39 points on NExT-QA, and 21.14 points on OpenEQA, demonstrating the benefit of preserving temporal structure before question-time retrieval. Adding graph-conditioned visual evidence provides further gains of 10.07, 4.66, and 0.47 points, respectively, while demand-aware evidence organization adds another 10.79, 1.54, and 4.11 points. These results indicate that temporal memory construction and question-dependent evidence organization provide complementary benefits. Since these comparisons involve complete evidence configurations, the differences should be interpreted as pipeline-level effects rather than strict one-component ablations.

4.4 Temporal Graph Component Ablation

We next examine the contribution of the temporal structures in TimeGraph. Starting from frame-level scene graphs, we progressively introduce entity continuity, temporal episodes, and transition representation. All variants use the same frame-level HSGG observations and downstream QA model; only the temporal organization of the observations is changed. Table 4 isolates the contribution of persistent temporal organization beyond frame-level graph representation. The intermediate variants separately examine entity continuity, temporal extent, and state or relation changes, while the comparison between the frame-level graph and the complete TimeGraph measures their combined contribution.

5 Conclusion

We presented *TimeGraph*, a temporal graph memory for long-horizon video question answering. TimeGraph organizes frame-level scene-graph observations into persistent entities, temporally bounded state and relation episodes, and explicit transitions, establishing temporal structure before question-time retrieval. The resulting memory provides a structured basis for retrieving and organizing evidence according to different temporal demands. Overall, our results show that explicitly preserving entity continuity and temporal changes is a useful way to support reasoning over long-horizon video observations.

6 Limitations

TimeGraph depends on the quality of frame-level scene-graph observations and temporal entity association. Errors in detection, recognition, or association can propagate to persistent entities, temporal episodes, and transitions, particularly under large appearance changes or long periods of disappearance. Improving robust long-range entity association and reducing error propagation are important directions for future work.

AI USE STATEMENT

In this work, we used Large Language Models (LLMs) to improve the clarity, grammar, and readability of the manuscript. After using these tools, we carefully reviewed and edited the content as needed and take full responsibility for the final content of this work.

ETHICS STATEMENT

This work focuses on improving long-horizon video question answering by explicitly organizing visual observations into a temporal graph memory. We do not collect new data or conduct experiments involving human participants. All experiments are conducted on existing video understanding benchmarks and publicly released pretrained models. Our method does not introduce additional data collection or alter the intended use of the underlying datasets and models. We report results at the aggregate benchmark level and do not perform analyses involving sensitive personal attributes. We encourage users to follow the licenses, usage conditions, and ethical guidelines associated with the original datasets and models.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of TimeGraph, including frame-level scene-graph construction, temporal entity association, temporal episode and transition modeling, and question-conditioned temporal subgraph retrieval, in Section 3. The experimental setup, benchmark datasets, answering protocol, compared methods, and implementation details are described in Section 4. We further provide component ablations, comparisons with frame-local retrieval and previously reported results, and analyses of question-dependent evidence retrieval in the main paper and appendix. These details are intended to facilitate reproduction and further evaluation of our results. The complete anonymous codebase, including the implementation of TimeGraph and the scripts used for evaluation, is provided in the supplementary material.

REFERENCES

Mohamad Al Mdfaa, Svetlana Lukina, Timur Akhtyamov, Arthur Nigmatzyanov, Dmitrii Nalberskii, Sergey Zagoruyko, and Gonzalo Ferrer. Spatiotemporal knowledge graphs as persistent scene memory for embodied question answering. *arXiv preprint arXiv:2510.01483*, 2025. doi: 10.48550/arXiv.2510.01483. URL https://arxiv.org/abs/2510.01483.

Abrar Anwar, John Welsh, Joydeep Biswas, Soha Pouya, and Yan Chang. ReMEmbR: Building and reasoning over long-horizon spatio-temporal memory for robot navigation. *arXiv preprint arXiv:2409.13682*, 2024.

Chu, Li, and Chua. Understanding long videos via llm-powered entity relation graphs. *arXiv preprint arXiv:2501.15953*, 2025. URL https://arxiv.org/abs/2501.15953.

Nicolas Gorlo, Lukas Schmid, and Luca Carlone. Describe anything anywhere at any moment. *arXiv preprint arXiv:2512.00565*, 2025. doi: 10.48550/arXiv.2512.00565. URL https://arxiv.org/abs/2512.00565.

Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 5021–5028, 2024. URL https://arxiv.org/abs/2309.16650.

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

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei.

Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, 2020.

Yiming Li, Xiaoshan Yang, Bing-Kun Bao, and Changsheng Xu. Graph prompts: Adapting video graph for video question answering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1485–1493, 2025. URL https://www.ijcai.org/proceedings/2025/0166.pdf.

Yunzhe Liu, Wenbiao Liu, Lihui Cen, Zhe Qu, Yigang Cen, Yixiong Liang, Yongfang Xie, and Shichao Kan. HSGG: Training-free hierarchical scene graph generation with geometry-guided relation reasoning. In *Proceedings of the International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/65975.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul McVay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent-Pierre Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Alexander Sax, and Aravind Rajeswaran. OpenEQA: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Eugene Ong and Young Jang. R-EQA: Retrieval-augmented generation for embodied question answering. In *CVPR 2025 Workshop on Embodied AI*, 2025. URL https://embodied-ai.org/papers/2025/6_R_EQA_Retrieval_Augmented_Ge.pdf.

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

Junbin Xiao, Pan Zhou, Tat-Seng Chua, and Shuicheng Yan. Video graph transformer for video question answering. In *Proceedings of the European Conference on Computer Vision*, 2022.

Junbin Xiao, Pan Zhou, Angela Yao, Yicong Li, Richang Hong, Shuicheng Yan, and Tat-Seng Chua. Contrastive video question answering via video graph transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Zhaoyang Yang et al. 3D-Mem: 3d scene memory for embodied exploration and reasoning. *arXiv preprint arXiv:2411.17735*, 2024. URL https://arxiv.org/abs/2411.17735.

