Beyond Flat Frames: Hierarchical Graph Reasoning for Long Video Understanding

Xinyue Liu[b,c] **Jiayang Sun**[b,c] **Zhe Jing**[a*] **Jie Cao**[b,c] **Ran He**[b,c] **Huaibo Huang**[b,c]

[a] School of Automation, Beijing Institute of Technology, Beijing, China
[b] MAIS & NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[c] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

{liuxinyue2025,sunjiayang2025}@ia.ac.cn jzjx12345@163.com
rhe@nlpr.ia.ac.cn huaibo.huang@cripac.ia.ac.cn

Abstract

Recent studies have shown that leveraging the reasoning and analytical capabilities of large language models (LLMs) for long video understanding has become a promising approach. However, these methods are constrained by a structural limitation: representing video as a flat frame sequence makes it difficult to model the temporal structure and logical dependencies of events, which in turn hampers long-range reasoning and risks overlooking important information. To address this limitation, we propose **LVGraph**, a framework that models video content using a coarse-to-fine hierarchical graph. LVGraph constructs a semantic hierarchy where high-level graphs characterize the global event structures within the video, while low-level graphs capture fine-grained, query-relevant information. Instead of linear scanning, reasoning is performed via a query-aware traversal of this graph, adaptively identifying and retrieving the most salient keyframes by navigating the nodes and edges most relevant to the query. Finally, we leverage a Vision Language Model (VLM) to augment the keyframe information, and subsequently answer the question by reasoning over the graph and the enriched keyframe captions. Extensive experiments on long video understanding benchmarks confirm the effectiveness of our method. LVGraph significantly outperforms existing LLM-based state-of-the-art methods on EgoSchema, NExT-QA, and long-duration segments (averaging 44 minutes) of the Video-MME benchmarks.

1 Introduction

Video understanding is a challenging task in computer vision, as it requires the joint modeling and reasoning of spatiotemporal information to accurately capture dynamic changes and the evolution of events within videos. While prior work has shown promising results on short video processing, understanding long videos remains a significant challenge. This task demands not only a comprehensive global grasp of the video content but also a detailed interpretation of individual keyframes—otherwise, the lack of sufficient information may lead to inaccurate responses.

To support complex video understanding, recent video Multimodal Large Language Model(MLLMs) Liu et al. [2024], Li et al. [2024b], Wang et al. [2024a], Song et al. [2024], Lin et al. [2024], Li et al. [2024c, 2026, 2023], Chen et al. [2024a] aim to align visual and linguistic modalities in an end-to-end fashion and project them into a shared semantic space, enabling large language models (LLMs) to interpret video content. However, since a single image typically consumes substantial computational

*Corresponding author

Preprint.

cost in vision language models (VLMs), these methods face a central challenge when dealing with long videos: the limited token budget constrains the visual representation, making it difficult to simultaneously capture fine-grained spatiotemporal details and model long-range dependencies. Although some MLLM-based approaches Li et al. [2024c], Liu et al. [2025], Song et al. [2024], Zeng et al. [2025], Li et al. [2023], Chen et al. [2024a] attempt to mitigate this issue through token compression and frame sampling, they still often suffer from hallucinations and the loss of critical information, leading to sub-optimal responses.

Recently, several approaches Wang et al. [2024b], Zhang et al. [2023], Ma et al. [2025], Wang et al. [2025d], Fan et al. [2024], Wang et al. [2023], Yan et al. [2025] have emerged that leverage the commonsense knowledge and reasoning capabilities of Large Language Models (LLMs) to address the challenge of long video understanding without additional training. A typical pipeline first employs VLMs to describe video frames, converting visual information into textual representations. The LLM then performs reasoning over the video content to answer the given query.

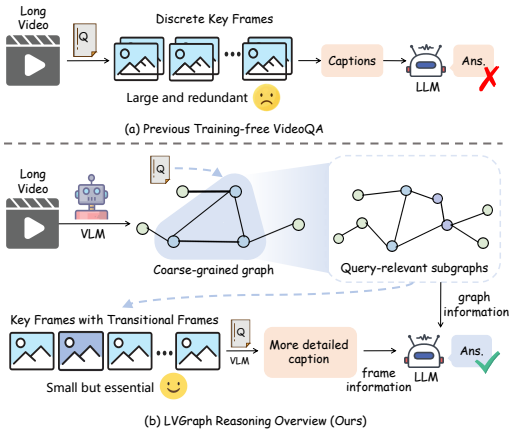


Figure 1: (a) Existing methods treat videos as linear frame sequences, which often fails to capture complex event structures and leads to unreliable reasoning despite using many keyframes. (b) LVGraph builds a hierarchical video graph and retrieves a compact query-relevant subgraph with keyframes and transitional frames, enabling accurate and efficient long-video reasoning.

While these LLM-based methods have demonstrated strong potential in long-range reasoning and achieved promising results, two key areas remain for further exploration: **1) Structural Modeling of Long-Form Video.** Long-form videos are inherently complex tapestries of events rather than monolithic streams. Current approaches Zhang et al. [2023], Ma et al. [2025], Wang et al. [2023] often treat videos as linear sequences or collections of disconnected segments. This perspective, while practical, can limit the ability to capture the intricate causal and temporal fabric connecting disparate events. As a result, reasoning over long temporal horizons may not fully reflect the underlying event evolution. **2) Reliance on Frame-Level Semantic Matching.** To locate query-relevant content, existing methods Ma et al. [2025], Wang et al. [2024b, 2025d], Fan et al. [2024], Wang et al. [2023] generally focus on direct semantic matching at the frame level. While this strategy is effective for identifying key moments, it may sometimes overlook transitional frames that are semantically distant from the query but vital for establishing a continuous event chain. The resulting semantic gaps can challenge the downstream LLM, potentially affecting the coherence and accuracy of its final conclusions.

To address the aforementioned challenges, we propose **LVGraph**, a training-free framework for long-form video understanding. Unlike the spatially-oriented graph in VideoMindPalace Huang et al. [2025], LVGraph constructs a hierarchical structure optimized for modeling temporal evolution and causal dependencies. LVGraph constructs a coarse-to-fine hierarchical graph to model video information and can adaptively retrieve query-relevant keyframes and transitional frames from the graph. The core idea is to reconstruct a holistic representation of the video using a compact graph structure, and to leverage the graph's strengths in long-range reasoning and critical information localization to enable accurate understanding of long-form videos.

Specifically, as shown in Fig. 1, LVGraph consists of three major stages:hierarchical graph construction, graph-based frame retrieval, and keyframe information enhancement.

Inspired by the human coarse-to-fine perception process, we first construct a hierarchical graph that captures the video's semantic structure. A coarse-level graph models major events and interactions, while a query-guided expansion produces a fine-grained, query-centric graph where nodes represent salient objects and edges denote relationships such as human-object interactions and spatial layouts. To retrieve query-relevant frames, we employ a graph-based frame retrieval mechanism that establishes explicit correspondences between video frames and the graph's nodes and edges,

enabling precise localization of keyframes by identifying entities and relations pertinent to the query. A centrality-enhanced multi-hop expansion further incorporates transitional frames that, though not directly aligned with the query, are critical for maintaining event continuity. Finally, we use VLMs to enrich the retrieved frames with detailed, query-focused captions, and the final answer is inferred by reasoning jointly over the hierarchical graph and these enhanced captions.

LVGraph demonstrates significant improvements over the strongest existing training-free methods on each benchmark, achieving gains of 3.9% on EgoSchema Mangalam et al. [2023], 3.1% on the NExT-QA validation set Xiao et al. [2021], and 11.5% on the long-video split of Video-MME Fu et al. [2025]. It consistently outperforms existing LLM-based state-of-the-art methods, while requiring fewer keyframes for inference. Notably, our method even outperforms the powerful proprietary GPT-4o model on the demanding long-clip benchmark of Video-MME. Furthermore, our ablation studies confirm that LVGraph consistently outperforms comparable methods (including VideoTree Wang et al. [2025d], VideoAgent Wang et al. [2024b], and LLoVi Zhang et al. [2023]) at all evaluated frame counts, showcasing a superior trade-off between efficiency and effectiveness.

The contributions of this work are summarized as follows:

- We propose a novel hierarchical, coarse-to-fine graph representation that structures long videos by capturing both high-level event progression and fine-grained, query-specific details.
- We present a graph-based retrieval mechanism that sources not only query-relevant keyframes but also structurally-critical transitional frames, ensuring temporal and causal coherence.
- LVGraph achieves state-of-the-art performance on multiple long-video benchmarks, including EgoSchema and the long-form splits of Video-MME, outperforming existing methods based on LLMs.

