Response-G1: Explicit Scene Graph Modeling for Proactive Streaming Video Understanding

Ke Ma[1,2†], Jiaqi Tang[3†], Bin Guo[1*], Xueting Han[1], Ruonan Xu[1], Qingfeng He[2], Ziheng Wang[1], Xu Wang[2], Qifeng Chen[3], Zhiwen Yu[1,4], Yunhao Liu[2*]

[1]Northwestern Polytechnical University, [2]Tsinghua University

[3]The Hong Kong University of Science and Technology, [4]Harbin Engineering University

Project Page: https://github.com/kadmkbl/Response-G1

Abstract

Proactive streaming video understanding requires Video-LLMs to decide when to respond as a video unfolds, a task where existing methods often fall short due to their implicit, query-agnostic modeling of visual evidence. We introduce **Response-G1**, a novel framework that establishes explicit, structured alignment between the accumulated video evidence and the query's expected response conditions via scene graphs. The framework operates in three fine-tuning-free stages: (1) online query-guided scene graph generation from streaming clips; (2) memory-based retrieval of the most semantically relevant historical scene graphs; and (3) retrieval-augmented trigger prompting for per-frame "silence/response" decisions. By grounding both evidence and conditions in a shared graph representation, **Response-G1** achieves more interpretable and accurate response timing decisions. Experimental results on established benchmarks demonstrate the superiority of our method in both proactive and reactive tasks, validating the advantage of explicit scene graph modeling and retrieval in streaming video understanding.

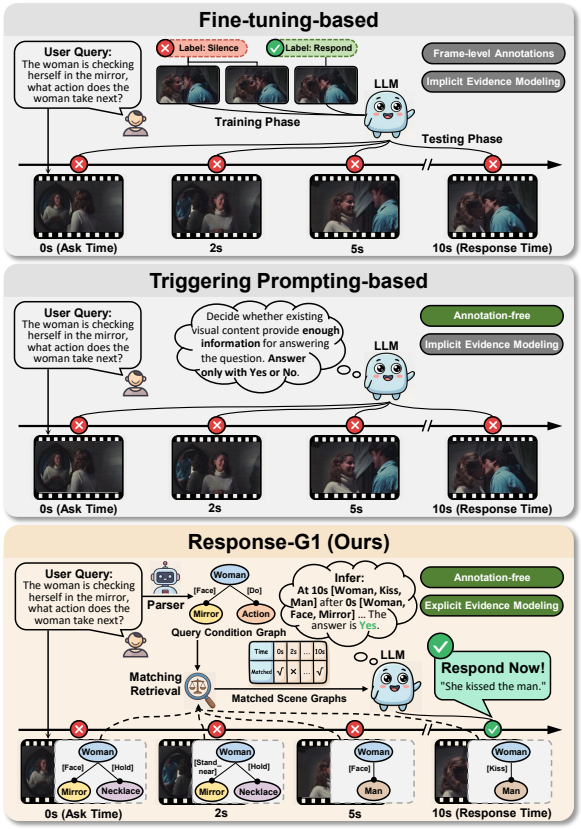

Figure 1: Existing proactive mechanisms in streaming video understanding. By explicitly modeling both observed visual evidence and query-specific conditions via scene graphs, **Response-G1** achieves accurate decisions of response timing.

1 Introduction

Video Large Language Models (Video-LLMs) have established themselves as a dominant paradigm for a wide range of video comprehension tasks (Wang et al., 2025b). A critical frontier within this domain is Streaming Video Understanding (SVU), which focuses on processing continuous, real-time video feeds (Qian et al., 2024; Zhang et al., 2025a). Unlike offline analysis, SVU models must perceive, reason, and interact based solely on the incrementally observed visual content up to the current moment, making it indispensable for applications demanding immediate responsiveness, such as conversational AI assistants (Wen et al., 2025) and autonomous embodied agents (Zhang et al., 2025b).

The prevalent paradigm in current SVU research is *reactive interaction*, where models are designed to respond immediately to user queries (Di et al., 2025; Xiong et al., 2025; Zeng et al., 2025). However, this paradigm is fundamentally limited for queries that are predictive or anticipatory in nature, as the information required for a sufficient answer may only appear in future video segments. To

† These authors contributed equally to this work.
* Corresponding authors. guob@nwpu.edu.cn, yunhao@tsinghua.edu.cn

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 44139–44153

address this, Video-LLMs need the capability for *proactive interaction*, to autonomously determine the optimal moment to answer based on accumulated evidence (Niu et al., 2025).

To implement proactive mechanisms, existing works have explored various approaches. One line of research (Chen et al., 2024a; Li et al., 2025) employs a streaming End-Of-Sequence (EOS) prediction pipeline, where the Video-LLM is trained to decode the EOS token to remain silent on frames that are not annotated for a response. Similarly relying on fine-tuning, another line of work (Qian et al., 2025; Wang et al., 2025a) introduces an auxiliary activation LLM with a binary classification head, which directly outputs a silence-or-response decision at each frame. However, as shown in Figure 1, fine-tuning-based methods critically rely on detailed, frame-wise annotations, where consecutive and visually similar frames are often assigned opposite silence-or-response labels. This inconsistency significantly hinders the learning of a reliable decision boundary. Among fine-tuning-free approaches, (Yao et al., 2025) employs thresholds triggered by inter-frame differences. While simple, this method completely ignores query semantics, leading to suboptimal response timing. (Yang et al., 2025c) adopts a multi-LLM-agent framework, where response timing in video streams is determined through prompting a dedicated planner.

A fundamental limitation common to these approaches is their **implicit modeling** of the visual evidence and the response conditions implied by the query. This implicit modeling limits the Video-LLM's ability to comprehensively understand and align the accumulated streaming evidence with query-specific conditions, thereby limiting the accuracy of response timing. To overcome this, we argue that explicit representations are necessary, as they disentangle the underlying semantics and support more principled reasoning over streaming evidence. Specifically, we observe that the user query typically depicts a scene that includes the objects and relations anticipated by the response conditions (see Figure 1, which shows a query targeting a woman checking herself in the mirror). Given that query-relevant scenes are inherently structured through objects and relations, we propose using scene graphs (Johnson et al., 2015) as a unifying representation to explicitly model both the visual evidence and the response conditions.

Driven by this insight, we propose **Response-G1**, a novel framework that establishes a complete scene-graph-driven pipeline for proactive streaming video understanding. Its core consists of three integrated components: (1) **Online Query-guided Scene Graph Generation**: We leverage Video-LLMs to abstract a scene graph for streaming video clips, focusing on salient, query-related evidence. (2) **Memory-based Scene Graph Retrieval**: A dynamic memory bank stores historical scene graphs. During streaming processing, we retrieve scene graphs most semantically relevant to the response conditions, providing a concise, aligned evidence context for decision-making. (3) **Retrieval-augmented Streaming Decision & Response**: The retrieved scene graphs, interleaved with temporal cues, are fed into the Video-LLM alongside the visual tokens. A trigger-prompting mechanism enables per-frame silence-or-response decisions, and upon triggering, the final answer is generated using the same enriched context. Our framework operates in a fine-tuning-free manner, enhancing the model's inherent capabilities.