Yu et al. Splatreasoner: Enhancing embodied reasoning and grounding by novel view synthesis. *arXiv preprint arXiv:2601.13132*, 2026. URL https://arxiv.org/abs/2601.13132.

Mingliang Zhai, Zhi Gao, Yuwei Wu, and Yunde Jia. Memory-centric embodied question answer. *arXiv preprint arXiv:2505.13948*, 2025. URL https://arxiv.org/abs/2505.13948.

Algorithm 1 TimeGraph for Long-Horizon Video Question Answering

Require: Frame-level observations $\mathcal{V} = \{(\tau_t, \mathcal{H}_t)\}_{t=1}^{N}$, question q
Ensure: Answer $\hat{y}$
Temporal memory construction
Compute pairwise temporal association scores between object observations within window w
Link observations satisfying the temporal and association constraints to form temporal chains
Consolidate each chain into a persistent entity $u = (\mathcal{I}_u, \Omega_u)$
Group consistent attributes, states, and relations into temporal episodes $h = (u, \kappa, [\tau_s, \tau_e], \Omega_h)$
Identify changes between consecutive episodes as transitions $\delta = (\kappa^- \to \kappa^+, \tau_b, \Omega_\delta)$
Construct $\mathcal{M} = (\mathbb{U}, \mathbb{H}, \mathbb{D}, \mathbb{T})$
Question-conditioned retrieval
Parse q into structured query $Q(q) = (T_q, C_q, O_q)$
Retrieve graph anchors $B_G \leftarrow R_G(\mathcal{M}, Q(q))$
Construct temporal subgraph $\mathcal{G}_q \leftarrow \mathrm{Subgraph}(\mathcal{M}; B_G)$
Retrieve visual candidates constrained by $\mathcal{G}_q$: $B_V \leftarrow R_V(\mathcal{V}, \mathcal{G}_q, Q(q))$
Evidence projection and answering
Construct question-dependent evidence $\mathcal{Z}(q) \leftarrow P(\mathcal{G}_q, B_V, Q(q))$
Generate answer $\hat{y} \leftarrow f_{\mathrm{QA}}(s_{\mathrm{sys}}, q, \mathcal{Z}(q))$
return $\hat{y}$

APPENDIX

.1 THE ALGORITHM OF TIMEGRAPH AND DETAILED PROCESSES

Algorithm 1 summarizes the TimeGraph pipeline. Given frame-level scene-graph observations, we first construct a question-independent temporal memory by establishing entity continuity and consolidating states, relations, and their changes over time. Given a question, we then retrieve a question-relevant temporal subgraph, use it to constrain visual evidence retrieval, and project the resulting graph and visual evidence into a textual representation for answer generation. The key design is to establish temporal structure before question-time retrieval, so that long-range dependencies can be retrieved through persistent entities and temporal episodes rather than reconstructed from isolated frames. The detailed processes are as follows.

Operational notation. Let

$$\mathcal{V} = \{(\tau_t, \mathcal{H}_t)\}_{t=1}^{N}$$

denote the frame-level scene-graph observations of a video, where N is the number of sampled frames, τ_t is the timestamp of frame t, and $\mathcal{H}_t$ is the corresponding hierarchical scene graph. Each observation contains detected entities together with their attributes, states, and relations, as well as the visual observations supporting these records. The temporal memory constructed from $\mathcal{V}$ is denoted by

$$\mathcal{M} = (\mathbb{U}, \mathbb{H}, \mathbb{D}, \mathbb{T}),$$

where $\mathbb{U}$ denotes persistent entities, $\mathbb{H}$ denotes temporal state and relation episodes, $\mathbb{D}$ denotes transitions between episodes, and $\mathbb{T}$ denotes the shared temporal axis used to index observations and episodes. The structural edges induced by the temporal memory are collected in

$$E_{\mathcal{M}} \subseteq (\mathbb{U} \cup \mathbb{H} \cup \mathbb{D}) \times (\mathbb{U} \cup \mathbb{H} \cup \mathbb{D}).$$

Each graph record available for retrieval is represented as

$$x = (\mathrm{id}_x, \mathrm{kind}_x, \mathcal{E}_x, [\tau_x^s, \tau_x^e], \mathrm{text}_x, \Omega_x),$$

where id_x is the record identifier, kind_x specifies the record type, $\mathcal{E}_x$ is the set of associated entities, $[\tau_x^s, \tau_x^e]$ denotes its temporal support, text_x is its textual description, and Ω_x contains the supporting frame-level observations. A persistent entity is obtained by linking observations that refer to the same underlying entity across frames. For an entity $u \in \mathbb{U}$, we denote its associated observations and supporting information by

$$u = (\mathcal{I}_u, \Omega_u),$$

where $\mathcal{I}_u$ is the set of observations associated with the entity. For an episode $h \in \mathbb{H}$, we use

$$h = (u, \kappa, [\tau_s, \tau_e], \Omega_h),$$

where u denotes the associated entity, κ denotes the corresponding state or relation description, $[\tau_s, \tau_e]$ is the temporal interval over which the state or relation holds, and Ω_h contains its supporting observations. A transition

$$\delta = (\kappa^- \to \kappa^+, \tau_b, \Omega_\delta)$$

records a change between two consecutive episodes, where κ^- and κ^+ denote the preceding and succeeding states or relations, τ_b is the transition time, and Ω_δ contains the supporting observations. The structural edges in $E_{\mathcal{M}}$ include entity–episode membership, episode participation, transition–episode links, shared-entity links, and temporal predecessor or successor links.

Temporal memory construction. Given frame-level observations $\mathcal{V}$, we first compute pairwise temporal association scores between object observations within a temporal window w. The association score combines appearance, geometric, motion, and local detection-order cues. Observations satisfying the temporal and association constraints are linked to form temporal chains, and each chain is consolidated into a persistent entity $u \in \mathbb{U}$. We then group consistent attributes, states, and relations associated with each entity into temporally bounded episodes $h \in \mathbb{H}$. Changes between consecutive episodes are represented as transition records $\delta \in \mathbb{D}$. The resulting memory $\mathcal{M} = (\mathbb{U}, \mathbb{H}, \mathbb{D}, \mathbb{T})$ is constructed independently of any question and is subsequently shared by all question-conditioned retrieval operations.

