Accelerating Long Video Understanding via Compressed Scene Graph-Enabled Chain-of-Thought

Tao Ling
The Hong Kong Polytechnic University
Hong Kong SAR, China
cstling@comp.polyu.edu.hk

Siping Shi
The Hong Kong Polytechnic University
Hong Kong SAR, China
si-ping.shi@connect.polyu.hk

Dan Wang*
The Hong Kong Polytechnic University
Hong Kong SAR, China
csdwang@comp.polyu.edu.hk

Abstract

Long video understanding, which leverages Video-LLMs to analyze and interpret extended video content to extract meaningful information, insights, or summaries, is a fundamental task in multimedia domain. Chain-of-thought (CoT) methods are widely adopted to enhance long video understanding by incorporating intermediate reasoning steps. However, the iterative nature of CoT—requiring a lengthy sequence of internal thoughts—significantly increases the latency of video object description. To address this challenge, we design Compressed Scene Graph-enabled CoT (*CSGCoT*), a novel approach that facilitates efficient and accurate long-video object description. Inspired by video codec principles, we propose a compressed scene graph composed of two components: Key-SG for key frames and Delta-SG for delta frames, enabling efficient encoding of scene information across video segments. Specifically, *CSGCoT* comprises three major modules: (1) a Key-SG Detector that identifies representative segments, (2) a Delta-SG Generator that produces compensated representations for delta segments, and (3) a SG-Query Manager that converts scene graphs into natural language prompts for video object description. Experiments show that *CSGCoT* achieves comparable accuracy to SOTA methods while reducing latency by over 62.6% on hour-long videos while maintaining comparable accuracy.

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**.

Keywords

Long Video Understanding, Chain-of-Thought, Scene Graph

ACM Reference Format:
Tao Ling, Siping Shi, and Dan Wang. 2025. Accelerating Long Video Understanding via Compressed Scene Graph-Enabled Chain-of-Thought. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746027.3754765

*Corresponding author.

MM '25, Dublin, Ireland.

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3754765

1 Introduction

In recent years, the advancement of Large Language Models (LLMs) has significantly transformed the landscape of artificial intelligence, with video-LLMs emerging as a promising frontier in multimodal understanding, particularly for complex multimedia tasks involving long-form video content analysis and comprehension [21]. As videos increase in length and complexity, such as hour-long videos with distinct storylines, the demands on computational resources and reasoning capabilities grow substantially. The temporal dimension of videos introduces unique challenges that image understanding does not face, particularly in maintaining coherent understanding across extended time periods [22, 48]. These challenges have motivated researchers to leverage sophisticated reasoning approaches, such as Chain-of-Thought (CoT) and agent-based methods, to enhance the performance of Video-LLMs in complex understanding tasks [37].

For long video understanding tasks, chain-of-thought has been widely adopted due to its ability to manage the high complexity and temporal dependencies inherent in extended video content. CoT enables Video-LLMs to break down reasoning tasks into a series of intermediate steps, making the reasoning process more transparent and often more accurate [4][31].

However, the iterative nature of CoT—requiring a lengthy sequence of internal thoughts—significantly increases the latency of video understanding. Especially for complex long videos, to maintain the performance of video understanding, the CoT-enhanced video-LLM requires sampling and captioning more frames or segments, resulting in unsatisfactory latency. We demonstrate our motivation through two initial experiments. Following common categorizations in the literature [19, 24, 41], we measure three typical video understanding tasks, i.e., Object Description (OD), Contextual Retrieval and Detection (CRD), and Temporal Grounding (TG), on the ALLVB dataset [19], with the widely used CoT (MCTS-CoT [36]) and video-LLM (VideoLLama-2 [2]).

Fig. 1 illustrates the variation in the performance of the CoT-enhanced video-LLM under different sampling segments per video (SPV) settings. As the SPV increases, the inference latency grows from 88.26s to 1310.32s. Despite this substantial computational investment, task performance shows mixed results: while CRD and TG tasks demonstrate steady precision improvements with increased sampling (from 33.45% to 37.9% for CRD and from 21.5% to 25.7% for TG), OD tasks counterintuitively exhibit a precision decrease from 37.34% to 34.8%, contradicting the expected gains from increased sampling density.

While increased sampling can improve accuracy on reasoning tasks, it leads to decreased efficiency in describing objects accurately

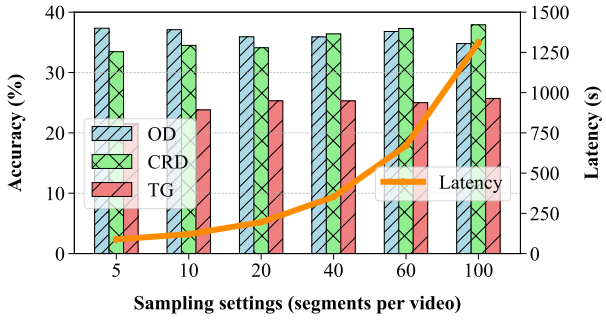

Figure 1: Impact of sampling settings on task accuracy and response latency.

Table 1: Repetitive content distribution across different modules inference.

Module	OD-related	CRD-related	TG-related
Video-LLM	73.2%	17.8%	9.0%
CoT-LLM	52.3%	24.4%	23.3%

throughout lengthy videos. To further investigate this phenomenon, we conducted additional analysis on repetitive content distribution (count with tokens). In Tab. 1, we classified and analyzed the distribution of repetitive tokens generated during inference across different modules. The results show that in standard Video-LLM processing, object description-related tokens account for a substantial 73.2% of repetitive content, compared to only 17.8% for contextual retrieval and 9.0% for temporal grounding. Although the CoT-LLM demonstrates a more balanced distribution, the high proportion of redundant object descriptions still indicates inefficient token utilization, contributing to both increased latency and potential performance degradation. The underlying reason is that repeated descriptions of the same objects are generated across each sampled segment. Videos lengthen and CoT chains expand, redundant object descriptions consume an excessive portion of the limited token window, causing LLMs to forget previously important descriptive information and slowing inference speed.