Extensive evaluations on StreamingBench (Lin et al., 2024b) and OVO-Bench (Niu et al., 2025) demonstrate that **Response-G1** significantly improves the accuracy of response timing and the quality of final answers, achieving state-of-the-art performance on streaming video understanding.

Our main contributions are:

- We address proactive interaction in SVU through the explicit modeling of observed visual evidence and query-specific response conditions using structured scene graphs, offering a novel and interpretable pathway.
- We design **Response-G1** , an end-to-end, fine-tuning-free framework that seamlessly integrates online scene graph generation, memory-based graph retrieval, and retrieval-augmented streaming decision-making. Our framework, through explicit scene graph modeling and evidence retrieval, yields more accurate response timing decisions.
- Experimental results on established benchmarks show that our method sets a new state-of-the-art for both *proactive* and *reactive* streaming video understanding tasks.

2 Related Works

Streaming Video Understanding. Streaming Video Understanding (SVU) refers to analyzing

continuously arriving video streams without observing the complete video (Zhou et al., 2024; Zheng et al., 2025; Wang et al., 2025c). Recently, Video-LLMs have achieved state-of-the-art performance on these tasks (Tang et al., 2024; Bai et al., 2025). This line of research focuses primarily on modeling dynamic long-term context (Huang et al., 2025) and enabling real-time processing (Zhang et al., 2025a) within a reactive interaction framework, where streaming Video-LLMs must generate responses immediately as users pose queries at arbitrary timestamps during the video stream.

In practice, however, users may often issue queries that require future observations. This necessitates streaming Video-LLMs capable of proactive interaction, which must autonomously decide when the currently observed evidence satisfies the condition to respond, or otherwise remain silent. To achieve query-aware proactive interaction, fine-tuning-based approaches train frame-level decision modules via streaming EOS token prediction (Chen et al., 2024a; Li et al., 2025) or auxiliary activation models (Qian et al., 2025; Wang et al., 2025a), while fine-tuning-free methods primarily rely on prompting to query the Video-LLMs whether to remain silent at each frame (Niu et al., 2025; Yang et al., 2025c).

However, such implicit modeling of visual evidence and response conditions hinders the understanding of response timing. In contrast, we introduce explicit scene graph modeling of both accumulated streaming evidence and query-specific conditions. By feeding the aligned scene graph evidence into the Video-LLM, our method achieves more accurate response timing decisions, within a fine-tuning-free framework.

Scene Graph for Retrieval and Reasoning. Scene graphs provide a structured semantic representation of objects, attributes, and their relations, widely used for visual retrieval (Johnson et al., 2015) and spatio-temporal reasoning (Xiao et al., 2023; Chu et al., 2025). In retrieval, they enable accurate semantic-level similarity measurement between queries and visual databases (Schroeder and Tripathi, 2020; Yoon et al., 2021). For proactive SVU tasks, the user query typically depicts an anticipated scene that includes target objects and relations. Such a scene naturally lends itself to a structured graph representation. Motivated by this, we propose a streaming pipeline that combines online scene graph generation with graph-retrieval augmentation. This allows us to explicitly model and reason over both the visual evidence and the response conditions, toward more accurate response timing decisions.

Scene Graph Generation. Scene Graph Generation (SGG) aims to parse visual inputs into structured object-relation graphs. Traditional methods (Cong et al., 2021; Nguyen et al., 2024) depend on closed-set detectors (e.g., Faster R-CNN (Ren et al., 2015)), limiting open-world applicability. To address this, recent approaches leverage LLMs for open-vocabulary SGG (Li et al., 2024b; Yang et al., 2025b; Nguyen et al., 2025). In our proactive SVU setting, we prompt the Video-LLM itself to generate scene graphs dynamically from the video stream, focusing on salient, query-related evidence.

3 Methodology

We propose **Response-G1** , a novel framework that empowers Video-LLMs with proactive interaction capabilities by explicitly modeling the accumulated evidence and the response conditions through structured scene graphs. As illustrated in Figure 2, our approach comprises three core components: (1) **online query-guided scene graph generation**, (2) **memory-based scene graph retrieval**, and (3) a **retrieval-augmented streaming pipeline**. We begin by formalizing the proactive streaming video understanding problem.

3.1 Problem Formulation

Let $\mathcal{V}_T$ denote a streaming video of duration T, represented as a temporal sequence of sampled frames $\mathcal{F} = \{f_1, f_2, \ldots, f_T\}$. At an arbitrary timestamp $t_{\text{ask}} \in [1, T]$ during the stream, a user issues a query $\mathcal{Q}_{t_{\text{ask}}}$.

Reactive vs. Proactive Paradigms. In the conventional reactive interaction paradigm, the Video-LLM is constrained to produce an immediate response at the query arrival moment. Formally, the response time t_{res} equals t_{ask}, and the answer is generated based solely on the observed prefix $\mathcal{F}_{1:t_{\text{res}}}$.

In contrast, the proactive interaction paradigm allows the model to strategically delay its response until sufficient evidence is accumulated. The response time t_{res} is a latent variable satisfying $t_{\text{ask}} \leq t_{\text{res}} \leq T$, determined dynamically by a per-frame decision function $\mathcal{D}(\cdot)$. At each time step $t \in [t_{\text{ask}}, t_{\text{res}}]$, the model evaluates whether

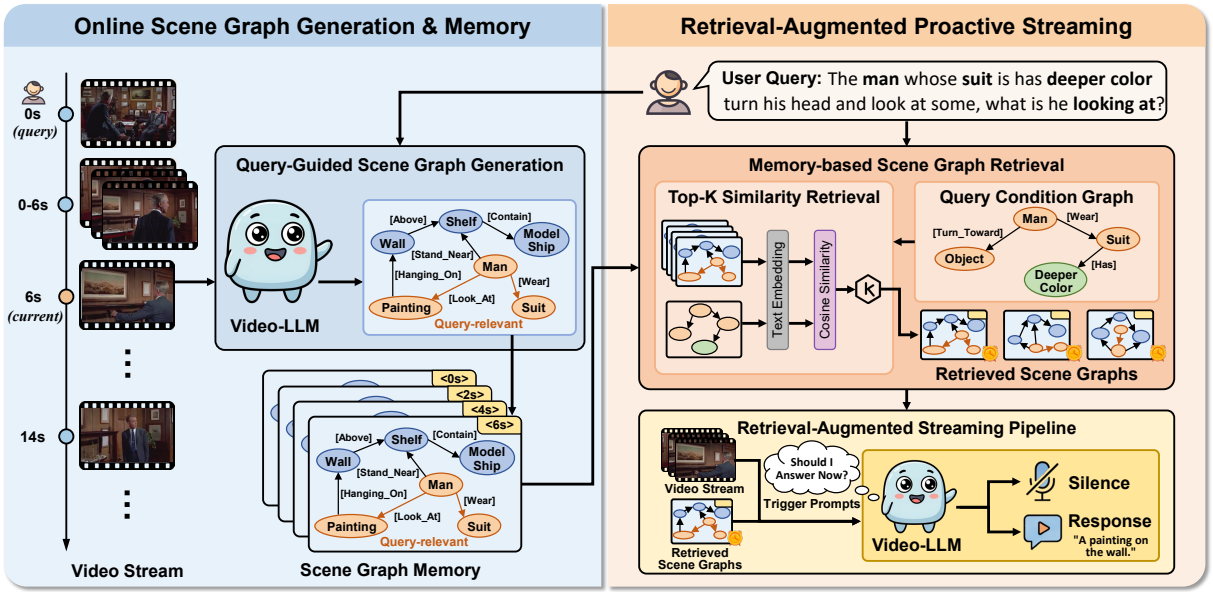

Figure 2: **Overview of the Response-G1 framework.** The system processes streaming video through three core components: (1) Online Query-Guided Scene Graph Generation, (2) Memory-Based Scene Graph Retrieval, and (3) Retrieval-Augmented Streaming Pipeline for proactive decision-making.

the observed evidence in $\mathcal{F}_{1:t}$ satisfies the response conditions implicit in $\mathcal{Q}_{t_{\text{ask}}}$, outputting a proactive action $r_t \in \mathcal{R} = \{\texttt{silence}, \texttt{response}\}$.