Question Parse. Given a question q, we first parse it into a structured query

$$Q(q) = (T_q, C_q, O_q),$$

where T_q identifies the target entity or event, C_q specifies the relevant entity, attribute, relation, spatial, and temporal constraints, and O_q specifies the temporal operation or evidence structure required by the question. This representation provides an interface between the natural language question and the temporal graph memory.

Graph-anchor retrieval. Given the structured query $Q(q) = (T_q, C_q, O_q)$, the graph retrieval function R_G first constructs a candidate set $\mathcal{X}_G$ from the records stored in $\mathcal{M}$. A record is considered compatible with the query if its target, constraints, temporal support, and record type satisfy the corresponding query requirements. We then rank the compatible records using

$$\begin{aligned} s_G(x, q) = {} & w_{\text{lex}} s_{\text{lex}}(x, q) + w_{\text{tar}} s_{\text{tar}}(x, q) + w_{\text{con}} s_{\text{con}}(x, q) \\ & + w_{\text{time}} s_{\text{time}}(x, q) + w_{\text{op}} s_{\text{op}}(x, O_q) + w_{\text{sem}} s_{\text{sem}}(x, q). \end{aligned}$$

All component scores are normalized to a common range before weighted aggregation. Here, s_{lex} measures lexical overlap between the query descriptors and the textual description of x; s_{tar} measures target-entity or target-event matching; s_{con} measures satisfaction of the constraints in C_q; s_{time} measures temporal compatibility; s_{op} measures compatibility between the record type and the temporal operation or evidence requirement O_q; and s_{sem} measures semantic similarity between the query and the canonical record when enabled. The weights are dataset-profile parameters fixed before evaluation and shared across all questions under the same profile. For the frozen NExT-QA graph-retrieval profile, the coefficients are

$$\begin{aligned} w_{\text{lex}} = 3.0, \quad w_{\text{tar}} = 1.1, \quad w_{\text{con}} = 0.9, \\ w_{\text{time}} = 1.2, \quad w_{\text{op}} = 1.1, \quad w_{\text{sem}} = 1.8. \end{aligned}$$

In this profile, exact-entity matching is captured by the target term, question-demand compatibility by the operation term, and canonical-record vector similarity by the semantic term. NAVQA and OpenEQA use the same scoring interface with their respective dataset-specific retrieval profiles and enabled components. Let

$$x^\star = \arg\max_{x \in \mathcal{X}_G} s_G(x, q)$$

denote the highest-ranked compatible graph record, where $\mathcal{X}_G$ is the candidate set formed from the graph records in $\mathcal{M}$. The initial anchor set is

$$A_0 = \{x^\star\}.$$

At expansion step r, the connected anchor set is updated as

$$A_{r+1} = A_r \cup \{x \in \mathcal{X}_G \setminus A_r : \exists y \in A_r,\ (x, y) \in E_{\mathcal{M}}\}.$$

The expansion stops when the dataset-specific graph-evidence budget $K_G^{(d)}$ is reached or when the connected candidate set reaches a fixed point. The final graph-evidence budgets are

$$K_G^{(\text{NAVQA})} = 10, \qquad K_G^{(\text{NExT-QA})} = 16, \qquad K_G^{(\text{OpenEQA})} = 10,$$

as specified in Table 5. The graph-anchor output is

$$B_G^{(d)}(q) = \text{TopK}_{K_G^{(d)}}(A_R, s_G).$$

Thus, R_G performs query-conditioned relevance ranking while preserving the structural connectivity encoded in the temporal memory.

Temporal-subgraph construction. The graph anchors provide the records most directly related to the question, but answering may require their surrounding entity, episode, and temporal context. We therefore construct a temporal subgraph

$$G_q = \text{Subgraph}(\mathcal{M}; B_G(q)),$$

which expands the retrieved anchors according to their structural and temporal links. Its retained node set is

$$V_q = B_G(q) \cup N_{\text{entity}}(B_G(q)) \cup N_{\text{episode}}(B_G(q)) \cup N_{\text{transition}}(B_G(q)) \cup N_{\text{temporal}}(B_G(q), O_q),$$

where N_{entity}, N_{episode}, $N_{\text{transition}}$, and N_{temporal} retrieve entity-linked records, state or relation episodes, transition records, and operation-relevant temporal predecessors or successors, respectively.

The retained structural relations are

$$E_q = \{(x, y) \in E_{\mathcal{M}} : x, y \in V_q\},$$

and the resulting temporal subgraph is

$$G_q = (V_q, E_q, \Omega_q),$$

where Ω_q contains supporting observation identifiers, episode boundaries, and transition times. Expansion is restricted to records connected to the retrieved anchors and terminates when the corresponding subgraph or evidence budget is reached. This construction preserves the persistent identity, temporal extent, and state or relation changes required for subsequent evidence retrieval.