2 Related Work

2.1 MLLMs for Long Video Understanding.

Recent video MLLMs Liu et al. [2024], Li et al. [2024b,a], Wang et al. [2025c, 2024c,a], Song et al. [2024], Lin et al. [2024], Li et al. [2024c, 2026, 2023], Chen et al. [2024a] encode video frames as visual tokens projected into the LLM's space to enable video understanding and reasoning. A primary challenge in long video understanding is capturing both fine-grained spatiotemporal details and long-range dependencies within the limited context window of LLMs. To address this, some methods adopt token compression strategies Song et al. [2024], Nguyen et al. [2022], Li et al. [2024c], Weng et al. [2024], Liu et al. [2025], Wang et al. [2025a], such as token merging, pruning, or hierarchical encoding, to reduce the number of visual tokens while preserving essential information and alleviating the computational burden on the LLM.

Another line of work focuses on frame sampling or keyframe selection Guo et al. [2025b], Tang et al. [2025], Ye et al. [2025], Park et al. [2024] to identify task- or query-relevant frames for video question answering. For example, VSLS Guo et al. [2025b] leverages query logic to guide keyframe selection. Beyond frame selection, MECD Chen et al. [2024b] formulates multi-event video reasoning as causal discovery, using Granger-causality-inspired masked event prediction and causal inference to uncover dependencies among temporally ordered events. Although frame sampling reduces visual redundancy, most existing methods treat videos as linear sequences and rely primarily on local semantic relevance, potentially overlooking transitional frames and structurally important events essential for capturing long-range temporal dependencies.

2.2 Long Video Understanding with LLMs.

Inspired by the strong reasoning capabilities of large language models (LLMs), recent studies Wang et al. [2024b], Zhang et al. [2023], Ma et al. [2025], Wang et al. [2025d], Fan et al. [2024], Wang et al. [2023], Zuo et al. [2025], Yan et al. [2025] have explored using LLM agents for long video understanding. These methods convert visual information into text using multimodal large language models (MLLMs), and then apply LLMs for reasoning and planning, significantly improving understanding without additional training. For example, VideoAgent Wang et al. [2024b] uses an LLM

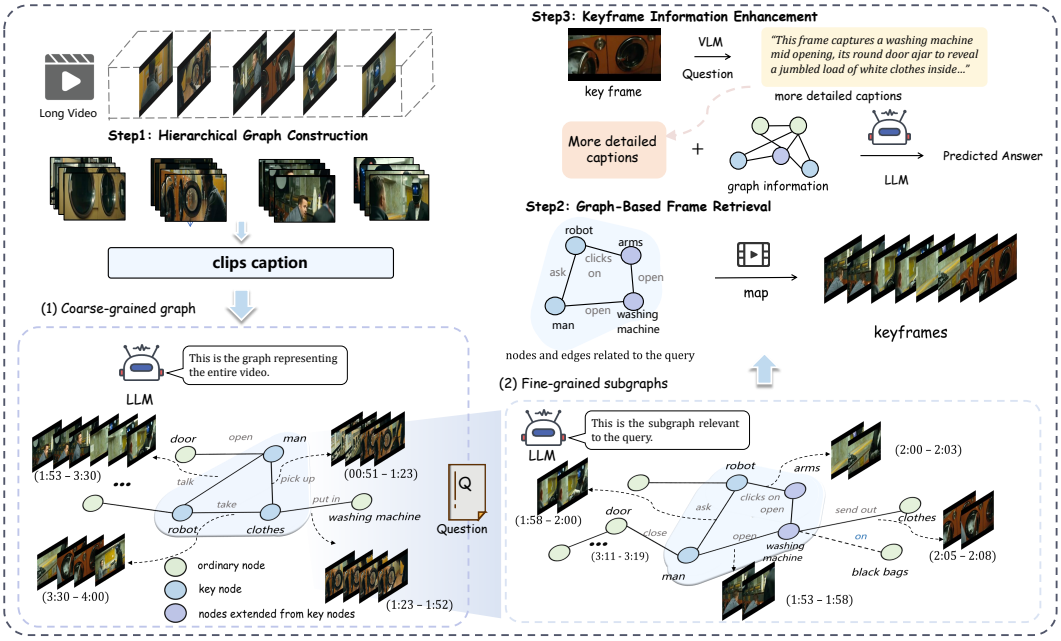


Figure 2: **Overview of our LVGraph framework.** LVGraph consists of three main steps. First, it segments the video into clips and constructs a hierarchical graph, which is refined into a query-specific subgraph. Second, it retrieves question-relevant nodes, edges, and keyframes from the graph. Third, it enhances the selected keyframes with detailed query-aware descriptions. The final answer is generated by jointly reasoning over the graph structure and enhanced keyframe information.

as the central controller and coordinates with CLIP to iteratively retrieve and aggregate keyframes for efficient long-video analysis. Similarly, DrVideo Ma et al. [2025] formulates video understanding as a long-document retrieval problem and employs an agentic loop to dynamically locate relevant visual evidence. These methods improve efficiency by selecting a small set of informative frames instead of processing the entire video. However, they mainly operate at the level of individual frames and lack explicit modeling of the spatiotemporal and semantic relationships among events. As a result, they may struggle to capture event progression, causal dependencies, and cross-segment interactions in complex long videos. In contrast, our method, LVGraph, organizes long video content using a coarse-to-fine hierarchical graph structure, which improves both efficiency and performance in long video understanding.

3 Methodology

We propose LVGraph, a query-conditioned coarse-to-fine graph framework for efficient and accurate long-video reasoning. As illustrated in Fig. 2, the proposed method consists of three main components. In this section, we first introduce Hierarchical Graph Construction in Section. 3.1. Next, we detail Graph-Based Frame Retrieval in Section. 3.2. Finally, we present Keyframe Information Enhancement in Section. 3.3.

3.1 Hierarchical Graph Construction

Long video data is usually highly redundant. For a given video query, it is necessary not only to deeply understand certain key frames in the video, but also to grasp the overall content of the video over a longer temporal span. However, existing methods Wang et al. [2024b], Zhang et al. [2023], Wang et al. [2025d], Fan et al. [2024] mainly aim to locate frames relevant to the question, without building multi-level and multi-granular understanding of long videos. This often leads to incorrect answers due to incomplete context. we introduce a hierarchical graph to model long video information, reorganizing content across different temporal scales through a compact graph representation. First, to obtain a coarse graph that summarizes the video globally, we divide the entire video V into k

clips $[c_1, c_2, \ldots, c_k]$, where each clip corresponds to a certain frame range. We use f_i to denote the i-th frame in the video, and $v(i) = k$ indicates that frame f_i belongs to the k-th clip. Then, we use large vision-language captioning models (e.g., LLaVA-NeXT Liu et al. [2024]) to obtain the text description of each clip. The aggregated descriptions $Clips_{caption}$ are then used to construct a global temporal entity-relation graph:

$$G_1 = VGraph(Clips_{caption}, P_{G1}) \tag{1}$$

where $VGraph$ denotes the LLM, P_{G1} denotes the associated instruction prompt. The resulting graph $G_1 = (V, E)$ comprises nodes V representing key entities (e.g., people, objects), and edges E describe their semantic relations (e.g., actions, spatial relations). Each node carries its associated attributes, and each edge is linked to the frames where the interaction occurs, ensuring temporal grounding. With the global graph G_1 constructed, we further extract a question-relevant subgraph:

$$G_2 = VQGraph(G_1, Q, P_{G2}) \tag{2}$$

where $VQGraph$ denotes the LLM, Q is the input question, and the subgraph $G_2 \subseteq G_1$ relevant to the question. Leveraging the pre-established frame-edge mapping, we retrieve the corresponding frame set $V_{question}$ and further partition it into shorter clips. Compared with the earlier coarse segmentation, these clips contain significantly fewer frames, allowing a more detailed depiction of the video content. We then rebuild the subgraph G_2 from these clips, resulting in an updated structure enriched with finer and more precise information.

We structure the video as a time-ordered entity–relation graph, offering a clearer and more compact way to capture most query-relevant information. This design helps suppress the heavy redundancy typical of long videos and makes the representation easier to interpret.