Evidence-Condition Modeling via Scene Graphs. To achieve proactive response, we explicitly model both evidence and response conditions using structured scene graphs. Let $\mathcal{J} : (\mathcal{Q}, \mathcal{F}_{1:t}) \mapsto \mathbf{H}_t$ be a joint encoding function that maps the query and video frames into a unified representation space, where $\mathbf{H}_t \in \mathbb{R}^{d_h}$ denotes the joint hidden state. Let $\mathcal{S} : \mathbf{H}_t \mapsto \mathcal{G}_t$ be a scene graph extraction function that produces a structured graph $\mathcal{G}_t$ capturing objects, attributes, and their relations. Our proactive decision function is then formulated as:

$$\mathcal{D}\big(\mathbf{H}_t, \mathcal{S}(\mathbf{H}_t)\big) \to r_t. \tag{1}$$

This explicit modeling enables more principled and interpretable reasoning over evidence-condition alignment, leading to more accurate response timing decisions.

3.2 Online Query-Guided Scene Graph Generation

To explicitly model the evolving visual evidence, we design a one-stage, LLM-based framework for online scene graph generation from streaming video clips.

Formal Scene Graph Representation. For a video clip $\mathcal{C}_t$ centered at timestamp t, we generate a scene graph $\mathcal{G}_t = (\mathcal{O}_t, \mathcal{P}_t)$ via a prompted Video-LLM. Here, $\mathcal{O}_t$ is the set of nodes representing visual objects (e.g., person, car) and their attributes (e.g., red, large). $\mathcal{P}_t$ is the set of predicates (edges) capturing spatio-temporal relations (e.g., next_to, holdin) between node pairs. Therefore, the graph can be represented as a collection of object-predicate-object triplets:

$$\mathcal{G}_t = \{\tau_t^{ij} = (o_t^i, p_t^{ij}, o_t^j) \mid o_t^i, o_t^j \in \mathcal{O}_t; p_t^{ij} \in \mathcal{P}_t\}. \tag{2}$$

Query-Guided Generation for Relevance. To suppress irrelevant visual details and focus on query-salient evidence, we condition the scene graph generation on the user query $\mathcal{Q}$. Specifically, we inject $\mathcal{Q}$ into the generation prompt (see Appendix), steering the Video-LLM to prioritize query-relevant triplets. The query-guided generation process is formalized as:

$$\mathcal{G}_t = \mathcal{S}(\mathcal{C}_t; \mathcal{Q}). \tag{3}$$

This directed generation enhances the relevance of the extracted scene graphs for subsequent evidence-condition matching.

3.3 Memory-Based Scene Graph Retrieval

To achieve fine-grained evidence-condition alignment, we introduce a memory module that stores historical scene graphs and retrieves the most query-relevant ones for semantic matching.

Textual Linearization and Embedding. For efficient retrieval, we linearize each scene graph triplet τ_t^{ij} into a natural language phrase ϕ_t^{ij} (e.g., $(woman, in, red) \rightarrow$ *"woman in red"*). The full graph $\mathcal{G}_t$ is then represented as the concatenation of all its triplet phrases:

$$\Phi_t = \bigoplus_{i,j} \phi_t^{ij}. \tag{4}$$

Similarly, we parse the user query $\mathcal{Q}$ into a query condition graph $\mathcal{G}_q$ (via the same Video-LLM) and derive its textual representation Φ_q. Using Φ_q ensures format consistency with Φ_t for fair similarity computation.

We employ the Video-LLM's text encoder $\mathcal{E}_{\text{text}}(\cdot)$ to obtain dense embeddings. Let $\mathbf{E}_t = \mathcal{E}_{\text{text}}(\Phi_t) \in \mathbb{R}^{n_t \times d}$ and $\mathbf{E}_q = \mathcal{E}_{\text{text}}(\Phi_q) \in \mathbb{R}^{n_q \times d}$, where n_t, n_q are token counts and d is the embedding dimension. The graph representation is obtained via mean pooling over the token dimension:

$$\mathbf{g}_t = \text{MeanPool}(\mathbf{E}_t) \in \mathbb{R}^d, \tag{5}$$

$$\mathbf{g}_q = \text{MeanPool}(\mathbf{E}_q) \in \mathbb{R}^d. \tag{6}$$

Similarity-Based Top-K Retrieval. The semantic relevance between a clip-wise scene graph $\mathcal{G}_t$ and the query $\mathcal{Q}$ is quantified by cosine similarity:

$$\text{sim}(\mathcal{G}_t, \mathcal{Q}) = \frac{\mathbf{g}_t \cdot \mathbf{g}_q}{\|\mathbf{g}_t\| \|\mathbf{g}_q\|}. \tag{7}$$

At each time step t, we maintain a memory bank $\mathcal{M}_t = \{\mathcal{G}_1, \ldots, \mathcal{G}_t\}$ of generated scene graphs up to t. The top-K most relevant graphs are retrieved:

$$\mathcal{G}_t^{\text{ctx}} = \{\mathcal{G}_\tau \mid \tau \in \text{TopK}\big(\{\text{sim}(\mathcal{G}_i, \mathcal{Q})\}_{i=1}^{t}, K\big)\}, \tag{8}$$

where $\mathcal{G}_t^{\text{ctx}}$ serves as the structured, response condition-aligned evidence context for the subsequent decision pipeline.

3.4 Retrieval-Augmented Streaming Pipeline

The final component integrates the retrieved scene graphs into a streaming pipeline that performs per-frame decision-making and response generation.

Trigger Phase: Silence-or-Response Decision. At each time step $t \geq t_{\text{ask}}$, the model decides whether to respond. The input to the Video-LLM is constructed as a token sequence:

$$[\mathbf{f}_1, \ldots, \mathbf{f}_t] \oplus \Psi(\mathcal{G}_t^{\text{ctx}}) \oplus \mathbf{p}_{\text{trg}}, \tag{9}$$

where $\mathbf{f}_i$ denotes the frame embedding for f_i, $\mathbf{p}_{\text{trg}}$ is the embedding of a trigger instruction (e.g., *"Should I answer now? Yes or No."*), and $\Psi(\cdot)$ encodes the retrieved scene graphs with timestamps.