Graph-conditioned visual retrieval. The retrieved temporal subgraph is then used to constrain visual evidence retrieval. For dataset d, the visual retrieval function is

$$B_V^{(d)}(q) = R_V^{(d)}(\mathcal{V}^{(d)}, G_q, Q(q)),$$

where $\mathcal{V}^{(d)}$ denotes the visual candidate space of dataset d. The visual query is constructed from the target entities, requested states or attributes, event and spatial constraints, temporal operation, and the original question:

$$\psi(q) = \text{concat}(T_q, C_q, O_q, q).$$

A visual candidate v is considered eligible if it is supported by the retrieved temporal subgraph and satisfies its associated entity, temporal, and scene constraints. Specifically, we define the eligible candidate set as

$$\mathcal{V}_q^{(d)} = \left\{ v \in \mathcal{V}^{(d)} : \begin{array}{l} \exists x \in V_q,\ \text{entity}(v) \cap \mathcal{E}_x \neq \varnothing, \\ \text{time}(v) \cap [\tau_x^s, \tau_x^e] \neq \varnothing, \\ \text{scene}(v) \text{ is compatible with } G_q \end{array} \right\}.$$

When C_q specifies an explicit temporal interval, we further require the candidate to overlap with that interval. We rank the eligible candidates using the visual relevance score

$$s_V(v, q) = \lambda_{\text{sim}} s_{\text{sim}}(v, \psi(q)) + \lambda_{\text{ent}} s_{\text{ent}}(v, G_q) + \lambda_{\text{time}} s_{\text{time}}(v, G_q) + \lambda_{\text{role}} s_{\text{role}}(v, O_q),$$

where s_{sim} measures visual-query similarity between v and $\psi(q)$, s_{ent} measures consistency between the candidate and the entities in G_q, s_{time} measures temporal compatibility with the retrieved graph context, and s_{role} measures compatibility with the evidence role specified by O_q. All four component scores are normalized to a common range before aggregation. The coefficients $\lambda_{\text{sim}}, \lambda_{\text{ent}}, \lambda_{\text{time}}, \lambda_{\text{role}}$ are fixed within each dataset profile and shared across all questions under the same profile. The corresponding candidate units and visual-evidence budgets are summarized in Table 5. The final visual evidence is obtained by selecting the top-ranked candidates:

$$B_V^{(d)}(q) = \text{TopK}_{K_V^{(d)}}\left(\mathcal{V}_q^{(d)}, s_V\right).$$

The resulting candidate set is fixed before evidence projection and is reused by all subsequent evidence organization strategies, ensuring that differences between these strategies arise from evidence organization rather than from different visual candidate pools.

Evidence-record serialization. The retrieved graph and visual items provide complementary evidence: graph records preserve persistent entities and temporal structure, while visual candidates provide the corresponding visual observations. We therefore serialize the retained evidence into a textual representation for the downstream QA model.

Each retained graph or visual item is represented as

$$e_i = (\text{id}_i, \text{source}_i, \text{kind}_i, \mathcal{E}_i, [\tau_i^s, \tau_i^e], \text{text}_i, \text{prov}_i),$$

where source_i identifies the graph or visual evidence source, kind_i specifies the record type, $\mathcal{E}_i$ contains the associated entities, $[\tau_i^s, \tau_i^e]$ gives its temporal support, text_i is the textual fact, and prov_i records the supporting entity, episode, transition, frame, or observation.

The shared projection is

$$Z_{\text{shared}}(q) = P_{\text{shared}}(G_q, B_V(q)) = \text{Serialize}\left(\text{Sort}\left(\{e_i\}\right)\right).$$

The serialized evidence follows the fixed template

$$[\text{rank}]\ [\text{source}]\ [\text{kind}]\ [\text{entities}]\ [\text{time}]\ [\text{fact}]\ [\text{provenance}].$$

Records are ordered by retrieval relevance and then by temporal support, with the evidence identifier used as a deterministic tie-breaker. The resulting evidence representation $\mathcal{Z}(q)$ is provided together with the question to the downstream QA model:

$$\hat{y} = f_{\text{QA}}(s_{\text{sys}}, q, \mathcal{Z}(q)).$$

Operation-conditioned evidence projection. The operation-conditioned projection is

$$Z_{O_q}(q) = P_{O_q}(G_q, B_V(q)).$$

It consumes the same evidence records as P_{shared} and adds an operation-specific view:

$$P_{O_q} : \begin{cases} \text{interval}, \text{duration} \to (\tau_s, \tau_e), \\ \text{count} \to \text{Deduplicate}(\mathcal{E}_i), \\ \text{order} \to \text{SortByTime}(e_i), \\ \text{previous}, \text{next} \to \text{ReferenceCentered}(e_i), \\ \text{change} \to (\kappa^-, \tau_b, \kappa^+). \end{cases}$$

Thus, P_{shared} and P_{O_q} use the same temporal memory and the same retrieved evidence. Their difference is the organization and textual form presented to the QA model.

.2 Question-Dependent Evidence Analysis

The same temporal graph memory can provide different forms of evidence depending on the information required by a question. We therefore analyze the effect of question-dependent evidence organization across question categories. Figure 4 compares the *demand-aware visual evidence* configuration with the *shared visual evidence* configuration across all 29 question categories.

Table 5: Final dataset-specific instantiations of the retrieval interfaces.

Dataset	Graph retrieval and budget	Visual retrieval and budget
NAVQA	Question-IR-conditioned hybrid graph retrieval over entities, episodes, and transitions; $K_G^{(d)} = 10$.	Same-sequence object-crop and entity-linked temporal retrieval; the final visual evidence budget is $K_V^{(d)} = 8$.
NExT-QA	Route-independent hybrid retrieval over canonical graph records and temporal links; $K_G^{(d)} = 16$.	Frames, objects, tracklets, temporal windows, event boundaries, and before/after pairs; $K_V^{(d)} = 12$.
OpenEQA	Canonical-record vector retrieval with fixed temporal mode; $K_G^{(d)} = 10$.	Episode-scoped object retrieval bound to canonical graph evidence; $K_V^{(d)} = 8$.