3.2 Graph-Based Frame Retrieval

With the graph constructed, we obtain an initial understanding of the video. However, for questions that demand fine-grained frame-level details, the graph alone remains inadequate. To compensate, we further extract keyframes tied to the query by analyzing the graph structure. We first identify question-related nodes and edges, then fetch the corresponding frames.

Specifically, we parse the question to extract key semantic components, including actions, objects, as well as temporal expressions such as "before" and "after." This yields a structured semantic representation denoted as Q_p. Subsequently, we find the nodes and edges in the graph $G_2 = (V, E)$ that are highly semantically relevant to the question. For each edge $e_i = (v_a, v_b) \in E$, we construct a textual description $T(e_i) = concat(v_a.name, e_i.action, v_b.name)$, and retrieve the top-$K$ edges based on the cosine similarity between $T(e_i)$ and the query text Q_p:

$$Sim(P, Q) = \cos(\phi_{emb}(P), \phi_{emb}(Q)) \tag{3}$$

$$topk_edges = \arg *topK_i \; Sim(T(e_i), Q_p) \tag{4}$$

where $\phi_{emb}(T(e_i))$ and $\phi_{emb}(Q_p)$ denote the embedding vectors of the edge description and the structured query, both of which are obtained using the OpenAI embedding model.

After obtaining the seed node set that is directly relevant to the question semantics, we further apply a centrality-enhanced multi-hop expansion strategy. This strategy aims to capture keyframes that are not directly semantically related to the question but play an essential role in the event logic, thereby improving the causal and temporal consistency between frames. At each expansion step, both semantic relevance and structural centrality of candidate nodes are jointly considered. Intuitively, relying solely on semantic similarity may overlook key nodes with high connectivity in the graph structure. These nodes might not directly match the query semantically but often play hub roles in the event logic (e.g., key frames connecting multiple sub-events). By incorporating centrality scores, the algorithm identifies nodes frequently traversed by paths originating from different seed nodes, thereby enhancing the coherence and robustness of the event representation.

The comprehensive score for a candidate neighbor node v is calculated as follows:

$$S(v) = \gamma^k \cdot \big(w \cdot \mathrm{Sim}(v, Q) + (1 - w) \cdot Centrality(v)\big) \tag{5}$$

where $Sim(v, Q)$ measures the semantic similarity between node v and the query Q, while $Centrality(v)$ reflects the normalized frequency with which the node is reached by different seed

Table 1: Comparison with state-of-the-art training-free methods on EgoSchema and NExT-QA.

Model	(M)LLM	EgoSchema	NExT-QA			
			Tem.	Cau.	Des.	Avg.
Based on Proprietary MLLMs						
IG-VLM	GPT-4V	59.8	63.6	69.8	74.7	68.6
LVNet	GPT-4o	68.2	65.5	75.0	81.5	72.9
Based on Open-source Captioners and LLMs						
MVU	Mistral-13B	60.3	55.4	48.1	64.1	55.2
LangRepo	Mixtral-8x7B	66.2	51.4	64.4	69.1	60.9
LLoVi	GPT-3.5	57.6	–	–	–	–
MoReVQA	PaLM-2	–	64.6	70.2	–	69.2
LLoVi	GPT-4	61.2	61.0	69.5	75.6	67.7
VideoAgent	GPT-4	60.2	64.5	72.7	81.1	71.3
VideoAgent	GPT-4	62.8	–	–	–	–
LifelongMemory	GPT-4	64.1	–	–	–	–
VideoTree	GPT-4	66.2	70.6	76.5	83.9	75.6
VideoMindPalace	GPT-4	70.1	–	–	–	75.8
DrVideo	GPT-4	66.4	–	–	–	–
CogniGPT	GPT-4	69.2	–	–	–	76.8
LVGraph (Ours)	GPT-4	**72.2**	**75.4**	**79.7**	**82.2**	**78.6**
LVGraph (Ours)	DeepSeek-R1	**74.0**	**77.1**	**80.4**	**84.1**	**79.9**

Table 2: MLVU comparison.

Method	Avg
Proprietary Models	
GPT-4V	49.2
GPT-4o	64.6
Open-sourced MLLMs	
VideoLLaMA2-72B	61.2
InternVL2-76B	59.9
Video-CCAM-14B	63.1
Video-XL-7B	64.9
LLaVA-OV-72B	66.4
LinVT-7B	68.9
Aria-25B	70.6
Oryx-1.5-32B	72.3
VideoLLaMA3-7B	73.0
VideoChat-Flash-7B	74.7
Training-free Methods	
VideoTree	60.4
LVGraph (Ours)	**65.3**

paths. The weight w balances semantic and structural information, and the decay factor r controls how a parent node's score attenuates as it propagates to child nodes, ensuring stable score updates during expansion. The variable k indicates the current hop count, starting from the seed node ($k = 0$) and increasing by one at each step. To limit expansion, we set a degree threshold and keep only the Top-K highest-scoring nodes at each hop, ensuring semantic relevance and effectively capturing key transitional nodes within the video. More details are in Appendix.

3.3 Keyframe Information Enhancement

Through the previous search process, we obtain key frames from the video. However, the captions associated with these frames are often coarse-grained. For example, a caption such as "a man holding a puppy" provides limited information and is insufficient to answer a question like "What color is the puppy in the man's arms?" In such cases, an LLM may be forced to make uninformed or biased guesses.

To address this, we extract more detailed information from these keyframes using the VLM. Specifically, we formulate this as

$$\text{Caption}_{f_i} = \text{VLM}(Q, f_i, P_{cap}), \tag{6}$$

where Q is the question, f_i is the i-th key frame, and P_{cap} represents the prompt template used to elicit detailed visual information. Finally, the subgraph relevant to the question and the enriched key frame descriptions are provided to the LLM to generate the final answer.

4 Experiments

In this section, we first introduce the datasets and implementation details, and then present the results, analysis, and ablation studies of LVGraph.

Datasets and Metrics. We evaluate our LVGraph on four diverse long video question-answering benchmarks, demonstrating its capability for zero-shot long video understanding. EgoSchema Mangalam et al. [2023] provides 5K multiple-choice questions paired with 250 hours of egocentric video covering diverse daily activities. Only 500 questions are publicly available, and full evaluation is performed through the official leaderboard. NExT-QA Xiao et al. [2021] is a video question answering benchmark designed for causal and temporal reasoning. It comprises 5,440 natural videos

Table 3: Video-MME (w/o subs) comparison.

Method	Short	Medium	Long	Avg.
Proprietary Models				
Gemini 1.5 Flash	78.8	68.8	61.1	70.3
GPT-4o	80.0	70.3	65.3	71.9
Gemini 1.5 Pro	81.7	74.3	67.4	75.0
Open-sourced MLLMs				
LLaVA-NeXT-Video	61.7	50.1	44.3	52.0
ViLAMP-7	78.9	65.8	57.8	67.5
VideoChat-Flash-7B	78.0	67.8	55.6	65.3
VideoLLaMA3-7B	80.1	63.7	54.9	66.2
Qwen2-VL-72B	80.1	71.3	62.2	71.2
AdaReTake-72B	80.6	74.9	65.0	73.5
LLaVA-OneVision-72B	76.7	62.2	60.0	66.3
LLaVA-Video-72B	81.4	68.9	61.5	70.6
InternVL2.5-72B	82.8	70.9	62.6	72.1
Training-free Methods				
VideoAgent	–	–	46.4	–
VideoTree	67.8	59.9	54.2	60.6
DrVideo	–	–	51.7	–
CogniGPT	–	–	54.7	–
MemVid	73.9	63.1	55.0	64.0
LVGraph (Ours)	**75.5**	**69.3**	**66.5**	**70.4**

Table 4: Comparison of different foundation models performance.

LLM	Type	Size	Acc. (%)
Mistral-8x7B	Open-Source	8x7B	69.8
DeepSeek-R1	Open-Source	671B	74.0
GPT-4	Proprietary	N/A	72.2
GPT-4o	Proprietary	N/A	72.4
GPT-5.1	Proprietary	N/A	72.8

Table 5: Ablation study results of LVGraph components.

Module	EgoSchema	Video-MME
LVGraph	74.0	70.4
– Hie. Graph Construction	71.2	67.6
– Graph-based Retrieval	72.4	68.3
– Keyframe Enhance.	73.1	69.0

Table 6: Ablation study on the balancing weight w in the centrality-enhanced multi-hop expansion strategy.

w	EgoSchema	Video-MME (long split)
1.0	72.7	63.7
0.8	**74.0**	65.8
0.7	73.7	**66.5**
0.2	73.1	64.9
0.0	71.5	63.0

Table 7: Performance with different max hops.