Timestamp-Aware Scene Graph Encoding. To provide temporal context, each retrieved graph $\mathcal{G}_i \in \mathcal{G}_t^{\text{ctx}}$ is prefixed with a textual timestamp token (e.g., `<2.0s>`) before encoding. Formally,

$$\Psi(\mathcal{G}_t^{\text{ctx}}) = \bigoplus_{i \in \mathcal{I}^{\text{ctx}}} \mathcal{E}_{\text{text}}\big(\texttt{<}t_i\texttt{s>} \oplus \Phi_i\big), \tag{10}$$

where $\mathcal{I}^{\text{ctx}}$ is the set of retrieved indices, t_i is the timestamp of $\mathcal{G}_i$, and Φ_i is its textual linearization. This encoding enhances the model's temporal reasoning about the evidence.

The Video-LLM processes this sequence and generates an interaction decision token (e.g., `Yes`/`No`). If the decision is `silence`, the process continues with the next frame. If `response`, the model proceeds to the response phase at $t_{\text{res}} = t$.

Response Phase: Answer Generation. Upon triggering, the final answer is generated using the context up to t_{res}. The input sequence is:

$$[\mathbf{f}_1, \ldots, \mathbf{f}_{t_{\text{res}}}] \oplus \Psi(\mathcal{G}_{t_{\text{res}}}^{\text{ctx}}) \oplus \mathbf{q}, \tag{11}$$

where $\mathbf{q} = \mathcal{E}_{\text{text}}(\mathcal{Q})$ is the embedding of the original user query. The Video-LLM then generates the natural language response.

Extension to Reactive Interaction. The same scene graph augmentation benefits traditional reactive interaction ($t_{\text{res}} = t_{\text{ask}}$). Using Equation 11 with $\mathcal{G}_{t_{\text{ask}}}^{\text{ctx}}$, the model also achieves enhanced spatio-temporal grounding and answer quality through scene graph modeling, demonstrating the framework's versatility.

4 Experiments

4.1 Settings

Datasets. We evaluate **Response-G1** on two established streaming video understanding benchmarks: OVO-Bench (Niu et al., 2025) and StreamingBench (Lin et al., 2024b). The evaluations are conducted in two distinct modes: (i) **Proactive mode**, which aims to validate **Response-G1** 's capability for more accurate response timing decisions. This mode employs the Forward Active Responding subtask in OVO-Bench and the PO (Proactive Output) subtask in StreamingBench, where the model must autonomously decide when to respond during the video stream. (ii) **Reactive mode**, which verifies the versatility of our framework by demonstrating improved spatio-temporal

Model	Params	Real-Time Visual Perception							Backward Tracing				Forward Active Responding				Overall
		OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.	REC	SSR	CRR	Avg.	Avg.
Human																	
Human	-	94.0	92.6	94.8	92.7	91.1	94.0	93.2	92.6	93.0	91.4	92.3	95.5	89.7	93.6	92.9	92.8
Proprietary MLLMs																	
GPT-4o (Hurst et al., 2024)	-	69.1	65.1	65.5	50.0	68.3	63.7	63.6	49.8	71.0	55.4	58.7	27.6	73.2	59.4	53.4	58.6
Gemini 1.5 Pro (Team et al., 2024)	-	87.3	67.0	80.2	54.5	68.3	67.4	70.8	68.6	75.7	52.7	62.3	35.5	74.2	61.7	57.2	65.3
Open-Source Video-LLMs																	
LongVU (Shen et al., 2024)	7B	55.7	49.5	59.5	48.3	68.3	63.0	57.4	43.1	66.2	9.1	39.5	16.6	69.0	60.0	48.5	48.5
InternVL-V2 (Chen et al., 2024b)	8B	68.5	58.7	69.0	44.9	67.3	56.0	60.7	43.1	61.5	27.4	44.0	25.8	57.6	52.9	45.4	50.1
Qwen2-VL (Wang et al., 2024)	7B	69.1	53.2	63.8	50.6	66.3	60.9	60.7	44.4	66.9	34.4	48.6	30.1	65.7	50.8	48.9	52.7
LLaVA-OneVision (Li et al., 2024a)	7B	67.1	58.7	69.8	49.4	71.3	60.3	62.8	52.5	58.8	23.7	45.0	24.8	66.9	60.8	50.9	52.9
LLaVA-NeXT-Video (Liu et al., 2024a)	7B	69.8	59.6	66.4	50.6	72.3	61.4	63.3	51.2	64.2	9.7	41.7	34.1	67.6	60.8	54.2	53.1
Open-Source Streaming Video-LLMs																	
VideoLLM-online (Chen et al., 2024a)	8B	8.1	23.9	12.1	14.0	45.5	21.2	20.8	22.2	18.8	12.2	17.7	-	-	-	-	-
Flash-Vstream (Zhang et al., 2025a)	7B	25.5	32.1	29.3	33.7	29.7	28.8	29.9	36.4	33.8	5.9	25.4	5.4	67.3	60.0	44.2	33.2
Dispider (Qian et al., 2025)	7B	57.7	49.5	62.1	44.9	61.4	51.6	54.5	48.5	55.4	4.3	36.1	18.0	37.4	48.8	34.7	41.8
TimeChat-Online (Yao et al., 2025)	7B	69.8	48.6	64.7	44.9	68.3	55.4	58.6	53.9	62.8	9.1	42.0	32.5	36.5	40.0	36.4	45.6
StreamAgent (Yang et al., 2025c)	7B	71.2	53.2	63.6	53.9	67.3	58.7	61.3	54.8	58.1	25.8	41.7	35.9	48.4	52.0	45.4	49.4
Response-G1 (Ours)	8B	**90.6**	**74.3**	**75.9**	**59.6**	**69.3**	**71.7**	**73.6**	**55.6**	**66.9**	**33.9**	**52.1**	**41.9**	**71.1**	**61.7**	**58.2**	**61.3**

Table 1: Performance comparison on OVO-Bench. Following the offical benchmark settings, the Forward Active Responding subtask is implemented using *proactive* interaction mode, where the model must determine when to respond. The Real-Time Visual Perception and Backward Tracing subtask follow the *reactive* interaction mode. Among open-source streaming Video-LLMs, the **best** and second-best scores are highlighted.

understanding through scene-graph-retrieved augmentation. This mode includes all remaining subtasks of the two benchmarks, where the response timing is pre-specified and coincides with the moment the user query is issued.

Implementation Details. We employ Qwen3-VL-8B (Bai et al., 2025) as our Video-LLM backbone, and follow the input pixel configuration established in prior work (Yao et al., 2025). For OVO-Bench (Niu et al., 2025), evaluations are performed at default 1 FPS. For StreamingBench (Lin et al., 2024b), we follow its official frame-sampling protocol: videos shorter than 300 frames are sampled at 1 FPS, those between 300 and 600 frames at 0.5 FPS, and videos longer than 600 frames at 0.2 FPS. All experiments are run on NVIDIA A100 (80GB) GPUs using FP16 precision.