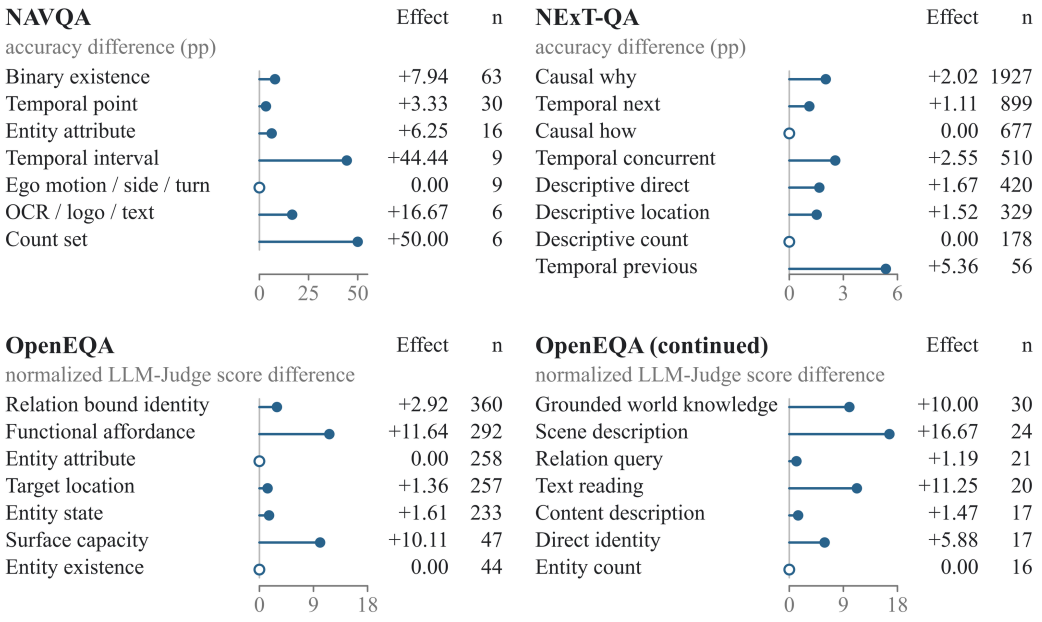


Figure 4: Question-category effects of demand-aware evidence organization. Differences compare demand-aware visual evidence with shared visual evidence across 29 question categories. Hollow markers denote zero change, and n indicates the number of questions in each category. NAVQA and NExT-QA report differences in percentage points, while OpenEQA reports differences in normalized LLM-Judge score points.

Figure 4 shows that the effect of evidence organization varies substantially across question categories. For example, demand-aware organization improves NAVQA temporal-interval questions by $+44.44$ points over 9 questions, NExT-QA temporal-previous questions by $+5.36$ points over 56 questions, and OpenEQA functional-affordance questions by $+11.64$ score points over 292 questions. In contrast, several categories show no aggregate change, including NAVQA ego-motion/side/turn, NExT-QA causal-how and descriptive-count, and several OpenEQA categories.

These variations are consistent with the different evidence structures required by different questions. Temporal-interval questions can benefit from explicit episode boundaries, temporal-order questions can use ordered events or transitions, and entity- or condition-based questions can benefit from persistent entity associations. The results therefore suggest that evidence organization is most useful when it matches the structure of the information required by the question. Because the category sizes vary substantially, we use these results to characterize category-dependent behavior rather than to claim uniform improvements across all question types.

Table 6: Clustered analysis of complete evidence comparisons. Positive, negative, and tied counts refer to the underlying sequence, video, or episode units.

Comparison	Unit	Mean Δ	Positive	Negative	Tied	95% CI
OpenEQA (episode-level)						
Temporal graph $\rightarrow$ Shared visual	Episode	+0.47	60	64	28	$[-0.65, 1.64]$
Shared visual $\rightarrow$ Demand-aware	Episode	+4.11	90	46	16	$[2.30, 5.93]$
NAVQA (sequence-level)						
Temporal graph $\rightarrow$ Shared visual	Sequence	+10.07	6	0	1	$[5.22, 14.77]$
Shared visual $\rightarrow$ Demand-aware	Sequence	+10.79	5	0	2	$[5.00, 16.18]$
NExT-QA (video-level)						
Temporal graph $\rightarrow$ Shared visual	Video	+4.66	245	134	191	$[3.40, 5.92]$
Shared visual $\rightarrow$ Demand-aware	Video	+1.54	190	134	246	$[0.55, 2.51]$

.3 Clustered Uncertainty Analysis

Question-level averages may overrepresent videos or episodes containing many questions. We therefore examine the consistency of the complete-system differences using clustered bootstrap analysis, as shown in Table 6. We use the natural underlying units of the three datasets: navigation sequence for NAVQA, video for NExT-QA, and episode for OpenEQA. For each comparison, we report the question-weighted mean difference, the numbers of positive, negative, and tied clusters, and percentile 95% confidence intervals obtained from 10,000 cluster-bootstrap replicates.

The additional gain from demand-aware visual evidence over shared visual evidence is consistently positive at the clustered level. The corresponding 95% intervals are $[+5.00, +16.18]$ percentage points on NAVQA, $[+0.55, +2.51]$ percentage points on NExT-QA, and $[+2.30, +5.93]$ score points on OpenEQA. Thus, the observed improvement is not driven solely by a small number of questions or individual videos.

The comparison between shared visual evidence and temporal graph memory shows a different pattern on OpenEQA. At the episode level, 60 episodes favor shared visual evidence, 64 favor temporal graph memory, and 28 are tied. The two-sided sign test gives $p = 0.788$, and the corresponding confidence interval includes zero. Therefore, the relatively small average improvement of shared visual evidence over temporal graph memory on OpenEQA is not consistently observed across episodes. Overall, the clustered analysis supports a conservative interpretation of the complete-system results: the additional benefit of demand-aware evidence organization is consistently observed across the underlying evaluation units, whereas the smaller gain obtained by adding shared visual evidence on OpenEQA is less stable at the episode level.

.4 Operational Details of Temporal Association

Equation (2) in subsection 3.3 defines a question-independent association score between two frame-level object observations. The association is performed during temporal memory construction, before any question-conditioned retrieval, and is therefore shared by all questions associated with the same video. Candidate pairs are first restricted by the temporal window, after which their association scores are computed. In the current implementation, the association score is a weighted combination of four cues:

$$S_{\text{assoc}}(o, o') = \lambda_{\text{app}}\phi_{\text{app}}(o, o') + \lambda_{\text{geom}}\phi_{\text{geom}}(o, o') + \lambda_{\text{motion}}\phi_{\text{motion}}(o, o') + \lambda_{\text{order}}\phi_{\text{order}}(o, o').$$

The four terms measure appearance consistency, geometric overlap, motion consistency, and local detection-order consistency, respectively. The area cue described in Section 3.3 is used as an additional candidate-admissibility criterion rather than as a weighted term in S_{assoc}. Thus, an observation pair is linked only when it satisfies the temporal and admissibility constraints and its association score exceeds the threshold.