To address these challenges, the key solution lies in compressing object descriptions to reduce redundancy while preserving critical relational information between objects across video segments. Scene graphs (SGs) function as structured representations that capture objects, attributes, and relationships in a compact form [44]. These representations have previously demonstrated effectiveness in enhancing CoT for image tasks [27]. Drawing inspiration from keyframe concepts in video encoding, we can apply similar principles to object representation in long videos by storing complete information at key points and only changes in between (See in Fig. 2). Based on these insights, we propose Compressed Scene Graph-enabled Chain-of-Thought (*CSGCoT*), an approach that leverages temporally encoded SGs to enhance CoT-assisted Video-LLMs. Our approach represents key temporal segments with complete SGs, while intermediate segments are described using differential SGs that only capture changes relative to the keyframes—a technique

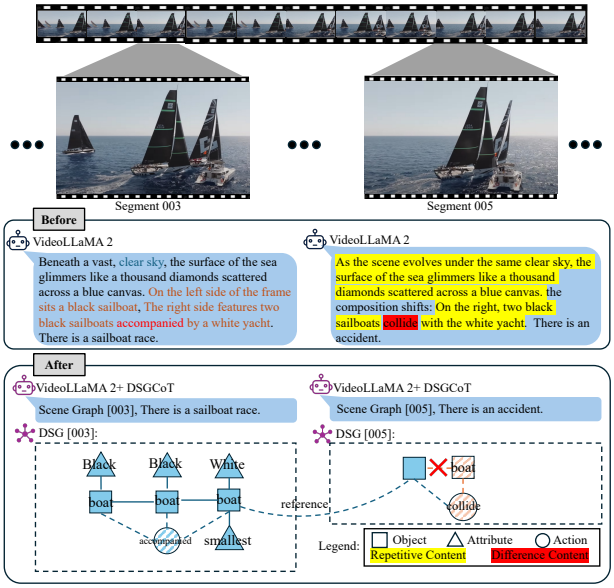

Figure 2: Compressed Scene Graph Example for representing repetitive content.

we term compressed SG. This approach reduces token usage by structuring object descriptions and facilitating efficient comparison between consecutive segments. *CSGCoT* comprises three modules: (1) **Key Scene Graph Detector** identifies critical segments and generates comprehensive SGs, (2) **Delta Scene Graph Generator** produces compact change-focused graphs for non-key segments, and (3) **Scene Graph Query Manager** converses object descriptions from the SG Memory during reasoning.

Our main contributions are:

- A novel compressed scene graph representation specifically designed for long video understanding, which efficiently captures object relationships while minimizing redundancy across temporal segments.
- Based on this compressed representation, we develop *CSGCoT*, a chain-of-thought framework that reduces inference latency while maintaining reasoning accuracy for complex video understanding tasks.
- Experiments show that *CSGCoT* achieves comparable accuracy to SOTA methods while reducing latency by over 62.6% on hour-long videos, making advanced video understanding more accessible across diverse computing environments.

2 Related Work

2.1 Multimodal LLM in Long Video Understanding

Multimodal LLM (MLLM) for video understanding typically accept both textual queries and video sequences as inputs, producing textual outputs. With the emergence of hour-scale video datasets [15, 30, 35], research has expanded into long video understanding, characterized by content spanning from 10 minutes to several hours, containing multiple scenes.

To address these extended temporal horizons, researchers have proposed hierarchical encoding approaches [43][25] and memory-augmented architectures [32][26] for MLLM. The fundamental challenges extend from understanding visual elements to reasoning about contextual causality and inferring temporal relationships between key events.

2.2 Multimodal Chain-of-Thought

The challenge in object description for long videos stems from the massive volume and redundancy of visual information, necessitating efficient methods to identify and extract question-relevant visual details. Multimodal Chain-of-Thought (MCoT) and agent-based approaches mimic human cognitive behaviors [1, 20, 28, 42].

From the rationale construction perspective [28], Multimodal Chain-of-Thought (MCoT) reasoning can be categorized into three approaches: (1) prompt-based methods [5, 7, 16, 26, 33, 34] that utilize carefully designed prompts to guide models in generating step-by-step reasoning without model modifications; (2) plan-based methods [9, 39, 45] that enable models to dynamically explore multiple reasoning paths simultaneously; and (3) learning-based methods [14, 18, 23, 40, 46] that embed rationale construction directly within the training or fine-tuning process. Our work addresses efficiency issues in object descriptions within MCoT and benchmarks against comparable approaches, both falling within the prompt-based methodology category.

2.3 Scene Graph in Vision Tasks

Scene graph are structured representations that explicitly model objects, attributes, and relationships within visual scenes [10, 38], originally developed for image retrieval [27] and generation tasks [11]. They provide a compact, symbolic encoding of visual content that bridges the gap between raw visual features and natural language understanding. With the advancement of multimodal LLMs, SGs were initially employed to enhance LLM comprehension of object relationships in static images, as demonstrated in works like PFVR [12] and RelViT [13]. CCoT [16] leverages SGs to guide step-by-step reasoning for image QA tasks.

As video understanding research has progressed, SG applications have extended to Video-LLMs, primarily focusing on accuracy improvements. Some works include STEP [17], which extends SGs to video for the fine-grained visual relations, and VoT [5], which introduces spatio-temporal SGs for video reasoning. Our work represents the first application of SGs specifically designed to enhance reasoning efficiency in video understanding tasks.

3 Method

We propose Compressed Scene Graph-enabled Chain-of-Thought (*CSGCoT*), a framework designed to enhance the efficiency of Video-LLMs CoT reasoning in long video understanding tasks. As illustrated in Fig. 3, *CSGCoT* integrates into existing CoT-enhanced Video-LLM frameworks. The conventional workflow involves video segmentation, caption generation through Video-LLM, memory storage of textual descriptions, and CoT reasoning for information retrieval. *CSGCoT* transforms this process by implementing a structured SG representation system. Following video segmentation, the **Key Scene Graph Detector** identifies critical segments and generates comprehensive SGs, while the **Delta Scene Graph Generator** creates compact differential representations for remaining segments. The **Scene Graph Query Manager** handles storage and retrieval during CoT reasoning, converting complete descriptions when necessary for final output generation.

3.1 Key Scene Graph Detector

The Key Scene Graph Detector (KSGD) identifies semantically significant video segments and generates comprehensive SGs for key moments. KSGD employs a selective strategy inspired by video keyframe encoding techniques, focusing on segments with significant visual information or state changes.

3.1.1 Key Segment Detection. Given a video V segmented into n temporal segments $S = \{s_1, s_2, ..., s_n\}$, KSGD evaluates segment significance using a dual-criteria mechanism:

$$K(s_i) = \alpha \cdot I(s_i) + (1 - \alpha) \cdot C(s_i, s_{i-1}) \tag{1}$$

where $I(s_i)$ represents the information density within segment s_i, $C(s_i, s_{i-1})$ quantifies the visual state change from the previous segment, and $\alpha(s_i)$ is an adaptive weight determined by:

$$\alpha(s_i) = \sigma\left(\frac{\bar{C}}{\bar{I}} \cdot \frac{I(s_i)}{C(s_i, s_{i-1}) + \epsilon}\right) \tag{2}$$

where $\bar{I}$ and $\bar{C}$ represent the mean information density and change metrics across all segments, ϵ is a small constant to prevent division by zero, and $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function that constrains $\alpha(s_i)$ to the range $(0, 1)$. This formulation automatically assigns higher weight to information density when it is significantly more prominent than visual change, and vice versa.