Max Hops	ES Acc
1	72.1
2	**74.0**
3	72.9

Table 8: Performance with different Top-K edges.

Top-K	ES Acc
3	72.4
6	**74.0**
9	73.6
12	72.7

featuring object interactions in everyday scenarios, accompanied by approximately 52K questions. The questions are categorized into three types: Temporal, Causal, and Descriptive, enabling a comprehensive assessment of video understanding models. Video-MME Fu et al. [2025] offers 900 videos with 2,700 questions, grouped by duration into short (<2 min), medium (4–15 min), and long (30–60 min). All results are reported under the no-subtitle setting. MLVU Zhou et al. [2024] is a long-video benchmark with nine tasks over 600 videos ranging from 3 minutes to 2 hours.

Implementation Details. For all main results, we adopt the open-sourced model DeepSeek-R1 Guo et al. [2025a] as our LLM. We also report results using other open-source LLMs and proprietary LLMs. For the EgoSchema Mangalam et al. [2023] benchmark, we use LaViLa Zhao et al. [2023] as the captioner, while for the NExT-QA Xiao et al. [2021] benchmark, we use CogAgent Hong et al. [2024]. For the MLVU Zhou et al. [2024] and Video-MME Fu et al. [2025] benchmarks, we adopt LLaVA-NeXT Liu et al. [2024] as the captioning model. Additional details are provided in the appendix.

4.1 Main Results

Comparison with Training-Free Methods. Tab. 1 compares LVGraph with existing training-free methods on the EgoSchema Mangalam et al. [2023] and NExT-QA Xiao et al. [2021] benchmarks. To ensure a fair comparison, in addition to using Deepseek-R1 Guo et al. [2025a] as the LLM, we also conducted experiments using GPT-4 OpenAI [2023]. Compared to methods built on the same VLM

Table 9: Performance comparison of LLoVi and VideoTree methods.

Method	Captions	Captioner (s)	Graph Cons. (s)	Keyfr. (s)	QA (s)	Overall (s)	Acc.
LLoVi	180	22.4	0	0	2.4	24.8	61.2
VideoTree	62.4	7.8	0	10.2	2.1	20.1	66.2
LVGraph	16	2.1	16.8	1.9	3.8	24.6	**74.0**

and LLM Ranasinghe et al. [2024], Kahatapitiya et al. [2025], Zhang et al. [2023], Min et al. [2024], Wang et al. [2024b], Fan et al. [2024], Wang et al. [2023, 2025d], Huang et al. [2025], Ma et al. [2025], Li et al. [2025], LVGraph achieves clearly superior results, outperforming VideoMindPalace by 3.9% on EgoSchema and CogniGPT by 3.1% on NExT-QA. Additionally, our method demonstrates substantial advantages over approaches based on open-source LLMs, underscoring the critical role of a strong LLM reasoning module.

Comparative results on longer video benchmarks. To further highlight the strength of our approach in long video understanding, we evaluate LVGraph on two additional benchmarks: Video-MME Fu et al. [2025] and MLVU Zhou et al. [2024]. The long-video subset of Video-MME Fu et al. [2025] contains videos up to 1 hour in length, with an average duration of 44 minutes. We categorize existing methods into three groups: proprietary models, open-source MLLMs trained on large-scale video or image data, and training-free approaches. As shown in Tab. 3, on the Video-MME benchmark, our LVGraph outperforms all existing training-free methods, achieving an average accuracy that surpasses the previous state-of-the-art MemVid Yuan et al. [2025] by 6.4%. Notably, in long video understanding tasks, LVGraph achieves the best performance among all open-source models, exceeding Adaretake-72B Wang et al. [2025b] by 1.5%, and reaching a level comparable to leading commercial proprietary models such as Gemini 1.5 Pro Team et al. [2024] and GPT-4o OpenAI [2024]. As shown in Tab. 2, on the MLVU benchmark Zhou et al. [2024], LVGraph achieves a 4.9% absolute improvement over the training-free baseline VideoTree Wang et al. [2025d] , demonstrating its effectiveness in real-world scenarios where video length is unknown. Compared to proprietary MLLMs, LVGraph outperforms the long-context proprietary model GPT-4o OpenAI [2024] by 0.7%. Furthermore, despite requiring no training, LVGraph surpasses several strong open-source MLLMs that have been extensively trained on large-scale video data, including Video-CCAM-14B Fei et al. [2024] and Video-XL-7B Shu et al. [2025].

4.2 Ablation Studies

In this section, we examine four aspects of LVGraph: the effects of different foundation models, centrality-enhanced multi-hop expansion, individual components, and key hyperparameters. Additional results on the number of input frames are provided in the Appendix.

Effects of Different Foundation Models. We evaluate LVGraph with both open-source (Mistral-8x7B Jiang et al. [2023], DeepSeek-R1 Guo et al. [2025a]) and proprietary foundation models (GPT-4 OpenAI [2023], GPT-4o OpenAI [2024], GPT-5.1 OpenAI [2025]), as shown in Tab. 4. Performance varies across models and generally improves with stronger model capabilities, demonstrating the scalability of LVGraph. Notably, DeepSeek-R1 outperforms GPT-5.1 in our setting. These results suggest that LVGraph is not tied to a specific LLM and can benefit from future advances in foundation models.

Centrality-enhanced Multi-hop Expansion. We ablate the balancing weight w in Eq. 5 to investigate the respective contributions of semantic similarity and graph centrality. As shown in Tab. 6, using only semantic similarity ($w = 1$) or only centrality ($w = 0$) yields lower performance on both EgoSchema and Video-MME. Semantic similarity helps identify nodes relevant to the query, while graph centrality captures the structural importance of nodes and helps uncover useful connections beyond directly relevant evidence. Relying on either cue alone may therefore limit the effectiveness of multi-hop expansion. By combining these complementary signals, our method achieves the best results, reaching 74.0% on EgoSchema at $w = 0.8$ and 66.5% on Video-MME at $w = 0.7$. The different optimal weights also suggest that the balance between semantic relevance and structural



Question: What are the primary components that c interacted with while working on the laptop and what was the overarching goal of this sequence?

A.C interacted with the laptop's screen, keyboard, and mouse. the overarching goal of this sequence was to clean the laptop.
B.C interacted with the laptop's battery, hard drive, and optical drive. the overarching goal of this sequence was to upgrade the laptop's components.
C.C interacted with the laptop's case, bezel, and keyboard deck. the overarching goal of this sequence was to repair the laptop.
D.C interacted with the laptop's motherboard, chips, and wires. the overarching goal of this sequence was to remove the motherboard from the laptop.
E.C interacted with the laptop's power button, volume buttons, and sleep button. the overarching goal of this sequence was to configure the laptop's settings.

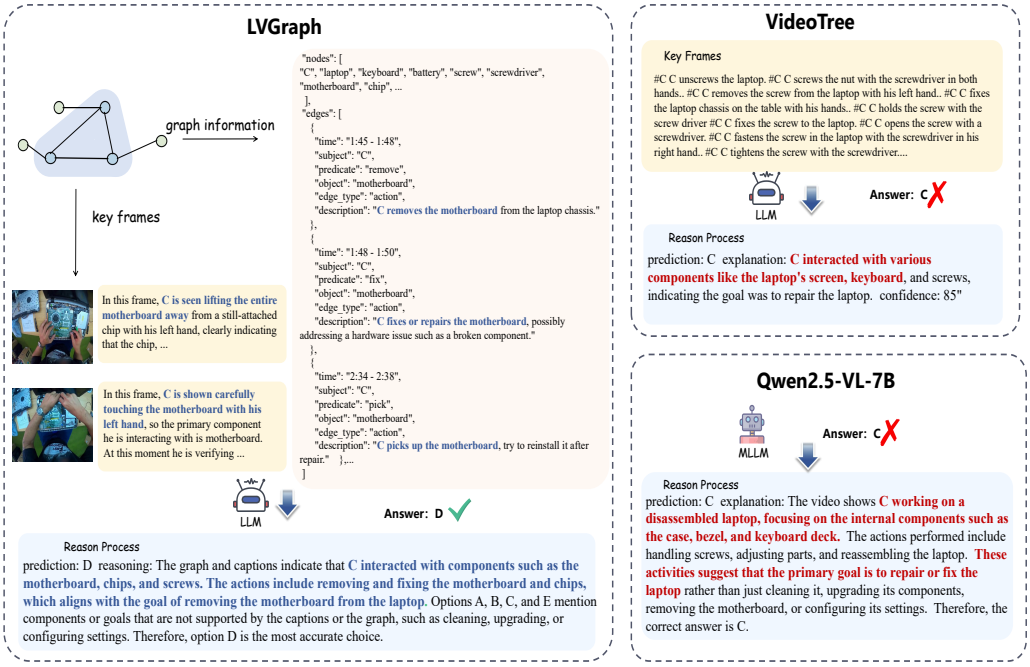


Figure 3: Qualitative examples of LVGraph.

importance varies across datasets. Overall, these results validate the effectiveness of jointly modeling semantic and structural information for multi-hop expansion.