4.2 Main Results

Proactive Streaming Video Understanding. For the *proactive* evaluation, we report results on the Forward Active Responding subtask in OVO-Bench, namely REC (Repetition Event Count), SSR (Sequential Steps Recognition), and CRR (Clues Reveal Responding), as summarized in Table 1. The results demonstrate our leading performance across these subtasks. Notably, the average score over the three subtasks surpasses the second-best open-source streaming Video-LLM by 12.8%, and even attains a level comparable to proprietary MLLMs. For StreamingBench, the performance on the PO (Proactive Output) subtask is presented in Table 2. **Response-G1** outperforms the second-best open-source streaming Video-LLM by 15.1%. These results further highlight the superiority of **Response-G1** in proactive streaming video understanding tasks.

Reactive Streaming Video Understanding. We also evaluate **Response-G1** on standard *reactive* streaming video understanding tasks. As reported in Table 1, **Response-G1** outperforms the second-best open-source streaming Video-LLM by 12.3% on the Real-Time Visual Perception subtask and by 10.1% on the Backward Tracing subtask in OVO-Bench. In StreamingBench shown in Table 2, it achieves a 2.1% improvement on the Real-Time Visual Understanding subtask. The consistent performance gains indicate that explicit scene graph modeling and retrieval not only lead to more accurate response timing in *proactive* settings, but also deliver substantial benefits for spatio-temporal understanding of objects and their relations on *reactive* subtasks.

4.3 Ablation Study

Impact of Retrieval-Augmented Streaming Pipeline. To verify whether explicit scene-graph modeling enhances the Video-LLM's streaming video understanding, we compare performance with and without graph-based retrieval augmentation. Additionally, we examine the impact of introducing timestamp encoding (refer to §3.4). As shown in Table 3, our explicit scene-graph modeling improves performance on both *reactive* and *proactive* subtasks. Moreover, timestamp-aware scene-graph encoding yields a notable gain on tasks that require temporal grounding, such as CRR.

Model	Params	Real-Time Visual Understanding											PO	Overall Avg.
		OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	All		
Human														
Human	-	89.5	92.0	93.6	91.5	95.7	92.5	88.0	88.8	89.7	91.3	91.5	100	92.0
Proprietary MLLMs														
Claude 3.5 Sonnet (Anthropic, 2024)	-	80.5	77.3	82.0	81.7	72.3	75.4	61.1	61.8	69.3	43.1	72.4	64.7	69.9
GPT-4o (Hurst et al., 2024)	-	77.1	80.5	83.9	76.5	70.2	83.8	66.7	62.2	69.1	49.2	73.3	56.9	70.5
Gemini 1.5 pro (Team et al., 2024)	-	79.0	80.5	83.5	79.7	80.0	84.7	77.8	64.2	72.0	48.7	75.7	45.1	72.3
Open-Source Video-LLMs														
Video-LLaMa2 (Cheng et al., 2024)	7B	55.9	55.5	57.4	58.2	52.8	43.6	39.8	42.7	45.6	35.2	49.5	0.0	44.2
VILA-1.5 (Lin et al., 2024a)	8B	53.7	49.2	71.0	56.9	43.4	53.9	54.6	48.8	50.1	17.6	52.3	17.7	47.0
Video-CCAM (Fei et al., 2024)	14B	56.4	57.8	65.3	62.8	64.6	51.4	42.6	48.0	49.6	31.6	54.0	22.7	50.2
LongVA (Zhang et al., 2024)	7B	70.0	63.3	61.2	70.9	62.7	59.5	61.1	53.7	54.7	34.7	60.0	15.9	55.2
InternVL-V2 (Chen et al., 2024b)	8B	68.1	60.9	69.4	77.1	67.7	62.9	59.3	53.3	55.0	56.5	63.7	40.9	61.0
Kangaroo (Liu et al., 2024b)	7B	71.1	84.4	70.7	73.2	67.1	61.7	56.5	55.7	62.0	38.9	64.6	16.0	59.7
LLaVA-NeXT-Video (Liu et al., 2024a)	32B	78.2	70.3	73.8	76.8	63.4	69.8	57.4	56.1	64.3	38.9	67.0	18.2	60.6
MiniCPM-V-2.6 (Hu et al., 2024)	8B	71.9	71.1	77.9	75.8	64.6	65.7	70.4	56.1	62.3	53.4	67.4	22.2	62.9
LLaVA-OneVision (Li et al., 2024a)	7B	80.4	74.2	76.0	80.7	72.7	71.7	67.6	65.5	65.7	45.1	71.1	29.6	66.3
Qwen2-VL (Wang et al., 2024)	7B	75.2	82.8	73.2	77.5	68.3	71.0	72.2	61.2	61.5	46.1	69.0	22.7	64.7
Open-Source Streaming Video-LLMs														
VideoLLM-online (Chen et al., 2024a)	8B	39.1	40.1	34.5	31.1	46.0	32.4	31.5	34.2	42.5	27.9	36.0	3.9	33.0
Flash-Vstream (Zhang et al., 2025a)	7B	25.9	43.6	24.9	23.9	27.3	13.1	18.5	25.2	23.9	48.7	23.2	2.0	25.2
Dispider (Qian et al., 2025)	7B	74.9	75.5	74.1	73.1	74.4	59.9	76.1	62.9	62.2	45.8	67.6	25.3	64.0
TimeChat-Online (Yao et al., 2025)	7B	80.2	**82.0**	79.5	83.3	**76.1**	78.5	78.7	64.6	69.6	**58.0**	75.4	28.8	70.9
StreamAgent (Yang et al., 2025c)	7B	79.6	78.3	79.3	75.9	74.7	76.9	82.9	**66.3**	**73.7**	55.4	74.3	28.9	70.2
Response-G1 (Ours)	8B	**84.0**	78.1	**88.0**	**84.6**	74.8	**83.5**	**83.3**	63.0	69.3	**58.0**	**77.5**	**44.0**	**73.7**

Table 2: Performance comparison on StreamingBench. Following the official benchmark settings, the PO (Proactive Output) subtask implemented using *proactive* interaction mode. All other subtasks follow the *reactive* interaction mode. Among open-source streaming Video-LLMs, the **best** and second-best scores are highlighted.

Strategies	OVO-Bench			StreamingBench		
	ACR	HLD	CRR	CS	PR	PO
W/o Retrieval Augmentation	66.1	28.0	55.4	83.6	79.6	36.8
W/o Timestamp Encoding	74.0	33.6	60.4	87.7	82.9	43.6
Full	**74.3**	**33.9**	**61.7**	**88.0**	**83.3**	**44.0**

Table 3: Performance of different retrieval-augmented streaming inference strategies. Evaluations are conducted on *proactive* subtasks (CRR, PO) and *reactive* subtasks (ACR, HLD, CS, PR).

Strategies	Proactive Subtasks			
	PO	REC	SSR	CRR
W/o Guidance	38.8	34.1	66.9	59.4
Object-Guidance	43.6	40.2	67.9	61.3
Query-Guidance	**44.0**	**41.9**	**71.1**	**61.7**

Table 4: Performance of different guidance strategies for online video scene graph generation. Evaluations are conducted on *proactive* subtasks.