For information density estimation, we utilize the Video-LLM's captioning capabilities to quantify visual richness:

$$I(s_i) = \frac{1}{|O_i|} \sum_{o \in O_i} \text{Conf}(o, s_i) \tag{3}$$

where O_i represents the set of objects detected in segment s_i, and $\text{Conf}(o, s_i)$ measures the confidence score for object o in segment s_i based on the Video-LLM's captioning output.

The change metric $C(s_i, s_{i-1})$ is computed by comparing caption embeddings:

$$C(s_i, s_{i-1}) = 1 - \cos(e(s_i), e(s_{i-1})) \tag{4}$$

where $e(s_i)$ denotes the embedding of the caption for segment s_i generated by the Video-LLM, and $\cos(\cdot)$ represents cosine similarity.

3.1.2 Key Scene Graph Generation. Once a key segment is identified, KSGD generates a complete SG (key SG) representation $G_k = (V_k, E_k, A_k)$, where $V_k = \{v_1, v_2, ..., v_m\}$ represents object nodes; $E_k = \{e_{ij}\}$ represents relation edges between objects v_i and v_j; $A_k = \{a_{ij}\}$ represents attribute nodes.

The SG construction employs a structured prompt template P_{SG} implemented as:

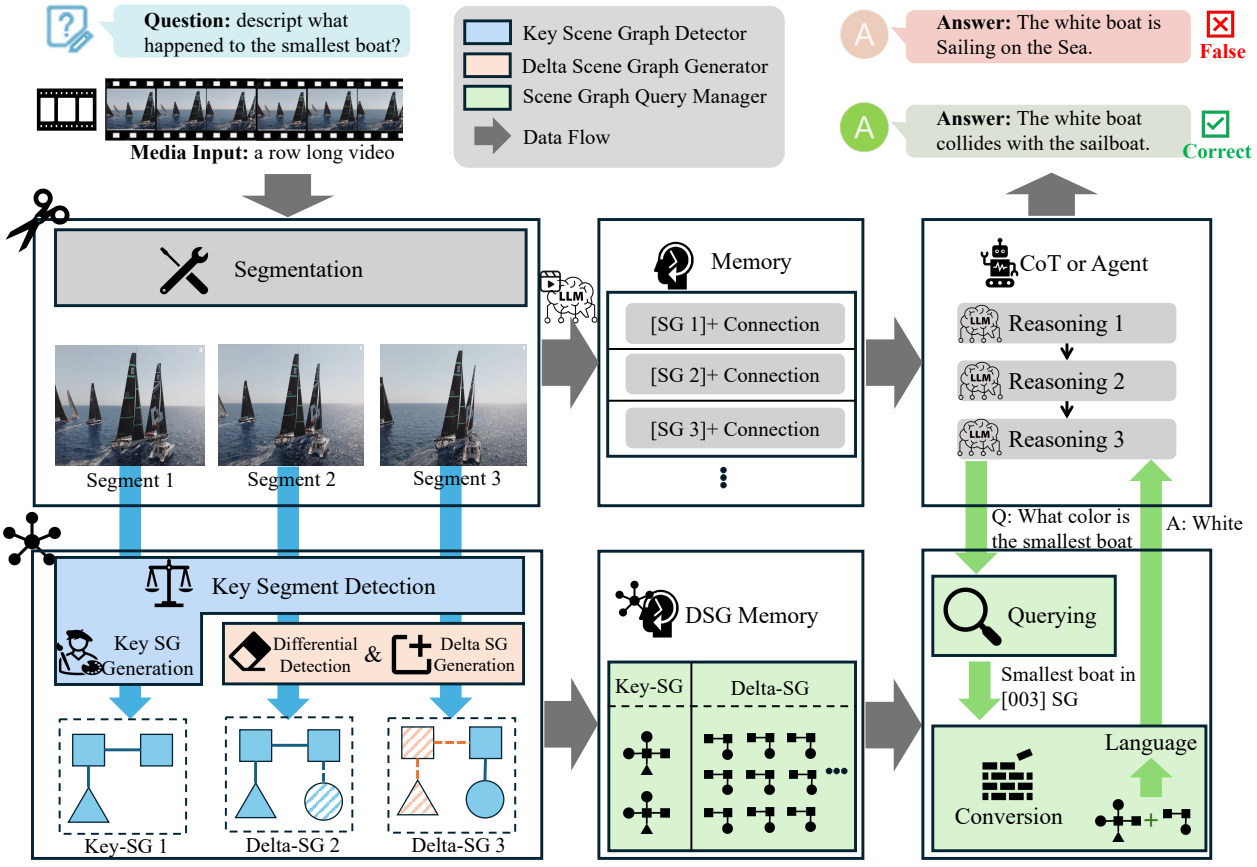

Figure 3: Overview of the Compressed Scene Graph in general Video-LLM CoT.

P_{SG}: Given the video segment $\{s_i\}$, generate a structured SG with the following components:
① Background: List all background elements and their attributes;
② Foreground: List all foreground objects and their attributes;
③ Actions: Describe relationships and interactions between objects.
Please ensure adherence to constraints C_{comp} and C_{cons}.

where two exemplar constraints are:

C_{comp}: The graph must capture all visually significant elements in the scene.
C_{cons}: Use consistent naming conventions for objects that appear across multiple segments.

The implementation of the SG generation function can be formulated as:

$$G_k = \text{VideoLLM}(P_{SG}, s_i) \tag{5}$$

This categorization into background and foreground elements is motivated by the relative stability of background objects, which facilitates subsequent information compression. The resulting SG undergoes normalization to ensure consistent object naming and relation predicates.

Fig. 4 illustrates a SG generated for a sailboat video segment. In the visualization, blue elements represent background components (such as "clear sky"), orange elements indicate foreground objects (including the sailboats with their attributes like "black" and "left"), and red elements denote actions and relationships (such as the "accompanied" relationship between vessels).

3.2 Delta Scene Graph Generator

After the key-SG Detector identifies critical segments that warrant complete SG representation, the remaining segments—termed delta segments—require a more efficient representation strategy. The Delta Scene Graph Generator (DSGG) addresses this need by creating compact, change-focused SGs that only capture modifications relative to the nearest key segment, significantly reducing token consumption while preserving semantic continuity.