LVGraph Components. We ablate each component of LVGraph to quantify its contribution, with results in Tab. 5. Removing the hierarchical graph construction module incurs the largest performance drop of 2.8%, underscoring the importance of this structure for modeling long-video content. Ablating the graph-based frame retrieval and keyframe enhancement modules degrades accuracy by 1.6% and 0.9% respectively. These results validate the effectiveness of our graph-driven keyframe selection and the utility of keyframe information enrichment.

Hyperparameter Analysis. In Tab. 8, we reports the effect of varying the number of top-K edges selected during graph search in LVGraph. The model achieves its best performance at K = 6. Retaining too many edges during retrieval introduces substantial noise and irrelevant information, which can hinder reasoning. This observation confirms the importance of moderate edge filtering for preserving information quality. In Tab. 7, we further examines the influence of the maximum number of hops (max hops) in the centrality-enhanced multi-hop expansion strategy. The model performs best when the hop limit is set to 2. A small hop limit may omit essential contextual information, whereas an excessively large hop count introduces redundant nodes and edges that interfere with the LLM's reasoning process, ultimately degrading overall performance. These findings reinforce our design motivation for long-video understanding: to provide the LLM with an efficient and compact graph representation that strengthens its multi-step reasoning capability.

4.3 Efficiency Analysis

Table 9 compares the inference efficiency and accuracy of LVGraph with existing methods, including VideoTree and LLoVi. LVGraph achieves an accuracy of 74.0%, outperforming LLoVi and VideoTree by 12.8 and 7.8 percentage points, respectively. In terms of inference efficiency, LVGraph reduces the time required for visual description generation to 2.1 seconds, compared with 22.4 seconds for LLoVi. This reduction allows more computation to be devoted to constructing structured representations rather than generating extensive visual descriptions.

Although graph construction introduces an additional overhead of 16.8 seconds, the resulting representation organizes video information into structured relationships, facilitating more effective evidence retrieval and multi-hop reasoning. This additional computation is therefore accompanied by a substantial improvement in reasoning accuracy. As a result, LVGraph achieves higher accuracy while maintaining a total inference time comparable to existing methods. These results demonstrate that LVGraph achieves a favorable balance between inference efficiency and reasoning accuracy by allocating computation to structured graph construction and reasoning.

4.4 Qualitative Comparison

In Fig. 3, we present qualitative results of LVGraph. Specifically, for a given video query, we demonstrate how video content is modeled using a hierarchical graph structure, along with the keyframes retrieved via the graph. The example is taken from the EgoSchema dataset, showing its query format, which includes a question and multiple candidate answers. In complex, multi-object video scenarios, LVGraph effectively captures the critical interactions between major objects in terms of actions and spatial relationships. By leveraging key nodes and edges in the graph, LVGraph accurately identifies the behavioral patterns of the main character C across temporally extended events and their interactions with core objects, clearly outperforming two baselines: the training-free VideoTree Wang et al. [2025d] and Qwen2.5-VL-7B Bai et al. [2025], which is trained on large-scale image and video datasets. These models often fail to localize key interacting objects and thus produce inaccurate or incomplete reasoning. In comparison, LVGraph integrates fine-grained semantic cues from keyframes to further infer the main intent of character C, ultimately enabling it to answer the given question correctly. Meanwhile, by leveraging more fine-grained semantic cues from keyframes, LVGraph further infers the primary intention of the target person C, ultimately enabling it to answer the given question correctly. In contrast, the training-free method VideoTree, when using 64 keyframes, struggles to grasp the overall semantics of the video and lacks effective modeling of fine-grained details, leading to incorrect answers and many redundant or irrelevant sampled frames. In comparison, our approach achieves accurate results using only 16 frames. More qualitative results are provided in the Appendix.

5 Conclusion

We propose LVGraph, a hierarchical graph-based reasoning strategy tailored for long video understanding. It allows large language models to maintain information consistency across extended temporal spans and tackle complex queries more efficiently. By building a multi-level graph from coarse to fine granularity, LVGraph structurally models event sequences, entity interactions, and temporal dependencies. It adaptively identifies key nodes and edges relevant to the query and maps them to corresponding keyframes. This design alleviates the redundancy and omissions inherent in linear traversal, enabling the model to progressively narrow its focus to the most relevant evidence across extensive video content. As a result, LVGraph supports robust long-range reasoning and enhanced fine-grained analysis. Extensive experiments show that LVGraph achieves state-of-the-art results across multiple benchmarks, significantly outperforming prior methods. With the continued advancement of visual and language models, LVGraph's structured approach offers a natural pathway to scaling, with strong potential to further advance the accuracy and scope of long video understanding.

6 Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 62550062, 62425606, 62576342, 32341009, and 62576338), the Beijing Natural Science Foundation (Grant Nos. L252145 and 4252054), and the Beijing Nova Program (Grant No. 202604841296).

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *ECCV*, 2024a.

Tieyuan Chen, Huabin Liu, Tianyao He, Yihang Chen, Chaofan Gan, Xiao Ma, Cheng Zhong, Yang Zhang, Yingxue Wang, Hui Lin, and Weiyao Lin. MECD: Unlocking multi-event causal discovery in video reasoning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 92554–92580. Curran Associates, Inc., 2024b. doi: 10.52202/079017-2939. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/a8320b6b9d95798dc286a867c44742a1-Paper-Conference.pdf`.

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

Jiajun Fei, Dian Li, Zhidong Deng, Zekun Wang, Gang Liu, and Hui Wang. Video-ccam: Enhancing video-language understanding with causal cross-attention masks for short and long videos. *arXiv preprint arXiv:2408.14023*, 2024.

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

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

Weiyu Guo, Ziyang Chen, Shaoguang Wang, Jianxiang He, Yijie Xu, Jinhui Ye, Ying Sun, and Hui Xiong. Logic-in-frames: Dynamic keyframe search via visual semantic-logical verification for long video understanding. *arXiv preprint arXiv:2503.13139*, 2025b.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. In *CVPR*, 2024.

Zeyi Huang, Yuyang Ji, Xiaofang Wang, Nikhil Mehta, Tong Xiao, Donghyun Lee, Sigmund Vanvalkenburgh, Shengxin Zha, Bolin Lai, Licheng Yu, et al. Building a mind palace: Structuring environment-grounded semantic graphs for effective long video analysis with llms. In *CVPR*, 2025.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *Mistral AI*, 2023. URL `https://arxiv.org/abs/2310.06825`.

Kumara Kahatapitiya, Kanchana Ranasinghe, Jongwoo Park, and Michael S Ryoo. Language repository for long video understanding. In *ACL*, 2025.

Jiahua Li, Kun Wei, Zhe Xu, Zibo Su, Xu Yang, and Cheng Deng. Perceive, reflect and understand long video: Progressive multi-granular clue exploration with interactive agents. *arXiv preprint arXiv:2509.24943*, 2025.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, 2024a.

Wei Li, Hehe Fan, Yongkang Wong, Mohan Kankanhalli, and Yi Yang. Topa: Extending large language models for video understanding via text-only pre-alignment. *NeurIPS*, 2024b.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for long-context video modeling. In *International Conference on Learning Representations*, volume 2026, pages 109089–109117, 2026.

Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *ECCV*, 2024c.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *EMNLP*, 2024.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge. Blog post, 2024. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`. Accessed: 2025-06-25.

Xiangrui Liu, Yan Shu, Zheng Liu, Ao Li, Yang Tian, and Bo Zhao. Video-xl-pro: Reconstructive token compression for extremely long video understanding. *arXiv preprint arXiv:2503.18478*, 2025.

Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. Drvideo: Document retrieval based long video understanding. In *CVPR*, 2025.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *NeurIPS*, 2023.