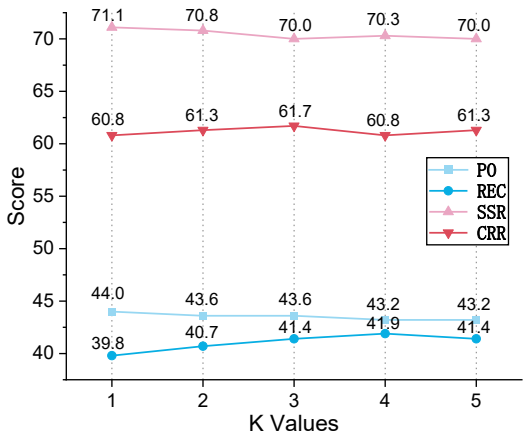

Figure 3: Performance of different K Values for top-K similarity-based scene graph retrieval. Evaluations are conducted on *proactive* subtasks.

Configurations of Online Query-Guided Scene Graph Generation. As outlined in §3.2, we inject the user query into the prompts for online video scene graph generation to guide the Video-LLM toward query-relevant descriptions and reduce redundant triplet generation (i.e., "Query-Guidance"). The effectiveness of this design is validated through performance comparisons on *proactive* subtasks. We also explore an alternative guidance strategy that directly injects parsed objects and relations into the prompts for scene graph generation (i.e., "Object-Guidance"). We provide the specific prompts and cases in the Appendix. As shown in Table 4, the query-guided generation strategy yields the best performance, confirming its efficacy. Failure cases in "Object-Guidance" reveal that direct object injection may lead to hallucination, where the model over-focuses on anticipated objects and generates non-existent triplets, causing premature responses and performance drops. This underscores the importance of balancing query relevance with factuality in LLM-based SGG.

Effectiveness of Memory-Based Scene Graph Retrieval. We evaluate the effectiveness of the

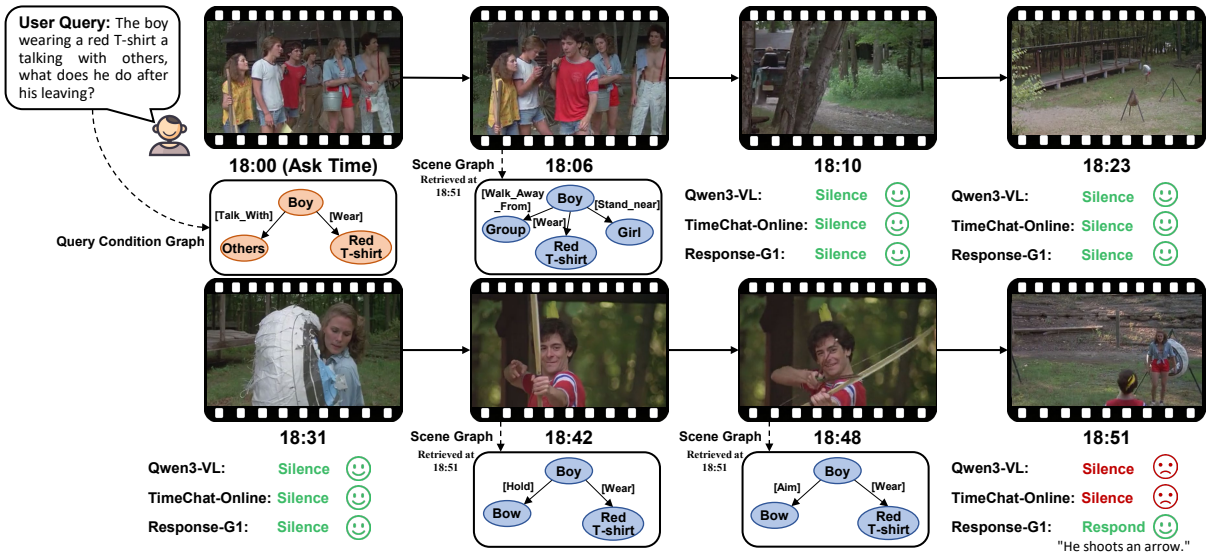

Figure 4: Case study of **Response-G1** on the CRR subtask in OVO-Bench. The user query describes a target object ("the boy wearing a red T-shirt") and a relation ("talking with others"). The results show that at time "18:51", **Response-G1** accurately retrieves query-relevant scene graphs (i.e., evidence) and triggers a response, whereas the baselines fail to respond throughout the video stream.

Strategies	Proactive Subtasks			
	PO	REC	SSR	CRR
Original Query Text	42.4	40.2	69.3	56.0
Query Graph Text	**44.0**	**41.9**	**71.1**	**61.7**

Table 5: Performance of different query embedding strategies for similarity-based scene graph retrieval. Evaluations are conducted on *proactive* subtasks.

proposed memory-based scene graph retrieval described in §3.3. Specifically, we compare the performance of using original query text versus the proposed graph text for embedding similarity calculation on *proactive* subtasks, as shown in Table 5. The results indicate that using the original query text for embedding similarity leads to format inconsistency between the query and the video scene graph, which consequently degrades both retrieval quality and downstream task performance. We provide the specific cases in the Appendix.

We further analyze the impact of the top-K parameter in *proactive* settings, as shown in Figure 3. The results reveal that overall performance remains relatively stable across different K values. Moreover, for tasks that primarily focus on latest-frame information (e.g., SSR and PO), $K = 1$ is sufficient to achieve superior performance.

4.4 Case Study

We conduct a case study to visualize our explicit scene graph modeling and graph-retrieval-augmented proactive streaming pipeline. For comparison, we select two advanced open-source Video-LLMs: Qwen3-VL (Yang et al., 2025a) and TimeChat-Online (Yao et al., 2025). The case is drawn from the CRR (Clues Reveal Responding) subtask in OVO-Bench. During the stream, each model is evaluated on whether it correctly remains silent or responds at the appropriate time. As illustrated in Figure 4, **Response-G1** achieves better understanding and decision-making of whether the accumulated evidence satisfies the query-specific response conditions through explicit scene graph modeling and retrieval. Moreover, it illustrates how our method provides more interpretable evidence retrieval for response timing decisions.

5 Conclusion

We introduce **Response-G1**, a scene-graph-driven pipeline for proactive streaming video understanding. By modeling both visual evidence and query-specific response conditions in an explicit, structured scene graph representation, our method enables Video-LLMs to achieve better understanding of evidence-condition alignment and more accurate response timing decisions, within a fine-tuning-free manner. Superior results across established benchmarks illustrate that explicit structural modeling can be a powerful principle for proactive interaction. We hope this work opens promising avenues toward more capable and interpretable multimodal interaction in real-world streaming applications.

Limitations

First, explicit scene graph modeling improves evidence-condition alignment, but its object-relation representation and similarity-based retrieval do not fully address all reasoning needs (e.g., "why"-style questions), where benefits may be limited. Future work could explore richer structural formulations (e.g., causal relations) and more memory and retrieval designs beyond top-K matching. Second, the fixed clip size for online scene graph generation could be improved by incorporating event-level or semantics-level perception. Adaptive mechanisms that dynamically determine when to trigger generation and how many frames to include would enhance both efficiency and temporal modeling. Third, LLM-based open-set scene graph generation avoids predefined vocabularies but faces a trade-off between query relevance and hallucination. While we rely on fine-tuning-free prompting, task-specific fine-tuning of the generator could yield more relevance-factuality balancing.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. U25B2042, 62532009, 62232004, 62332016, 62302259), and the Research Grants Council of HKSAR under grant number AoE/E-601/24-N.