3.2.1 Differential Detection. Given a delta segment s_δ and its temporally nearest preceding key segment s_k with corresponding key SG $G_k = (V_k, E_k, A_k)$, the DSGG's objective is to generate a delta SG $G_\delta = (V_\delta, E_\delta, A_\delta)$ that exclusively represents the elements that have changed:

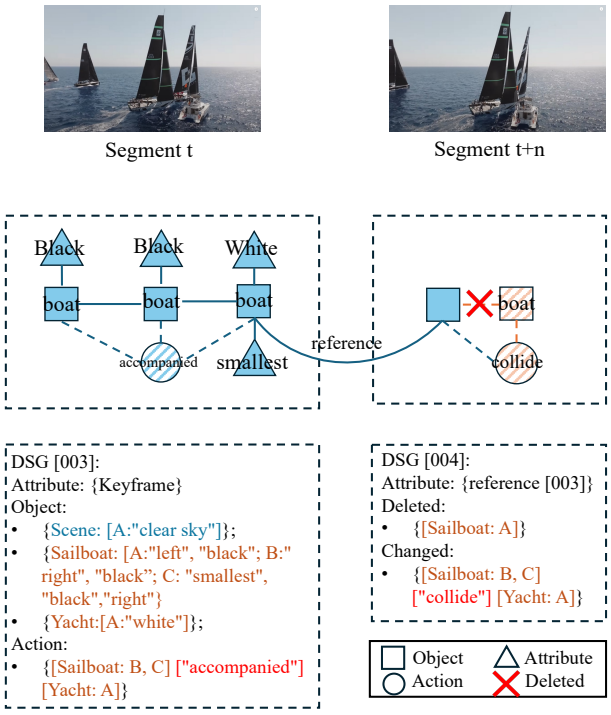

Figure 4: Compressed Scene Graph example.

$$G_\delta = \mathcal{F}_{\text{diff}}(s_\delta, s_k, G_k) \tag{6}$$

where $\mathcal{F}_{\text{diff}}$ is a differential analysis function implemented through structured prompting of the Video-LLM. Specifically, we instruct the model to focus on detecting changes across five categories:

$$\delta = \{\delta_{\text{appear}}, \delta_{\text{disappear}}, \delta_{\text{attribute}}, \delta_{\text{relation}}, \delta_{\text{position}}\} \tag{7}$$

where δ_{appear} represents new objects appearing in the scene, $\delta_{\text{disappear}}$ denotes objects no longer present, $\delta_{\text{attribute}}$ captures changes in object attributes, δ_{relation} identifies modified relationships between objects, and δ_{position} tracks significant spatial repositioning of persistent objects.

3.2.2 Delta Scene Graph Generation. To efficiently extract these changes, we prompt the Video-LLM with a specialized instruction template P_δ implemented as:

> P_δ: Given the video segment $\{s_\delta\}$ and reference SG $\{G_k\}$, identify only the changes that have occurred. Focus on:
> ① New objects that have appeared;
> ② Objects that have disappeared;
> ③ Changes in object attributes;
> ④ Modified relationships between objects;
> ⑤ Significant spatial repositioning.
> Please ensure adherence to constraints C_{min} and C_{ref}.

where two exemplar constraints are:

> C_{min}: Include only elements have changed from the $\{G_k\}$.
> C_{ref}: Maintain clear references to objects in the G_k.

The implementation of the differential SG generation function can be formulated as:

$$G_\delta = \text{VideoLLM}(P_\delta, s_\delta, G_k) \tag{8}$$

The resulting delta SG maintains references to the key SG through directional edges, creating an incomplete but connected graph structure. For newly appearing objects $v_{\text{new}} \in V_\delta$, we add them as complete nodes with their attributes. For persistent objects that experience attribute or relation changes, we create partial nodes with single-directional reference edges pointing to their complete counterparts in the key SG:

$$E_{\text{ref}} = \{(v_i^\delta \rightarrow v_j^k) | v_i^\delta \in V_\delta, v_j^k \in V_k, \text{id}(v_i^\delta) = \text{id}(v_j^k)\} \tag{9}$$

where $\text{id}(v)$ represents the unique identifier of an object node, enabling cross-graph references.

Fig. 4 illustrates this concept with a delta SG (labeled as DSG [004]) that references a previous key SG (DSG [003]). In this example, the delta SG captures three key changes: the disappearance of Sailboat A, a collision between Sailboats B and C with Yacht A (replacing the previous "accompanied" relationship), and maintains referential links to unchanged elements in the key SG. In the visualization, these reference links are depicted as directed edges connecting to corresponding nodes in the key SG, eliminating the need to duplicate unchanged information.

The token efficiency gain of delta SGs is particularly pronounced when consecutive delta segments exhibit minimal changes. Mathematically, the token reduction ratio R can be expressed as:

$$R = 1 - \frac{\sum_{i \in \delta} |G_{\delta_i}|}{\sum_{i \in \delta} |G_{\text{full}_i}|} \tag{10}$$

where $|G_{\delta_i}|$ represents the token count of the delta SG for segment i, and $|G_{\text{full}_i}|$ is the token count of a hypothetical full SG for the same segment. In our experiments, this ratio typically ranges from 0.4 to 0.7.

3.3 Scene Graph Query Manager

The Scene Graph Query Manager (SGQM) completes our *CSGCoT* framework by storing, retrieving, and converting SGs to support the CoT reasoning process. After generation, both key SGs and delta SGs are stored in a structured repository called SG Memory.

3.3.1 Scene Graph Memory and Query. The SGQM operates on a query-based retrieval paradigm. Each SG stored in SG Memory is indexed with a unique identifier that corresponds to its video segment:

$$\text{SG\_Memory} = \{(id_1, G_1), (id_2, G_2), ..., (id_n, G_n)\} \tag{11}$$

where id_i is the unique identifier for SG G_i.

The implementation of the query function can be formulated as:

Q_{SG}: Given the SG identifier $\{id_i\}$ referenced in the CoT reasoning process, retrieve the corresponding SG from SG Memory and determine whether conversion is required.
If G_i is a key SG, return it directly. If G_i is a delta SG, perform conversion using its reference key SG before returning.

3.3.2 Scene Graph Conversion. When the CoT process requires detailed visual information about a specific segment, it includes the embedded SG identifier in its query. The SGQM intercepts these queries and processes them according to the following retrieval algorithm:

$$G_{\text{retrieved}} = \begin{cases} G_k & \text{if } G_{\text{requested}} \text{ is a key SG} \\ \mathcal{R}(G_\delta, G_k) & \text{if } G_{\text{requested}} \text{ is a delta SG} \end{cases} \tag{12}$$

where $\mathcal{R}$ represents the conversion function that completes a delta SG by incorporating information from its corresponding key SG. This conversion process follows a graph merging operation:

$$\mathcal{R}(G_\delta, G_k) = (V_k \cup V_\delta \setminus V_{\text{disappeared}}, E_k \cup E_\delta \setminus E_{\text{changed}}, A_k \cup A_\delta \setminus A_{\text{changed}}) \tag{13}$$

where $V_{\text{disappeared}}$, E_{changed}, and A_{changed} represent nodes, edges, and attributes that have been modified or removed according to the delta SG.