The appearance cue is defined as the cosine similarity between the visual representations of the two observations:

$$\phi_{\text{app}}(o, o') = \cos(e_o, e_{o'}).$$

Table 7: Fixed operational parameters for temporal association across all three benchmarks.

Symbol	Operational meaning	Setting
K	Number of association cues	4
λ_{app}	Appearance consistency weight	0.45
λ_{geom}	Geometric-overlap weight	0.15
λ_{motion}	Motion-consistency weight	0.35
λ_{order}	Detection-order weight	0.05
w	Temporal association window	1 sampled observation
η	Association-score threshold	0.60

Table 8: Temporal-coordinate conventions used to instantiate the shared temporal axis for each benchmark.

Dataset	Coordinate source	Interpretation
NAVQA	Sequence time index	Dataset-provided temporal coordinate
NExT-QA	Frame index and video FPS	Nominal timestamp
OpenEQA	Sampled-frame order	Ordinal coordinate, not physical seconds

The geometric cue is the bounding-box intersection-over-union:

$$\phi_{\text{geom}}(o, o') = \text{IoU}(b_o, b_{o'}) = \frac{|b_o \cap b_{o'}|}{|b_o \cup b_{o'}|},$$

where b_o and $b_{o'}$ are bounding boxes of objects o and o', respectively. The motion cue measures normalized center displacement:

$$\phi_{\text{motion}}(o, o') = \max\left(0,\, 1 - \frac{d_{\text{center}}(b_o, b_{o'})}{\max\{\min(d_o, d_{o'}), 1\}}\right),$$

where d_o and $d_{o'}$ denote the diagonal lengths of the two bounding boxes. The local detection-order cue is defined as

$$\phi_{\text{order}}(o, o') = \frac{1}{1 + |j_o - j_{o'}|},$$

where j_o and $j_{o'}$ denote the corresponding local detection indices. The operational values of the association window, cue weights, admissibility thresholds, and score threshold are fixed before question-conditioned retrieval and are shared across all questions. Their values in the frozen implementation are reported in Table 7. As summarized in Table 7, the association score combines four continuously weighted cues: appearance consistency, geometric overlap, motion consistency, and detection-order consistency. Their weights are set to 0.45, 0.15, 0.35, and 0.05, respectively. The temporal association window is $w = 1$, and the association threshold is $\eta = 0.60$. Here, w is defined in the order of sampled observations rather than in physical time; thus, $w = 1$ restricts direct association to adjacent sampled observations. These settings are fixed for all datasets and are not tuned per benchmark.

.5 Coordinate Convention

The temporal coordinate τ_t follows the convention of each benchmark, as summarized in Table 8. NAVQA uses its dataset-provided sequence time index, NExT-QA uses the frame index together with the video frame rate to obtain a nominal timestamp, and OpenEQA uses the sampled-frame order defined by its graph-construction protocol. Therefore, the temporal coordinate does not necessarily represent physical seconds across all benchmarks. Instead, it provides a consistent ordering and indexing system for constructing the shared temporal axis. Episode endpoints τ_s, τ_e and transition boundaries τ_b are all defined in the same coordinate system as the corresponding benchmark.

.6 Case Analysis

We examine representative cases to illustrate how the temporal graph memory and subsequent evidence organization affect the available evidence for question answering. We first consider

a temporal-next question, followed by a case where question-dependent evidence organization changes the retrieved context. We then present two additional examples illustrating temporal structure and persistent observations, and finally discuss a failure caused by an incorrect upstream scene-graph relation.

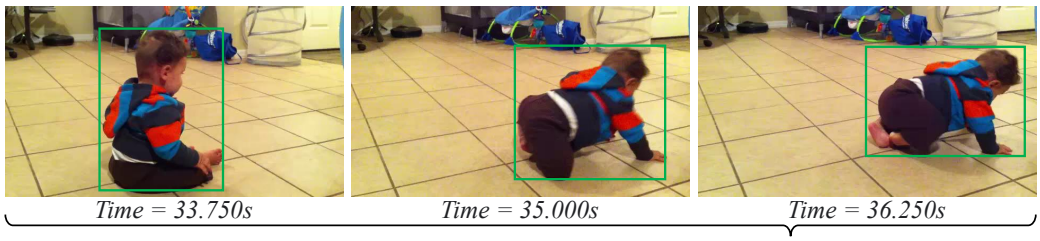
Figure 5 shows a NExT-QA temporal-next question. The temporal graph memory records a sitting observation at 33.750 s and identifies the subsequent crawling observation in the window $[35.000, 36.250]$. However, graph-memory evidence alone leads to the prediction "touch the cloth". When graph-conditioned visual evidence is introduced, both the shared and demand-aware settings retrieve the subsequent crawling observation and predict "crawl". This example illustrates the complementarity between temporal graph structure and visual evidence: the temporal graph identifies the relevant successor window, while the visual observation provides the action information needed for answering. Notably, the sitting and crawling observations have different graph entity IDs, so the example demonstrates successor-window retrieval rather than persistent identity continuity.

Figure 6 provides an example in which the organization of visual evidence according to question demand changes the answer. The temporal graph memory and shared visual evidence identify the target and repeated running observations, but the resulting context leads to the unsupported answer "chased by dog". The demand-aware setting reorganizes the available observations around the information required by the question, retains kite-related evidence, and recovers the correct answer "flying kite". Unlike the previous case, where the main benefit comes from locating the relevant temporal window, this example illustrates the role of question-dependent evidence organization after the relevant temporal context has been identified.