References

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, and 1 others. 2025. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*.

Anthropic. 2024. Claude 3.5 sonnet. https://www.anthropic.com/news/claude-3-5-sonnet.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. 2024a. Videollm-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18407–18418.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, and 1 others. 2024b. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67(12):220101.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and 1 others. 2024. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*.

Sanghyeok Chu, Seonguk Seo, and Bohyung Han. 2025. Fine-grained captioning of long videos through scene graph consolidation. In *Forty-second International Conference on Machine Learning*.

Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. 2021. Spatial-temporal transformer for dynamic scene graph generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16372–16382.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Tao Zhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. 2025. Streaming video question-answering with in-context video kv-cache retrieval. *arXiv preprint arXiv:2503.00540*.

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

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, and 1 others. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. 2025. Online video understanding: Ovbench and videochat-online. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3328–3338.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. 2015. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang,

Yanwei Li, Ziwei Liu, and 1 others. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Rongjie Li, Songyang Zhang, Dahua Lin, Kai Chen, and Xuming He. 2024b. From pixels to graphs: Open-vocabulary scene graph generation with vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 28076–28086.

Wei Li, Bing Hu, Rui Shao, Leyang Shen, and Liqiang Nie. 2025. Lion-fs: Fast & slow video-language thinker as online video assistant. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3240–3251.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. 2024a. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26689–26699.

Junming Lin, Zheng Fang, Chi Chen, Zihao Wan, Fuwen Luo, Peng Li, Yang Liu, and Maosong Sun. 2024b. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. *arXiv preprint arXiv:2411.03628*.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. Llavanext: Improved reasoning, ocr, and world knowledge.

Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, Xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. 2024b. Kangaroo: A powerful video-language model supporting long-context video input. *arXiv preprint arXiv:2408.15542*.

Trong-Thuan Nguyen, Pha Nguyen, Jackson Cothren, Alper Yilmaz, and Khoa Luu. 2025. Hyperglm: Hypergraph for video scene graph generation and anticipation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29150–29160.

Trong-Thuan Nguyen, Pha Nguyen, and Khoa Luu. 2024. Hig: Hierarchical interlacement graph approach to scene graph generation in video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18384–18394.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, and 1 others. 2025. Ovo-bench: How far is your video-llms from real-world online video understanding? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18902–18913.

Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. 2025. Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24045–24055.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. 2024. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37:119336–119360.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Brigit Schroeder and Subarna Tripathi. 2020. Structured query-based image retrieval using scene graphs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 178–179.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, and 1 others. 2024. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*.

Jiaqi Tang, Hao Lu, Ruizheng Wu, Xiaogang Xu, Ke Ma, Cheng Fang, Bin Guo, Jiangbo Lu, Qifeng Chen, and Ying-Cong Chen. 2024. Hawk: Learning to understand open-world video anomalies. *Advances in Neural Information Processing Systems*, 37:139751–139785.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Haibo Wang, Bo Feng, Zhengfeng Lai, Mingze Xu, Shiyu Li, Weifeng Ge, Afshin Dehghan, Meng Cao, and Ping Huang. 2025a. Streambridge: Turning your offline video large language model into a proactive streaming assistant. *arXiv preprint arXiv:2505.05467*.

Hao Wang, Bin Guo, Mengqi Chen, Qiuyun Zhang, Yasan Ding, Ying Zhang, and Zhiwen Yu. 2025b. Cascade context-oriented spatio-temporal attention network for efficient and fine-grained video-grounded dialogues. *Frontiers of Computer Science*, 19(7):197329.

Hao Wang, Bin Guo, Yating Zeng, Mengqi Chen, Yasan Ding, Ying Zhang, Lina Yao, and Zhiwen Yu. 2025c. Enabling harmonious human-machine interaction with visual-context augmented dialogue system: A review. *ACM Transactions on Information Systems*, 43(3):1–59.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin

Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Zichen Wen, Yiyu Wang, Chenfei Liao, Boxue Yang, Junxian Li, Weifeng Liu, Haocong He, Bolong Feng, Xuyang Liu, Yuanhuiyi Lyu, and 1 others. 2025. Ai for service: Proactive assistance with ai glasses. *arXiv preprint arXiv:2510.14359*.

Junbin Xiao, Pan Zhou, Angela Yao, Yicong Li, Richang Hong, Shuicheng Yan, and Tat-Seng Chua. 2023. Contrastive video question answering via video graph transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13265–13280.

Haomiao Xiong, Zongxin Yang, Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Jiawen Zhu, and Huchuan Lu. 2025. Streaming video understanding and multi-round interaction with memory-enhanced knowledge. *arXiv preprint arXiv:2501.13468*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Dongil Yang, Minjin Kim, Sunghwan Mac Kim, Beongwoo Kwak, Minjun Park, Jinseok Hong, Woontack Woo, and Jinyoung Yeo. 2025b. Llm meets scene graph: Can large language models understand and generate scene graphs? a benchmark and empirical study. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21335–21360.

Haolin Yang, Feilong Tang, Lingxiao Zhao, Xiang An, Ming Hu, Huifa Li, Xinlin Zhuang, Yifan Lu, Xiaofeng Zhang, Abdalla Swikir, and 1 others. 2025c. Streamagent: Towards anticipatory agents for streaming video understanding. *arXiv preprint arXiv:2508.01875*.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, and 1 others. 2025. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 10807–10816.

Sangwoong Yoon, Woo Young Kang, Sungwook Jeon, SeongEun Lee, Changjin Han, Jonghun Park, and Eun-Sol Kim. 2021. Image-to-image retrieval by learning similarity between scene graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10718–10726.

Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, and 1 others. 2025. Streamforest: Efficient online video understanding with persistent event memory. *arXiv preprint arXiv:2509.24871*.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. 2025a. Flash-vstream: Efficient real-time understanding for long video streams. *arXiv preprint arXiv:2506.23825*.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. 2024. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*.

Yulin Zhang, Cheng Shi, Yang Wang, and Sibei Yang. 2025b. Eyes wide open: Ego proactive video-llm for streaming video. *arXiv preprint arXiv:2510.14560*.

Minghang Zheng, Yuxin Peng, Benyuan Sun, Yi Yang, and Yang Liu. 2025. Hierarchical event memory for accurate and low-latency online video temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21589–21599.

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

Appendix

A Latency Analysis

To assess real-world feasibility, we implement latency analysis of **Response-G1** and a naive Qwen3-VL-8B baseline on the PO subtask in StreamingBench (1 FPS sampling). Latency is defined as the average per-frame execution time from frame input to scene graph generation (if applicable), scene graph retrieval (if applicable), and trigger decision upon each frame arrival. Maximum FPS, computed as $1s$/Total Latency, represents the maximum sampling rate supported by the given configuration.

Additionally, we incorporate a streaming KV-Cache mechanism: instead of storing historical frames as visual embeddings, the memory bank stores them as KV caches, avoiding redundant computation during streaming inference.

As shown in Table 6, **Response-G1** achieves a maximum FPS of 1.2, which satisfies the 1 FPS sampling rate. With the streaming KV-Cache mechanism, the maximum FPS further increases to 2.1 without sacrificing accuracy, confirming the real-world feasibility of our method.