The conversion process resolves reference edges by replacing them with the complete subgraphs from the key SG, ensuring that the resulting SG contains comprehensive information about all relevant objects and their relationships.

3.3.3 Natural Language Conversion. For CoT operations that require natural language descriptions rather than structured graph representations, the SGQM includes a graph-to-text conversion module. This module transforms the structured SG into fluent natural language through a template-based approach enhanced by the Video-LLM's language capabilities:

$$\text{Text}_G = \mathcal{F}_{G2T}(G_{\text{retrieved}}) = \mathcal{F}_{G2T}(\mathcal{B}(G), \mathcal{F}(G), \mathcal{A}(G)) \tag{14}$$

where $\mathcal{F}_{G2T}$ represents the graph-to-text conversion function, and $\mathcal{B}(G)$, $\mathcal{F}(G)$, and $\mathcal{A}(G)$ extract the background, foreground, and action components from the SG, respectively.

The implementation of conversion function is formulated as:

$\mathcal{F}_{G2T}$: Given the converted SG $\{G_{\text{retrieved}}\}$, generate a coherent natural language description that captures all significant elements:
① Describe the background scene elements first;
② Introduce foreground objects with their attributes;
③ Articulate the relationships and actions between objects.

4 Evaluation

4.1 Evaluation Setup

Our work focuses on efficiency improvements, making it suitable for edge computing or local deployment scenarios. Therefore, we assess the performance of our technique using a testbed equipped with two NVIDIA RTX 3090 GPUs and an AMD Ryzen 9 5900X CPU, running on Ubuntu 20.04 LTS. The source code can be accessed at https://github.com/helloLT/CSGCoT/.

Table 2: Comparison of Long Video QA Datasets

Dataset	#Videos	Avg.Len.(min)	OD-Type Tasks
EgoSchema [15]	5,063	3	—
MoVQA [41]	100	16.53	SP
MLVU [47]	1,334	12	NQA, SSC
Video-MME [6]	900	17	AR, OR, AP
LV Bench [24]	103	68	ED, AR, SP
ALLVB [19]	1,376	114.62	SR, ODT, AR, VCap

4.1.1 Backbone Models. To evaluate *CSGCoT*'s performance and generalizability, we use **VideoLLaMA-2** [2] as the visual perception model and tested with different large language models as CoT reasoning backbones. This approach demonstrates that our method is model-agnostic and effective across different architectures. Specifically, we employed **Vicuna-7B** [3] for fair comparisons with other works, and **LLaMA-3.1-8B** [8], which is an improved version of Meta's third-generation LLaMA model offering enhanced performance (denoted with (*) in Tab. 3). For fair comparison, the benchmark methods are implemented using identical backbone models.

4.1.2 Datasets. We evaluate *CSGCoT* using recent multimodal question answering (QA) of long video datasets with durations ranging from 3 minutes to over 2 hours. In ascending order of average length, these include EgoSchema [15], MLVU [47], MoVQA [41], Video-MME [6], LV Bench [24], and ALLVB [19], as summarized in Tab. 2.

For identifying Object Description (OD) tasks, we employed two approaches. First, we manually selected OD-related subtasks based on official data split, including Spatial Perception (SP), Needle Question-Answering (NQA), Sub-Scene Captioning (SSC), Action Recognition (AR), Object Recognition (OR), Attribute Perception (AP), Entity Detection (ED), Scene Recognition (SR), Object Detection, Tracking (ODT) and Video Captioning (VCap). Second, for fair comparison across all datasets (including EgoSchema which lacks predefined categories), we implemented automatic task classification using GPT-4o to systematically identify OD-type tasks.

4.1.3 Evaluation metrics. To evaluate the performance and effectiveness of *CSGCoT* and benchmark methods, we consider two primary metrics: (1) Accuracy, which is the percentage of correctly answered questions, measuring the method's performance in producing correct responses; (2) Latency, which is the time elapsed from question input to answer generation, measuring the method's computational efficiency.

4.1.4 Benchmark methods. We compare *CSGCoT* with the following SOTA prompt-based MCoT reasoning methods:

- **VideoCoT** [29]: A prompt-based Video-CoT reasoning method that uses structured visual representations (FAMOuS). We maintain the same parameters as in the original paper.
- **VideoAgent** [26]: An efficiency-focused prompt-based Video-CoT reasoning method that emphasizes processing reasoning problems in long videos with limited frames. It represents a STOA approach balancing efficiency and precision.

Table 3: Performance comparison of different methods across various video understanding benchmarks.

Method	without subtitle						with subtitle					
	EgoSchema		MLVU		LV Bench		Video-MME		MoVQA		ALLVB	
	Acc.(%)	Latency(s)	Acc.(%)	Latency(s)	Acc.(%)	Latency(s)	Acc.(%)	Latency(s)	Acc.(%)	Latency(s)	Acc.(%)	Latency(s)
VideoCoT [29]	34.3	38	44.9	87	30.1	373	61.7	117	43.4	115	34.7	563
VideoCoT [29]	37.7	53	48.1	115	35.4	517	65.4	165	46.9	154	40.8	782
VideoAgent [26]	46.7	31	46.2	**25**	38.6	**163**	63.3	**37**	45.1	**31**	43.9	**119**
VideoAgent* [26]	**53.2**	28	51.9	33	38.2	195	68.7	47	50.3	39	43.6	162
VoT [5]	49.8	22	54.6	54	41.3	211	72.3	78	53.4	69	45.5	352
VoT* [5]	52.2	30	**56.7**	75	42.9	393	**76.8**	107	**57.3**	97	54.3	588
CSGCoT	49.3	**18**	52.9	49	41.5	168	70.1	65	52.8	66	45.2	148
CSGCoT*	52.7	25	56.2	62	**43.8**	260	75.6	85	55.9	82	**55.3**	220

- **VoT** [5]: A method using spatiotemporal scene graphs (STSG), providing a comparison of different design directions with similar technical approaches. The visual perception model remains ViT-L/14 as in the original implementation, with parameters matching the original recommendations: 8fps sampling rate with LoRA technology.

4.2 Overall Performance Comparison

Tab. 3 presents a comprehensive comparison of *CSGCoT* against SOTA methods across long video understanding benchmarks, which are categorized into videos without subtitles (EgoSchema, MLVU, LV Bench) and with subtitles (Video-MME, MoVQA, ALLVB).

4.2.1 Accuracy Performance in Long Video Understanding. For accuracy performance, *CSGCoT* and VoT demonstrate superior accuracy compared to VideoCoT and VideoAgent across all benchmarks. When compared with VoT, which achieves the highest accuracy on several benchmarks, *CSGCoT* shows competitive performance with only a marginal 1-3% accuracy difference in most datasets. *CSGCoT** achieves 43.8% accuracy on LV Bench and 55.3% on ALLVB, outperforming VoT* (42.9% and 54.3%, respectively). This suggests that *CSGCoT* exhibits particular strengths in hour-long videos.