Juhong Min, Shyamal Buch, Arsha Nagrani, Minsu Cho, and Cordelia Schmid. Morevqa: Exploring modular reasoning models for video question answering. In *CVPR*, 2024.

Eric Nguyen, Karan Goel, Albert Gu, Gordon Downs, Preey Shah, Tri Dao, Stephen Baccus, and Christopher Ré. S4nd: Modeling images and videos as multidimensional signals with state spaces. *NeurIPS*, 2022.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. URL `https://arxiv.org/abs/2303.08774`.

OpenAI. GPT-4o Mini: Advancing Cost-Efficient Intelligence. `https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/`, 2024. Accessed on 2025-11-14.

OpenAI. Gpt-5, 2025. `https://openai.com/research/gpt-5`.

Jongwoo Park, Kanchana Ranasinghe, Kumara Kahatapitiya, Wonjeong Ryu, Donghyun Kim, and Michael S Ryoo. Too many frames, not all useful: Efficient strategies for long-form video qa. *arXiv preprint arXiv:2406.09396*, 2024.

Kanchana Ranasinghe, Xiang Li, Kumara Kahatapitiya, and Michael S Ryoo. Understanding long videos in one multimodal language model pass. *arXiv preprint arXiv:2403.16998*, 2024.

Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. In *CVPR*, 2025.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *CVPR*, 2024.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *CVPR*, 2025.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

Shaoguang Wang, Ziyang Chen, Yijie Xu, Weiyu Guo, and Hui Xiong. Less is more: Token-efficient video-qa via adaptive frame-pruning and semantic graph integration. *arXiv preprint arXiv:2508.03337*, 2025a.

Xiao Wang, Qingyi Si, Shiyu Zhu, Jianlong Wu, Li Cao, and Liqiang Nie. Adaretake: Adaptive redundancy reduction to perceive longer for video-language understanding. In *ACL*, 2025b.

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

Xidong Wang, Dingjie Song, Shunian Chen, Junyin Chen, Zhenyang Cai, Chen Zhang, Lichao Sun, and Benyou Wang. Longllava: Scaling multi-modal llms to 1000 images efficiently via a hybrid architecture. *arXiv preprint arXiv:2409.02889*, 2024c.

Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025c.

Ying Wang, Yanlai Yang, and Mengye Ren. Lifelongmemory: Leveraging llms for answering queries in long-form egocentric videos. *arXiv preprint arXiv:2312.05269*, 2023.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. In *CVPR*, 2025d.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. Longvlm: Efficient long video understanding via large language models. In *ECCV*, 2024.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021.

Yuxuan Yan, Shiqi Jiang, Ting Cao, Yifan Yang, Qianqian Yang, Yuanchao Shu, Yuqing Yang, and Lili Qiu. Empowering agentic video analytics systems with video language models. *arXiv preprint arXiv:2505.00254*, 2025.

Jinhui Ye, Zihan Wang, Haosen Sun, Keshigeyan Chandrasegaran, Zane Durante, Cristobal Eyzaguirre, Yonatan Bisk, Juan Carlos Niebles, Ehsan Adeli, Li Fei-Fei, et al. Re-thinking temporal search for long-form video understanding. In *CVPR*, 2025.

Huaying Yuan, Zheng Liu, Minghao Qin, Hongjin Qian, Yan Shu, Zhicheng Dou, Ji-Rong Wen, and Nicu Sebe. Memory-enhanced retrieval augmentation for long video understanding. *arXiv preprint arXiv:2503.09149*, 2025.

Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, et al. Timesuite: Improving mllms for long video understanding via grounded tuning. *ICLR*, 2025.

Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple llm framework for long-range video question-answering. *EMNLP*, 2023.

Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *CVPR*, 2023.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024.

Jialong Zuo, Yongtai Deng, Lingdong Kong, Jingkang Yang, Rui Jin, Yiwei Zhang, Nong Sang, Liang Pan, Ziwei Liu, and Changxin Gao. Videolucy: Deep memory backtracking for long video understanding. *NeurIPS*, 2025.

Appendix

In these supplementary materials, we present the following: additional quantitative results (Sec. A), the detailed algorithm (Sec. B), comparison of inference efficiency and accuracy (Sec. C), additional implementation details (Sec. D), additional qualitative analysis (Sec. E) and limitations(Sec. F).

A Additional Quantitative Results

A.1 Comparison with RAG Methods.

In Tab. 10, we present a detailed comparison between LVGraph and state-of-the-art retrieval-augmented generation (RAG) methods on the MLVU and VideoMME benchmarks. Unlike Video-RAG, which depends on external modules such as object detection and OCR for frame-level information extraction, our method constructs hierarchical graphs and performs reasoning using only VLMs and LLMs. This design offers greater flexibility, and by leveraging the capabilities of strong vision and language models, our approach achieves significantly higher reasoning accuracy and generates more reliable answers.

Table 10: RAG methods comparison.

Method	Size	MLVU	VideoMME
Open-Source Models			
Video-RAG	3B	62.2	60.3
Video-RAG	7B	63.4	60.5
Proprietary Models			
VideoTree	-	60.4	60.6
LVGraph	-	**65.3**	**70.4**

Table 11: Comparison of different MLLM methods on EgoSchema.

Method	VLM	Acc.	Frames
Proprietary MLLM			
Gemini 1.5-Flash	-	65.7	180
Gemini 1.5-Pro	-	70.2	16
Gemini 1.5-Pro	-	71.2	180
GPT-4o	-	72.2	180
Open-Source MLLM			
LLaVA-OneVision-72B	-	62.0	32
InternVideo2.5	-	63.9	180
Qwen2.5-VL-7B	-	65.0	180
Qwen2.5-VL-72B	-	76.2	180
Training-Free LLM Agent			
Ours	LaViLa	**74.0**	**16**

A.2 Comparison with advanced VideoLLMs on EgoSchema and NExT-QA.

Comparison on EgoSchema. As demonstrated in Tab. 11, our training-free method achieves a compelling accuracy of 74.0% on the EgoSchema benchmark using merely 16 frames. This result not only surpasses the proprietary Gemini 1.5-Pro by a significant margin of 3.8% under an identical few-frame setting, but also outperforms GPT-4o by 1.8%, despite the latter leveraging over 10x

Table 12: Comparison of different MLLM methods on NExT-QA.

Method	Acc.
Open-Source MLLM	
Tarsier-34B	79.2
VideoChat2	61.7
Training-Free LLM Agent	
Ours	**79.9**

Algorithm 1 LVGraph

Require: Video frames V, query Q, number of clips k, number of top edges K, LLM F_{llm}, VLM (Captioner) F_{vlm}, Embedding Model E_{emb}, Prompts P_{G1}, P_{G2}, P_{cap}

$Clips \leftarrow SegmentVideo(V, k)$ ▷ Hierarchical Graph Construction
$Clips_{caption} \leftarrow CaptionClips(F_{vlm}, Clips)$
$G_1 \leftarrow VGraph(F_{llm}, Clips_{caption}, P_{G1})$ ▷ Construct coarse-grained global graph
$G_2 \leftarrow VQGraph(F_{llm}, G_1, Q, P_{G2})$ ▷ Extract query-relevant subgraph

$Q_p \leftarrow ParseQuery(Q)$ ▷ Graph-Based Frame Retrieval
$seed_edges \leftarrow FindSeedEdges(G_2, Q_p, K, E_{emb})$
$seed_nodes \leftarrow GetNodesFromEdges(seed_edges)$
$key_edges \leftarrow ExpandNodes(seed_nodes, G_2, Q)$ ▷ Centrality-enhanced multi-hop expansion
$Frames_{keys} \leftarrow GetFramesFromEdges(key_edges)$

$Caps_{enhanced} \leftarrow []$ ▷ Keyframe Information Enhancement
for each $f_i \in Frames_{keys}$ **do**
    $detailed_cap \leftarrow F_{vlm}(Q, f_i, P_{cap})$
    $Caps_{enhanced}.append(detailed_cap)$
end for

$pred_answer \leftarrow LLMReasoning(F_{llm}, G_2, Caps_{enhanced}, Q)$ ▷ Final Answer Generation
return $pred_answer$

more frames. When compared with open-source MLLMs, our approach exhibits a superior trade-off between performance and efficiency. While the 72B-parameter Qwen2.5-VL attains a higher accuracy, it relies on a dense input of 180 frames. In stark contrast, our method delivers highly competitive performance with a substantially lower computational budget, underscoring its scalability and practical value in resource-constrained scenarios.