Figures 7 and 8 provide additional examples of how temporal structure is used beyond a single successor relation. In Figure 7, the retrieved sequence follows the man–plane interaction from holding at 2.0 s through a relation boundary at 4.0 s to subsequent taxiing and flight at 6.0–12.0 s. The temporal graph memory alone predicts "spinning", whereas both shared and demand-aware visual evidence recover the correct answer "hold the plane" by incorporating observations around the relevant temporal boundary. This example highlights the role of explicit temporal ordering and episode boundaries in organizing evidence. Figure 8 illustrates a different use of persistent temporal structure: repeated man–rope observations across frames are consolidated into a coherent temporal record. Despite the absence of an explicitly grounded boat entity for the query, all TimeGraph configurations recover the correct answer "hold on to the long rope", whereas the external HGA-BERT-FT baseline predicts "hand signals". The example shows how aggregating repeated observations can preserve answer-relevant evidence even when the queried entity is not explicitly grounded in every observation.

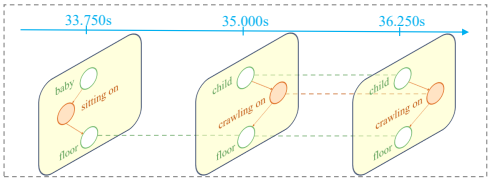
Finally, Figure 9 illustrates a failure caused by an upstream scene-graph error. For the OpenEQA question "What is the white object on the wall above the TV?", the correct visual candidate is an air conditioning unit, but the HSGG graph contains the incorrect relation "painting placed above television". Because this erroneous relation is ranked as the top evidence, all three configurations predict "painting" instead of the ground-truth answer "air conditioning unit". The case shows that evidence organization cannot recover from an incorrect upstream spatial relation when the erroneous graph record dominates evidence ranking.

Together, these cases illustrate three complementary properties of the pipeline. Temporal graph memory can identify relevant temporal context and preserve episode boundaries and repeated observations; graph-conditioned visual retrieval can recover fine-grained visual information that is absent from graph evidence alone; and question-dependent evidence organization can rearrange the retrieved context according to the information required by the question. At the same time, the final failure case shows that these benefits remain dependent on the correctness of the upstream scene-graph representation and evidence ranking.



Question

What does the baby do after sitting for a while in the middle?

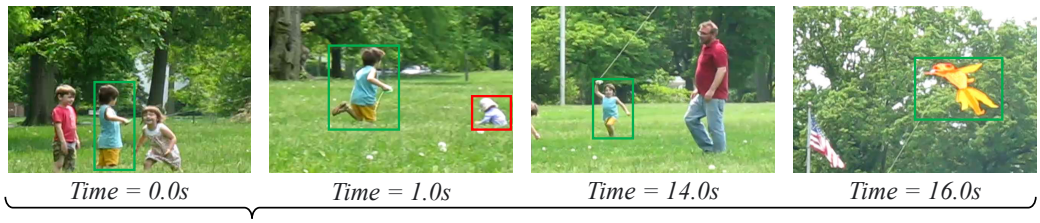


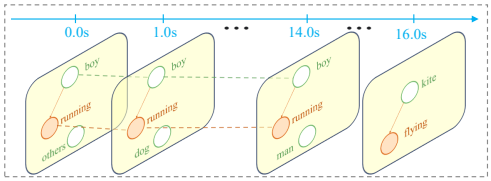
Real evidence in the log

33.750 s: baby #418 — sitting on → floor #419
35.000 s: child #429 — crawling on → floor #419
36.250 s: child #429 — crawling on → floor #419
Subsequent window of the full method reference: 35.000–40.000 s, which includes the crawling on relationship.
Final answer: crawl

Settings	Answer	Result
Temporal graph memory	touch the cloth	Wrong
Shared visual evidence	Crawl	Right
Demand-aware visual evidence	Crawl	Right

Figure 5: **Temporal-next retrieval on NExT-QA.** The temporal graph records a baby sitting on floor #419 at 33.750 s, while the subsequent window $[35.000, 36.250]$ contains a child crawling on the same floor. Graph-memory evidence alone predicts "touch the cloth", whereas both shared and demand-aware visual evidence retrieve the subsequent crawling observation and predict "crawl". The two observations have distinct graph entity IDs, so the example illustrates successor-window retrieval rather than persistent identity continuity.





Question: Why does the boy in blue keep running?

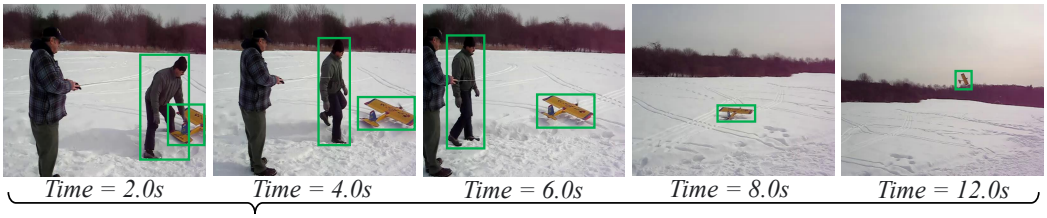
Real evidence in the log

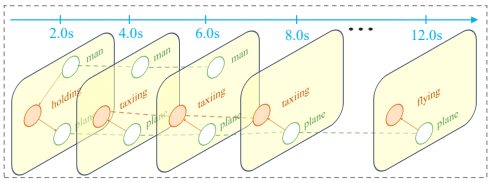
0.000 s: boy in blue
1.001 s: boy in blue
14.014 s: boy, kite string
16.016 s: kite
action: throwing ball
state: running
state: running(boy)
relation: man → watching → boy
attribute: orange kite flying in air

Subject Grounding: The boy in blue is identified as the target.
↓
Action Confirmation: Repeated observations confirm that the boy keeps running.
↓
Causal-Context Retrieval: The **evidence demand** make the retrieved subsequent kite evidence explicitly appear in the causal structure
↓
Hypothesis Verification: The observations of kite confirm this activity, while no evidence supports "chased by dog."
↓
Answer Confirmation: Flying kite.