4.2.2 Efficiency Performance in Long Video Understanding. For efficiency performance, *CSGCoT* and VideoAgent demonstrate markedly lower latencies compared to VideoCoT and VoT across all benchmarks. In comparison to VideoAgent, *CSGCoT* demonstrates only a modest increase in latency (within 10% for most benchmarks) while delivering substantially higher accuracy. For instance, on MLVU, *CSGCoT* achieves 52.9% accuracy with 49s latency compared to VideoAgent's 46.2% accuracy with 25s latency—a 6.7% accuracy improvement with reasonable latency trade-off.

The efficiency advantage of *CSGCoT* becomes especially evident in hour-long videos such as LV Bench and ALLVB. On LV Bench, *CSGCoT* achieves a latency of 168s, approaching VideoAgent's 163s while offering significantly better accuracy (41.5% vs. 38.6%). Similarly, on ALLVB, *CSGCoT* maintains a competitive latency of 148s compared to VideoAgent's 119s, while delivering superior accuracy (45.2% vs. 43.9%).

The contrast between *CSGCoT* and VoT in terms of latency further validates our approach. While both methods leverage SG representations, *CSGCoT* reduces computational overhead. For example, on ALLVB, *CSGCoT** achieves 55.3% accuracy with 220s latency, while VoT* reaches 54.3% accuracy with 588s latency—a mere 1% accuracy difference with a 62.6% reduction in processing time.

4.3 Object Detection Task Performance Comparison

Tab. 4 presents performance comparisons on OD tasks across benchmarks. The table is divided into manual data split (based on datasets' official OD task categorizations) and automatic data split (using LLM for consistent classification across datasets), enabling both comparison with prior work and fair cross-dataset evaluation.

4.3.1 Accuracy Performance in Object Detection Tasks. *CSGCoT* demonstrates superior accuracy across most datasets in both split settings. In the manual split, *CSGCoT** achieves the highest accuracy on all benchmarks: 59.0% on MLVU, 47.2% on LV Bench, 78.4% on Video-MME, 59.3% on MoVQA, and 57.6% on ALLVB, outperforming other approaches by margins of 8.7-18.7%.

When compared with VoT, *CSGCoT* shows consistent improvements, especially on longer videos. For instance, *CSGCoT** outperforms VoT* by 3.1% on MLVU and 4.9% on both LV Bench and ALLVB. In the automatic split setting, *CSGCoT** achieves highest accuracy on five of six datasets, with comparable performance (55.3% vs. 55.9%) to VideoAgent* on EgoSchema.

4.3.2 Efficiency Performance in Object Detection Tasks. In the manual split, *CSGCoT* delivers the lowest latency on LV Bench (170s) while outperforming all methods in accuracy. For efficiency-accuracy tradeoffs, key examples include MLVU, where *CSGCoT* achieves 10.7% higher accuracy than VideoAgent with just 16s additional latency (56.3% at 54s vs. 45.6% at 38s), and MoVQA, where a 13.3% accuracy gain comes with reasonable additional processing time (57.6% at 73s vs. 44.3% at 36s).

In the automatic split, *CSGCoT* achieves the lowest latency on EgoSchema (20s) and maintains competitive efficiency across all datasets. The advantage becomes more pronounced on longer videos: on ALLVB, *CSGCoT* maintains 196s latency (vs. VideoAgent's 145s) while delivering substantially higher accuracy (49.8% vs. 40.3%). Compared to VoT, *CSGCoT* demonstrates efficiency gains across all datasets. On ALLVB, *CSGCoT** achieves 4.9% better accuracy with 56.6% reduced processing time (57.6% at 254s vs. 52.7% at 585s).

4.4 Ablation Study

To investigate the impact of each component in *CSGCoT*, we conducted ablation studies on three key modules using both the EgoSchema dataset and the ALLVB dataset. Tab. 5 presents the results.

4.4.1 Study on Key Scene Graph Detector. We evaluated the impact of different SG generation algorithms for the key-SG detector by

Table 4: Performance comparison of different methods across object description tasks.

	EgoSchema		MLVU		LV Bench		Video-MME		MoVQA		ALLVB	
	Acc.(%)	Latency(s)	Acc.(%)	Latency(s)	Acc.(%)	Latency(s)	Acc.(%)	Latency(s)	Acc.(%)	Latency(s)	Acc.(%)	Latency(s)
OD tasks with manual data split												
VideoCOT [29]	-	-	43.1	100	29.2	430	59.8	135	42.5	132	33.4	645
VideoCOT* [29]	-	-	46.3	115	33.7	515	63.2	160	45.8	150	38.9	780
VideoAgent [26]	-	-	45.6	38	37.5	185	61.9	55	44.3	**36**	41.2	**140**
VideoAgent* [26]	-	-	50.3	**28**	41.8	210	66.5	**43**	49.1	45	46.2	185
VoT [5]	-	-	53.2	62	40.1	245	70.8	92	52.6	81	47.3	405
VoT* [5]	-	-	55.9	75	42.3	390	74.1	105	55.8	95	52.7	585
CSGCoT	-	-	56.3	54	45.7	**170**	73.9	72	57.6	73	50.5	173
CSGCoT*	-	-	**59.0**	69	**47.2**	247	**78.4**	94	**59.3**	91	**57.6**	254
OD tasks with automatic data split												
VideoCOT [29]	36.8	44	44.7	107	28.1	449	58.3	142	43.9	138	32.6	658
VideoCOT* [29]	39.6	62	47.5	127	32.9	535	61.8	175	46.3	163	37.4	812
VideoAgent [26]	49.1	36	47.2	42	35.9	**193**	59.6	58	46.1	**39**	40.3	**145**
VideoAgent* [26]	**55.9**	33	53.8	**33**	39.7	218	63.1	**47**	51.4	48	45.9	203
VoT [5]	52.3	26	55.8	68	38.4	261	67.3	97	54.7	86	47.6	422
VoT* [5]	52.2	30	56.3	82	41.7	409	73.8	112	55.1	103	53.9	602
CSGCoT	51.8	**20**	58.7	59	44.3	197	72.9	77	56.8	79	49.8	196
CSGCoT*	55.3	28	**61.4**	75	**45.7**	302	**78.1**	101	**58.3**	98	**57.9**	286

Table 5: Performance comparison in ablation study. The bottom line is the component of our method.