Model	Memory	SGG (ms)	SGR (ms)	Trigger (ms)	Total (ms)	Maximum FPS
Qwen3-VL-8B	Embedding	\	\	324	324	3.1
	KV-Cache	\	\	182	182	5.5
Response-G1	Embedding	448	21	356	825	1.2
	KV-Cache	249	20	204	473	2.1

Table 6: Latency analysis on the PO subtask. SGG: Scene Graph Generation; SGR: Scene Graph Retrieval.

B Evaluation Across Architectures

We extend our evaluation to LLaVA-OneVision-1.5-8B (An et al., 2025), the latest open-source model in the LLaVA-OneVision series. As shown in Table 7, which presents a comparison among open-source streaming Video-LLMs, **Response-G1** (i.e., Ours) consistently achieves leading performance on both benchmarks when deployed across different architectures. These results demonstrate the generalizability of our scene-graph-driven approach across diverse Video-LLM architectures.

C Prompts

This section provides the detailed prompts used in **Response-G1** for (i) scene graph generation (video clip → textual scene graph), (ii) query parsing (user query → textual query condition graph), and (iii) graph-retrieval-augmented streaming trigger.

Model		OVO-Bench				StreamingBench		
		RTVP	BT	FAR	Overall	RTVU	PO	Overall
VideoLLM-online		20.8	17.7	-	-	36.0	3.9	33.0
Flash-Vstream		29.9	25.4	44.2	33.2	23.2	2.0	25.2
Dispider		54.5	36.1	34.7	41.8	67.6	25.3	64.0
TimeChat-Online		58.6	42.0	36.4	45.6	75.4	28.8	70.9
StreamAgent		61.3	41.7	45.4	49.4	74.3	28.9	70.2
Ours	**LLaVA-OV-1.5-8B**	66.7	48.3	54.8	56.1	74.8	35.6	71.2
	Qwen3-VL-8B	**73.6**	**52.1**	**58.2**	**61.3**	**77.5**	**44.0**	**73.7**

Table 7: Performance comparison among open-source streaming Video-LLMs on OVO-Bench and StreamingBench. RTVP: Real-Time Visual Perception; BT: Backward Tracing; FAR: Forward Active Responding; RTVU: Real-Time Visual Understanding; PO: Proactive Output.

Scene Graph Generation Prompts As described in §3.2, we perform query-guided scene graph generation on streaming video clips. Figure 5 illustrates the complete prompt template used for this process.

In Table 4, we validate three guidance strategies for scene graph generation. Below, we illustrate each using a concrete case from the PO subtask in StreamingBench, where the query asks to respond when `"the number 20 appears in the middle of the sun"`.

- "W/o Guidance" relies solely on video clips (`{query} None`). Lacking query awareness, it may generate irrelevant triplets (e.g., `[grass, on, ground]`), introducing noise for retrieval.
- "Object-Guidance" takes parsed query elements as input (`{query} objects: Sun, 20. relations: appear_in`). Over-focusing on given elements may cause hallucination (e.g., generating the scene graph triplet `[number 20, appears_in, sun]` before evidence actually appears), leading to premature triggering.
- "Query-Guidance" uses the original query (`{query} When the number 20 appears in the middle of the sun in the video, output "20"`). Experimental results show that it maintains query relevance in scene graph generation while avoiding overcommitment to objects not yet visible, leading to more accurate response timing decisions.

Query Parsing Prompts As described in §3.3, we perform scene graph retrieval to enable explicit evidence-condition alignment for subsequent trigger decisions. However, as shown in Table 5, directly using the original query text (e.g., `"the number 20 appears in the middle of the`

sun") leads to format inconsistency between the video scene graph and the user query, which degrades retrieval and downstream task performance. To address this, we also parse the user query into a unified structured graph representation (e.g., `[number 20, appears_in, sun]`) immediately upon issuance, thereby eliminating the format inconsistency that hinders similarity-based retrieval. Figure 6 provides the complete prompt template for this query parsing process.

Retrieval-Augmented Trigger Prompts As described in §3.4, **Response-G1** performs retrieval-augmented trigger decisions. Figures 7, 8, 9, and 10 show the trigger decision prompt templates for the CRR subtask in OVO-Bench (original and ours) and the PO subtask in StreamingBench (original and ours), respectively.

Prompts

```
As a scene graph analysis expert, please analyze the current video clip and
generate a focused Scene Graph that is most relevant to the user's query.


**User Query:** "{query}"


**Analysis Guidelines:**
- First, objectively describe what is actually present in the clip
- Never invent objects or relationships that are not actually visible
- Do NOT include duplicate triples with identical meaning


**Scene Graph**
- Format: [Subject, Relation, Object]
- Include: Objects, their spatial relationships (near, left_of, right_of,
above, below, etc), and actions (holding, walking, running, hitting, etc)

Please only output the scene graph triplets in the following format:
Scene Graph:
    1. [person, holding, cup]
    2. [dog, running_towards, park]
    3. [car, parked_near, building]
    ...
```

Figure 5: Prompt template for query-guided online scene graph generation.

Prompts

```
As a scene graph analysis expert, please parse the user query into a
structured scene graph that explicitly captures its anticipated visual scene
and response conditions.

**User Query:** "{query}"

**Analysis Guidelines:**
- Identify all objects, attributes, and relationships explicitly mentioned or
logically implied in the query
- Focus on elements that define the visual conditions required for a valid
response
- Maintain semantic faithfulness to the original query intent
- Do NOT include duplicate triples with identical meaning

**Scene Graph**
- Format: [Subject, Relation, Object]
- Include: Objects, their spatial relationships (near, left_of, right_of,
above, below, etc), and actions (holding, walking, running, hitting, etc)

Please only output the scene graph triplets in the following format:
Scene Graph:
    1. [person, holding, cup]
    2. [dog, running_towards, park]
    3. [car, parked_near, building]
    ...
```

Figure 6: Prompt template for query parsing.

Prompts

```
You're responsible of answering questions based on the video content.

The following question are relevant to the latest frames, i.e. the end of the video.

{query}

Decide whether existing visual content, especially latest frames, i.e. frames that near the end of the video, provide enough information for answering the question.

Answer only with "Yes" or "No".

Do not include any additional text or explanation in your response.
```

Figure 7: Prompt template for the original trigger on the CRR subtask in OVO-Bench.

Prompts

```
You're responsible of answering questions based on the video content.

{retrieved_textual_scene_graphs}

The following question are relevant to the latest frames, i.e. the end of the video.

{query}

Decide whether existing visual content, especially latest frames, i.e. frames that near the end of the video, provide enough information for answering the question.

Answer only with "Yes" or "No".

Do not include any additional text or explanation in your response.
```

Figure 8: Prompt template for **Response-G1**'s trigger on the CRR subtask in OVO-Bench.

Prompts

```
{query}

Is it the right time to output \"{ground_truth_output}\"?

You can only answer yes or no.
```

Figure 9: Prompt template for the original trigger on the PO subtask in StreamingBench.

Prompts

```
{query}

Is it the right time to output \"{ground_truth_output}\"?

{retrieved_textual_scene_graphs}

You can only answer yes or no.
```

Figure 10: Prompt template for **Response-G1**'s trigger on the PO subtask in StreamingBench.