Settings	Answer	Result
Temporal graph memory	chased by dog	Wrong
Shared visual evidence	chased by dog	Wrong
Demand-aware visual evidence	flying kite	Right

Figure 6: **Question-dependent evidence organization.** Temporal graph memory and shared visual evidence identify the target and repeated running observations but yield the unsupported answer "chased by dog". Organizing the observation context according to the question demand retains kite-related evidence and recovers the correct answer "flying kite". The case illustrates how the same retrieved temporal context can support different evidence organizations for different question demands.





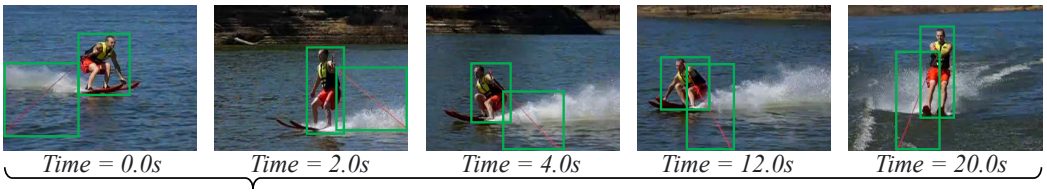
Question: What did the man in grey do before the plane took off?

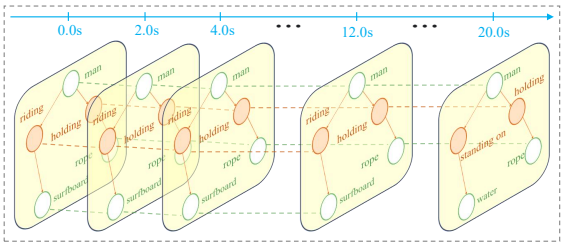
Real evidence in the log
2.0 s: man2 — holding → plane
4.0 s: end boundary of the relationship between man and the plane
6.0 s: plane continues moving away from the man
8.0 s: plane continues to taxiing
12.0 s: plane flying in sky
Final answer: hold the plane

Question-Demand: The question requires evidence before target event.
↓
Subject and Event Grounding: The man in grey and the plane are identified as the target entities, and the flying event is localized on the shared time axis.
↓
Ordered Evidence Retrieval: the earlier relation "man holding plane" with the temporal window surrounding the departure boundary.
↓
Temporal Continuity Verification: The observations at 8.0 s and 12.0 s show that the flying event continues after the end boundary.
↓
Answer Confirmation: Hold the plane.

Settings	Answer	Result
Temporal graph memory	spinning	Wrong
Shared visual evidence	hold the plane	Right
Demand-aware visual evidence	hold the plane	Right

Figure 7: **Temporal-before retrieval around a relation boundary.** The retrieved sequence follows the man–plane interaction from holding at 2.0 s, through the relation boundary at 4.0 s, to subsequent taxiing and flight at 6.0–12.0 s. Temporal graph memory alone predicts "spinning", whereas both shared and demand-aware visual evidence recover the correct answer, "hold the plane", by incorporating observations around the relevant temporal boundary.





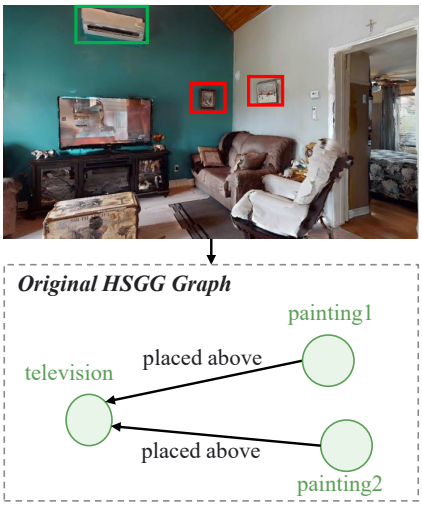
Question: How did the man in red pants keep contact with the speeding boat?

Why TimeGraph helps: Individual frame graphs are highly similar, but **TimeGraph** aggregates recurring man–rope evidence into a persistent interaction record, supporting the answer "hold on to long rope."

Question Demand: The interaction mechanism must be inferred across time.
↓
Persistent Memory: TimeGraph links recurring man–rope observations.
↓
Evidence Consolidation: Cross-frame evidence forms a coherent contact chain despite the absence of a single frame showing the complete man–boat interaction.
↓
Answer: Hold on to long rope.

Settings	Answer	Result
HGA-BERT-FT	hand signals	Wrong
Temporal graph memory	hold on to long rope	Right
Shared visual evidence	hold on to long rope	Right
Demand-aware visual evidence	hold on to long rope	Right

Figure 8: **Persistent temporal evidence across repeated observations.** Repeated man–rope observations across frames form a coherent temporal record. Despite the absence of an explicitly grounded boat entity for the query, all TimeGraph configurations recover the correct answer, "hold on to the long rope", whereas the external HGA-BERT-FT baseline predicts "hand signals". The example illustrates how repeated observations are consolidated into a temporally coherent record.



Question

What is the white object on the wall above the TV?

Real evidence in the log
Correct visual candidate: air conditioner
Incorrect relation: painting placed above television
Incorrect relation becomes **rank-1 evidence**
Evidence rank 1:"Spatial answer candidate painting: painting above television"
Final answer: painting
GT: air conditioning unit

Settings	Answer	Result
Temporal graph memory	Painting	Wrong
Shared visual evidence	Painting	Wrong
Demand-aware visual evidence	Painting	Wrong

Figure 9: **Failure from spatial-relation grounding and evidence ranking in OpenEQA.** For the question "What is the white object on the wall above the TV?", the correct visual candidate is an air conditioning unit. However, the HSGG graph encodes the incorrect relation "painting placed above television", which is ranked first. All three configurations consequently predict "painting" instead of "air conditioning unit". The case illustrates that evidence organization cannot recover from an incorrect upstream spatial relation when the erroneous graph record dominates evidence ranking.