(a) Study on Key Scene Graph Detector

Method	EgoSchema		ALLVB	
	Acc.(%)	Latency(s)	Acc.(%)	Latency(s)
SG (CCoT)	51.1	27	54.1	269
STSG (VoT)	**53.9**	27	**55.7**	277
Key SG	53.7	**25**	55.3	**220**

(b) Study on Delta Scene Graph Generator

Method	EgoSchema		ALLVB	
	Acc.(%)	Latency(s)	Acc.(%)	Latency(s)
Full SG	51.5	25	52.9	490
Static SG	50.1	31	51.8	429
Delta SG	**52.7**	**25**	**55.3**	**220**

(c) Study on Scene Graph Query Manager

Method	EgoSchema		ALLVB	
	Acc.(%)	Latency(s)	Acc.(%)	Latency(s)
w/o Recons.	51.0	**25**	53.8	**203**
Full Recons.	**53.2**	31	**56.0**	391
Sel. Recons.	52.7	**25**	55.3	220

comparing our approach (Key SG) with two alternatives: a static image-based SG used in CCoT [16] and a spatial-temporal scene graph (STSG) employed in VoT [5]. Our key-SG approach shows minimal accuracy trade-offs (0.2-0.4% reduction) while achieving significant efficiency improvements, particularly on ALLVB with a 20.6% reduction in latency (220s vs. 277s). This efficiency stems from selectively capturing only critical information at key frames, effectively balancing accuracy and computational demands.

4.4.2 Study on Delta Scene Graph Generator. To evaluate the effectiveness of our delta-SG generator, we compared our approach (Delta SG) with two alternative strategies: generating a complete SG for each segment (Full SG) and reusing the previous key SG information without updates (Static SG). Delta SG outperforms both alternatives in accuracy and efficiency, achieving 1.2-2.4% higher accuracy than Full SG while reducing latency by up to 55.1% on ALLVB (220s vs. 490s). Compared to Static SG, Delta SG delivers 2.6-3.5% better accuracy with improved latency across both datasets, validating our approach of preserving essential information while minimizing redundancy.

4.4.3 Study on Scene Graph Query Manager. We investigated different strategies for reconstructing SGs into natural language descriptions in our SG Query Manager. We compared our selective reconstruction approach (Sel. Recons.) with two alternatives: omitting reconstruction entirely (w/o Recons.) and performing full reconstruction for all SGs (Full Recons.). While Full Recons. achieves slightly higher accuracy (+0.5-0.7%), it increases latency by 24-77.7%. Conversely, w/o Recons. offers similar efficiency but reduces accuracy by 1.5-1.7%. Our selective reconstruction achieves nearly the same speed as w/o Recons. while maintaining accuracy close to Full Recons. by only reconstructing SGs that directly contribute to answer generation, demonstrating an optimal balance between computational efficiency and reasoning accuracy.

5 Conclusion

We introduced *CSGCoT*, an approach that leverages compressed SGs to enhance CoT reasoning for long video understanding. By employing key segment representation and differential encoding for intermediate frames, our method reduces computational demands while preserving semantic information. We assess the performance of our technique in edge computing or local deployment scenarios, and experiments show that *CSGCoT* achieves comparable accuracy to SOTA methods while reducing latency by over 62.6% on hour-long videos.

Acknowledgments

Dan Wang's work is supported in part by RGC GRF 15200321, 15201322, 15230624, 15239925, ITC ITF-ITS/056/22MX, ITS/052/23MX, and PolyU 1-CDKK, G-SAC8, K-ZYAP. Dan Wang is the corresponding author.

References

[1] Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wangxiang Che. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567* (2025).

[2] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. 2024. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476* (2024).

[3] Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *https://vicuna.lmsys.org* (2023).

[4] Xiuliang Duan, Dating Tan, Liangda Fang, Yuyu Zhou, Chaobo He, Ziliang Chen, Lusheng Wu, Guanliang Chen, Zhiguo Gong, Weiqi Luo, et al. 2024. Reason-and-execute prompting: Enhancing multi-modal large language models for solving geometry questions. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 6959–6968.

[5] Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, Meishan Zhang, Mong Li Lee, and Wynne Hsu. 2024. Video-of-thought: step-by-step video reasoning from perception to cognition. In *Proceedings of the 41st International Conference on Machine Learning*. 13109–13125.

[6] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. 2024. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075* (2024).

[7] Timin Gao, Peixian Chen, Mengdan Zhang, Chaoyou Fu, Yunhang Shen, Yan Zhang, Shengchuan Zhang, Xiawu Zheng, Xing Sun, Liujuan Cao, et al. 2024. Cantor: Inspiring multimodal chain-of-thought of mllm. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 9096–9105.

[8] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[9] Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Peng Gao, Hongsheng Li, and Pheng-Ann Heng. 2025. Can We Generate Images with CoT? Let's Verify and Reinforce Image Generation Step by Step. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[10] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. 2018. Mapping images to scene graphs with permutation-invariant structured prediction. *Advances in Neural Information Processing Systems* 31 (2018).

[11] Justin Johnson, Agrim Gupta, and Li Fei-Fei. 2018. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1219–1228.

[12] Jiapeng Liu, Chengyang Fang, Liang Li, Bing Li, Dayong Hu, and Can Ma. 2024. Prompting large language models with fine-grained visual relations from scene graph for visual question answering. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 8125–8129.

[13] Xiaojian Ma, Weili Nie, Zhiding Yu, Huaizu Jiang, Chaowei Xiao, Yuke Zhu, Song-Chun Zhu, and Anima Anandkumar. 2022. RelViT: Concept-guided Vision Transformer for Visual Relational Reasoning. In *International Conference on Learning Representations*.

[14] Yingzi Ma, Yulong Cao, Jiachen Sun, Marco Pavone, and Chaowei Xiao. 2024. Dolphins: Multimodal language model for driving. In *European Conference on Computer Vision*. Springer, 403–420.

[15] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. 2023. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems* 36 (2023), 46212–46244.

[16] Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. 2024. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14420–14431.

[17] Haiyi Qiu, Minghe Gao, Long Qian, Kaihang Pan, Qifan Yu, Juncheng Li, Wenjie Wang, Siliang Tang, Yueting Zhuang, and Tat-Seng Chua. 2024. STEP: Enhancing Video-LLMs' Compositional Reasoning by Spatio-Temporal Graph-guided Self-Training. *arXiv preprint arXiv:2412.00161* (2024).

[18] Cheng Tan, Jingxuan Wei, Zhangyang Gao, Linzhuang Sun, Siyuan Li, Ruifeng Guo, Bihui Yu, and Stan Z Li. 2024. Boosting the power of small multimodal reasoning models to match larger models with self-consistency training. In *European Conference on Computer Vision*. Springer, 305–322.