Comparison on NExT-QA. As shown in Tab. 12, on the NExT-QA benchmark, our proposed training-free method achieves an accuracy of 79.9%, surpassing all existing open-source MLLMs. Specifically, our approach outperforms the current best open-source model Tarsier-34B by 0.7%, while exceeding VideoChat2 by a significant margin of 18.2%.

B Detailed Algorithm

In Algorithm 1, we present further details of the algorithm behind LVGraph. Definitions of the relevant notations have been provided in the main text.

C Comparison of inference efficiency and accuracy

Table 13 compares the per-video inference efficiency and accuracy of LVGraph with three existing methods. LVGraph achieves the highest accuracy of 72.2%, outperforming LLoVi, VideoTree, and DrVideo by 11.0, 6.0, and 5.8 percentage points, respectively. Meanwhile, it requires only

Table 13: Comparison of inference efficiency and accuracy.

Method	Inference VLM Calls	LLM Calls	Tokens	Wall-Clock Time (s)	Acc. (%)
LLoVi	180	1	2.1k	24.8	61.2
VideoTree	91	3.2	12.2k	20.1	66.2
DrVideo	114	3.4	4.9k	19.3	66.4
LVGraph	87	3.1	5.8k	22.5	**72.2**

87 VLM calls per video, representing reductions of 51.7%, 4.4%, and 23.7% compared with the three baselines, respectively. This efficiency stems from its hierarchical processing strategy, which samples approximately 180 frames at 1 FPS, groups every four frames into about 45 coarse clips, and selectively allocates 26 VLM calls to fine-grained graph construction and 16 calls to detailed evidence-frame captioning. LVGraph also consumes 5.8k tokens per video, less than half of VideoTree's 12.2k tokens. Although its wall-clock time of 22.5 seconds is slightly longer than those of VideoTree and DrVideo, it remains faster than LLoVi, which takes 24.8 seconds. These results demonstrate that LVGraph achieves substantially better video understanding accuracy with fewer VLM calls and moderate inference overhead, highlighting the effectiveness of hierarchical graph construction in balancing reasoning performance and inference cost.

D Additional Implementation Details

Prompt Details. We provide detailed prompts for graph construction in Tab. 14, prompt for obtaining subgraph in Tab. 15, prompt for keyframe information enhancement in Tab. 16, and prompt for LLM reasoning in Tab. 17.

Additional LVGraph Implementation Details. We preprocess the video as follows: for the EgoSchema and NExT-QA benchmarks, we uniformly sample the original videos at a rate of 1 FPS. For the short and medium split of Video-MME, we also sample at 1 FPS, while for the long split, we use a sampling rate of 0.2 FPS. For the MLVU benchmark, we sample the original frames at 0.2 FPS. During hierarchical graph construction, we define clips of k_1 frames for generating the coarse-grained graph and clips of k_2 frames for the fine-grained graph. Specifically, we set $k_1 = 4$ and $k_2 = 1$ for EgoSchema, NExT-QA, and the short and medium split of Video-MME. For the long split of Video-MME, we use $k_1 = 6$ and $k_2 = 1$, and for MLVU, we set $k_1 = 5$ and $k_2 = 1$.

E Additional Qualitative Analysis

E.1 Additional Case Study.

In Fig. 4, we present another visualization result of LVGraph on the VideoMME dataset. The video narrates the development of general relativity, its observational evidence, and the history of its scientific dissemination. By analyzing the graph structure, we accurately identified nodes and edges related to "general relativity" and further localized the key frames. Using VLMs, we extracted more relevant information from these key frames, ultimately leading to the correct answer.

E.2 Failure Case.

In Fig. 5, we provides a qualitative visualization of a failure case. We observe that although the graph structure and key frame information contained accurate and sufficient clues to answer the question, the LLM incorrectly categorized the activity involving cards and dice as "matching and classifying games." This case highlights the current limitations of VLM and LLM in complex scenario understanding. We anticipate that more powerful multimodal reasoning models in the future will help resolve such failures.

F Limitations

Like other LLM-based video reasoning systems, our framework is intrinsically linked to the capabilities of its constituent models: the VLM and the LLM. First, the fidelity of the VLM's perceptual

Table 14: Prompt for graph construction

You are given a set of clip-level textual descriptions {captions} extracted from a long video. Your task is to construct a global temporal entity–relation graph that summarizes all key entities and their interactions across the entire video. Each clip description contains actions, objects, spatial relations, and your goal is to transform them into a coherent set of nodes and edges. You must process the description lines strictly in the order they appear. All original frame numbers appearing after #C or #O must be preserved exactly as given, and you must not skip, renumber, or reorder events.

When multiple consecutive lines contain the same (actor; action or spatial predicate; object) triple, you must merge them into a single edge whose frame interval spans from the first occurrence to the last. When a triple changes or is interrupted, you must begin a new edge. Nodes must correspond to concrete entities that appear at least twice across the entire collection of clips, including persons, objects, game components, or environmental elements. Each edge must include its subject, predicate, object, frame or frame interval, and an edge_type, which can be either "action" or "spatial". The edge must also include a concise natural-language description explaining the contextual significance of the interaction or spatial relation.

Your final output must be a JSON object containing two fields: a list of nodes and a list of edges. The nodes field must contain all entities that meet the frequency requirement. The edges field must contain temporal edge entries in the order they appear in the source clips, with each entry containing its frames, subject, predicate, object, edge_type, and description.

Below is an example of the required JSON output format:
Examples
Now generate the global graph for the given clip descriptions.

Table 15: Prompt for obtaining subgraph

You are given a global temporal entity–relation graph G1 {graph} that summarizes a long video and your task is to extract a question-relevant subgraph G2 based on a question Q {Q}.

You must interpret the question Q and identify all nodes and edges that are semantically relevant to the question. Relevance may come from direct mention, implied relationships, or contextual alignment between the question and the edge descriptions. Your reasoning should consider the subject, predicate, object, textual description, and edge_type of each edge within G1. You must include an edge in G2 if the edge itself or any of its participating entities is relevant to the question. When an edge is selected, all entities involved in that edge must be included in the final node list of G2.

You must preserve the exact frame or frame interval strings from G1 without modifying any values. The original temporal order of edges must also be preserved. You must not add new edges or nodes that do not already exist in G1. You must not infer new relations, just only extract relevant ones.

The final output must be a JSON object containing two fields: "nodes" and "edges". The nodes list includes all unique entities present in the selected edges. The edges list contains all edges relevant to the question, expressed in the same JSON structure as in G1, preserving all fields, textual descriptions, ordering, and frame information.

Below is an example of the expected input graph format and the output subgraph :
Examples
Now, using the above rules, extract the subgraph G2 that is relevant to the question Q, and output it in the same JSON format with fields "nodes" and "edges".

analysis presents a critical performance bottleneck. Any inaccuracies, such as hallucinations or descriptive omissions at the frame level, will propagate errors through the subsequent pipeline. This process can corrupt the structural integrity and semantic consistency of the graph, ultimately leading to erroneous inferences. Furthermore, the system's performance is contingent on the LLM's reasoning prowess and its practical constraints. Faulty logical inferences can degrade accuracy, while architectural limitations, specifically the finite context window and inference latency, restrict the model's scalability to long videos and its overall computational efficiency. Nevertheless, owing to the

Table 16: Prompt for keyframe information enhancement

You are given a single video keyframe image and a question Q {Q}. Your task is to generate a detailed, fine-grained caption for the keyframe that focuses specifically on visual details relevant to answering the question. The caption must not be generic or high-level; instead, it must extract and articulate the most precise and question-relevant visual information present in the image. You should closely examine colors, textures, spatial relationships, small objects, fine actions, attributes, and any subtle details that could contribute to answering the question.

You must not invent or hallucinate information that cannot be visually verified in the image. If the image does not contain the detail that the question asks for, describe exactly what can and cannot be seen. You must describe visual evidence rather than making assumptions.

Your caption should integrate awareness of the question Q so that the resulting description is tailored to the specific aspects that matter for answering that question. If Q mentions an entity, ensure that entity is described thoroughly in terms of appearance, position, orientation, interaction, and local context. If multiple entities are involved, describe their relationships, relative positions, and roles within the scene. If the question references color, number, shape, identity, or state, emphasize these attributes where visible.

The final caption must be a single coherent paragraph that provides rich, fine-grained, visually grounded information which an LLM could use to answer Q accurately. Only describe what is actually visible in the frame.