[19] Xichen Tan, Yuanjing Luo, Yunfan Ye, Fang Liu, and Zhiping Cai. 2025. ALLVB: All-in-One Long Video Understanding Benchmark. *arXiv preprint arXiv:2503.07298* (2025).

[20] Lv Tang, Peng-Tao Jiang, Zhi-Hao Shen, Hao Zhang, Jin-Wei Chen, and Bo Li. 2024. Chain of visual perception: Harnessing multimodal large language models for zero-shot camouflaged object detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 8805–8814.

[21] Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, et al. 2023. Video understanding with large language models: A survey. *arXiv preprint arXiv:2312.17432* (2023).

[22] Ye Tian, Mengyu Yang, Lanshan Zhang, Zhizhen Zhang, Yang Liu, Xiaohui Xie, Xirong Que, and Wendong Wang. 2023. View while moving: Efficient video recognition in long-untrimmed videos. In *Proceedings of the 31st ACM International Conference on Multimedia*. 173–183.

[23] Lei Wang, Yi Hu, Jiabang He, Xing Xu, Ning Liu, Hui Liu, and Heng Tao Shen. 2024. T-sciq: Teaching multimodal chain-of-thought reasoning via large language model signals for science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 19162–19170.

[24] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, et al. 2024. Lvbench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035* (2024).

[25] Xiao Wang, Yaoyu Li, Tian Gan, Zheng Zhang, Jingjing Lv, and Liqiang Nie. 2023. RTQ: Rethinking Video-language Understanding Based on Image-text Model. In *Proceedings of the 31st ACM International Conference on Multimedia*. 557–566.

[26] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. 2024. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*. Springer, 58–76.

[27] Yulu Wang, Pengwen Dai, Xiaojun Jia, Zhitao Zeng, Rui Li, and Xiaochun Cao. 2023. Hi-sigir: Hierachical semantic-guided image-to-image retrieval via scene graph. In *Proceedings of the 31st ACM International Conference on Multimedia*. 6400–6409.

[28] Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, William Wang, Ziwei Liu, Jiebo Luo, and Hao Fei. 2025. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605* (2025).

[29] Yan Wang, Yawen Zeng, Jingsheng Zheng, Xiaofen Xing, Jin Xu, and Xiangmin Xu. 2024. VideoCoT: A Video Chain-of-Thought Dataset with Active Annotation Tool. In *Proceedings of the 3rd Workshop on Advances in Language and Vision Research (ALVR)*. 92–101.

[30] Chao-Yuan Wu and Philipp Krahenbuhl. 2021. Towards long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1884–1894.

[31] Minghui Wu, Chenxu Zhao, Anyang Su, Donglin Di, Tianyu Fu, Da An, Min He, Ya Gao, Meng Ma, Kun Yan, et al. 2024. Hypergraph Multi-modal Large Language Model: Exploiting EEG and Eye-tracking Modalities to Evaluate Heterogeneous Responses for Video Understanding. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 7316–7325.

[32] Peng Wu, Xuerong Zhou, Guansong Pang, Zhiwei Yang, Qingsen Yan, Peng Wang, and Yanning Zhang. 2024. Weakly supervised video anomaly detection and localization with spatio-temporal prompts. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 9301–9310.

[33] Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. 2024. Mind's Eye of LLMs: Visualization-of-Thought Elicits Spatial Reasoning in Large Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[34] Yixuan Wu, Yizhou Wang, Shixiang Tang, Wenhao Wu, Tong He, Wanli Ouyang, Philip Torr, and Jian Wu. 2024. Dettoolchain: A new prompting paradigm to unleash detection ability of mllm. In *European Conference on Computer Vision*. Springer, 164–182.

[35] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9777–9786.

[36] Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P Lillicrap, Kenji Kawaguchi, and Michael Shieh. [n. d.]. Monte Carlo Tree Search Boosts Reasoning via Iterative Preference Learning. In *The First Workshop on System-2 Reasoning at Scale, NeurIPS'24*.

[37] Yuanxing Xu, Yuting Wei, and Bin Wu. 2023. Query-aware Long Video Localization and Relation Discrimination for Deep Video Understanding. In *Proceedings of the 31st ACM International Conference on Multimedia*. 9591–9595.

[38] Jingkang Yang, Yi Zhe Ang, Zujin Guo, Kaiyang Zhou, Wayne Zhang, and Ziwei Liu. 2022. Panoptic scene graph generation. In *European Conference on Computer Vision*. Springer, 178–196.

[39] Jun Cheng Yang, Zuchao Li, Shuai Xie, Wei Yu, Shijun Li, and Bo Du. 2024. Soft-Prompting with Graph-of-Thought for Multi-modal Representation Learning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 15024–15036.

[40] Junyi Yao, Yijiang Liu, Zhen Dong, Mingfei Guo, Helan Hu, Kurt Keutzer, Li Du, Daquan Zhou, and Shanghang Zhang. 2024. PromptCoT: Align Prompt Distribution via Adapted Chain-of-Thought. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7027–7037.

[41] Hongjie Zhang, Yi Liu, Lu Dong, Yifei Huang, Zhen-Hua Ling, Yali Wang, Limin Wang, and Yu Qiao. 2023. Movqa: A benchmark of versatile question-answering for long-form movie understanding. *arXiv preprint arXiv:2312.04817* (2023).

[42] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923* (2023).

[43] Jianing Zhao, Jingjing Wang, Yujie Jin, Jiamin Luo, and Guodong Zhou. 2024. Hawkeye: Discovering and Grounding Implicit Anomalous Sentiment in Recon-videos via Scene-enhanced Video Large Language Model. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 592–601.

[44] Yu Zhao, Hao Fei, Yixin Cao, Bobo Li, Meishan Zhang, Jianguo Wei, Min Zhang, and Tat-Seng Chua. 2023. Constructing holistic spatio-temporal scene graph for video semantic role labeling. In *Proceedings of the 31st ACM International Conference on Multimedia*. 5281–5291.

[45] Changmeng Zheng, Dayong Liang, Wengyu Zhang, Xiao-Yong Wei, Tat-Seng Chua, and Qing Li. 2024. A Picture Is Worth a Graph: A Blueprint Debate Paradigm for Multimodal Reasoning. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 419–428.

[46] Shanshan Zhong, Zhongzhan Huang, Shanghua Gao, Wushao Wen, Liang Lin, Marinka Zitnik, and Pan Zhou. 2024. Let's think outside the box: Exploring leap-of-thought in large language models with creative humor generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13246–13257.

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

[48] Heqing Zou, Tianze Luo, Guiyang Xie, Fengmao Lv, Guangcong Wang, Junyang Chen, Zhuochen Wang, Hansheng Zhang, Huaijian Zhang, et al. 2024. From seconds to hours: Reviewing multimodal large language models on comprehensive long video understanding. *arXiv preprint arXiv:2409.18938* (2024).