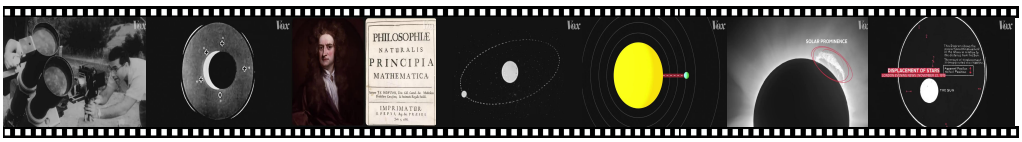
Below is an example illustrating the expected style of detailed, question-aware captioning:
Examples

Table 17: Prompt for LLM reasoning

You are given a temporal entity–relation graph {graph} and a set of keyframe captions {key_frames}, both providing the visual and semantic evidence needed to answer the following question: {question}. Your task is to determine the correct answer by relying strictly on the information contained in the graph and the captions. If no option is entirely correct, you should evaluate each remaining option according to how closely it aligns with the available evidence, assign a confidence score from 1 to 3 to indicate its degree of correctness, and then select the option with the highest score. Your final output must follow the format “prediction:\n reasoning:\n”.

Below is an example :
Examples

modular design of our method, improvements in VLMs and LLMs can be directly integrated into the LVGraph framework to enhance model performance.



Question: Which specific phenomenon during the video demonstrated the correctness of general relativity?

A. The existence of gravitational waves.
B. Light bends when it passes through the gravitational field of the sun.
C. The gravitational redshift phenomenon.
D. The existence of black holes.

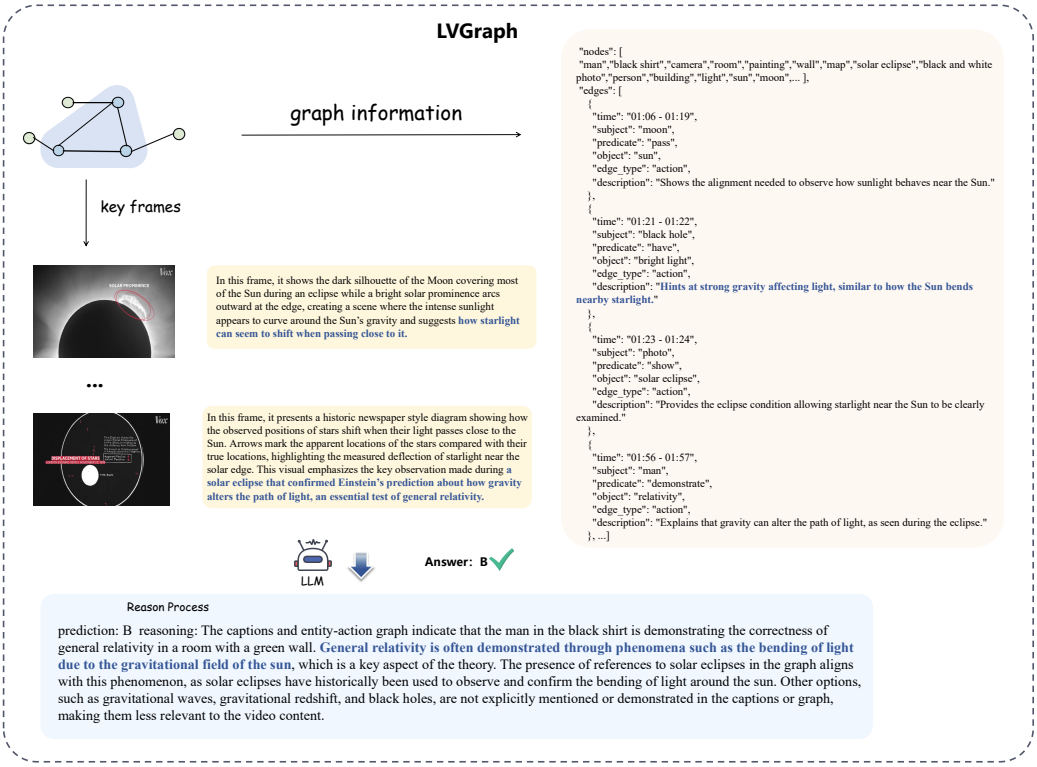


Figure 4: Qualitative examples of LVGraph.



Question: In this video, there are two primary activities involving cards and dice. summarize these activities and how they are related.

A.In gaming, the two primary activities frequently involving cards and dice consist of rolling and throwing actions.
B.In recreational games, the two primary activities involving cards and dice typically consist of counting and adding numbers.
C.In gaming, the two primary activities involving playing cards and dice are typically matching and sorting for various games.
D.The two primary activities involving cards and dice are drawing and painting.
E.The two primary activities involving cards and dice are shuffling and dealing.

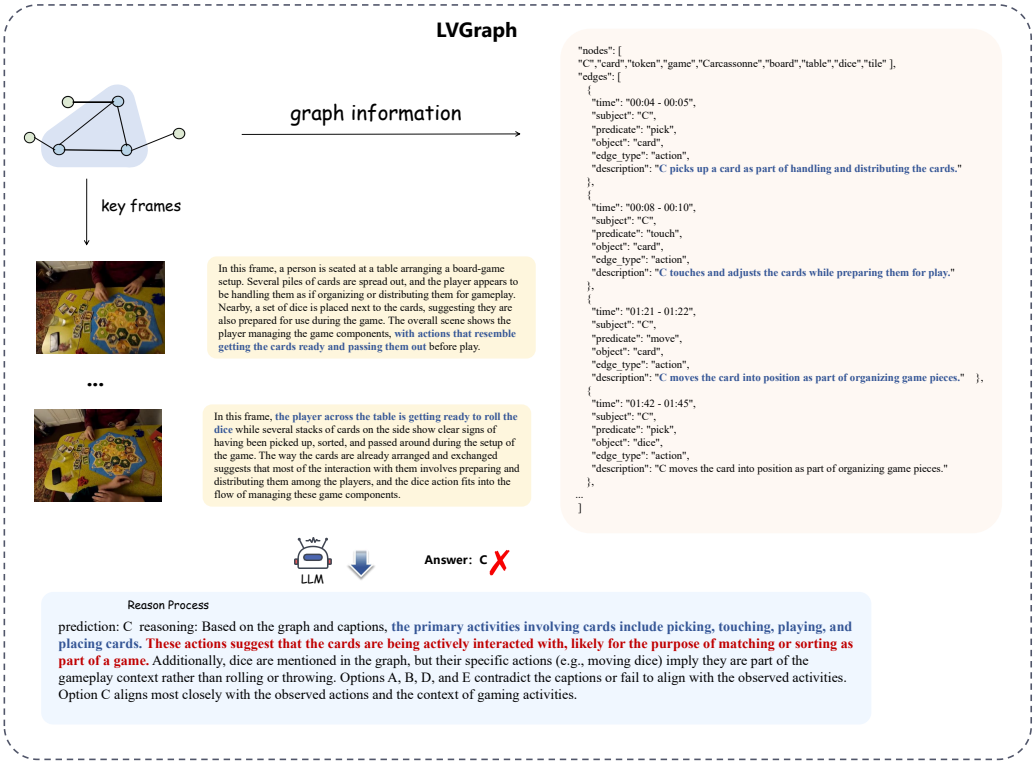


Figure 5: Failure case of LVGraph.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We describe it in Section 3

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We describe it in Section F

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: Without theory assumptions and proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We describe it in Section 4

   Guidelines:

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

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

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

Answer: [No]

Justification: The data and code used in our experiments are currently under review and contain proprietary elements, which restricts their immediate open access. However, we plan to make the code available upon publication.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We describe it in Section 4

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Our results are not accompanied by error bars, confidence intervals, or statistical significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We describe it in Section 4

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We confirm, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This work presents a general-purpose methodological framework for long-video understanding and is evaluated on existing public benchmarks. It does not target specific societal applications, involve surveillance deployment, or perform identity recognition or sensitive-attribute inference. Therefore, we do not identify direct societal impacts within the scope of this work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: No such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All existing assets such as code, data, and models used in our research are properly credited to their respective creators or owners. We have explicitly mentioned the licenses and terms of use for each asset in our documentation. Additionally, we have ensured that all terms of use are strictly adhered to, including obtaining necessary permissions for assets with restrictive licenses. This practice supports ethical research standards and ensures legal compliance in the use and distribution of third-party resources.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: No new assets released.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No human subjects participated.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: We describe it in Section 4

    Guidelines:

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