Towards Anticipatory Agents for Streaming Video Understanding

Haolin Yang[1]* **Feilong Tang**[1]* **Lingxiao Zhao**[2]* **Xinlin Zhuang**[1] **Yifan Lu**[1]
Xiang An[3] **Ming Hu**[1] **Xiaofeng Zhang**[4] **Abdalla Swikir**[1] **Junjun He**[2]
Zongyuan Ge[5] **Muhammad Haris Khan**[1] **Imran Razzak**[1]†

[1]MBZUAI [2]Shanghai AI Laboratory [3]DeepGlint
[4]Shanghai Jiao Tong University [5]Monash University

{haolin.yang, imran.razzak}@mbzuai.ac.ae

Abstract

Real-time streaming video understanding requires continuous perception, proactive decision making, and responsive interaction with dynamically evolving visual content, posing challenges beyond conventional offline video processing. However, existing methods rely on alternating perception-reaction or asynchronous triggers, lacking *task-driven planning* and *future anticipation*, which limits their real-time responsiveness in evolving video streams. To this end, we propose a **StreamAgent** that anticipates the temporal intervals and spatial regions expected to contain future task-relevant information to enable proactive and goal-driven responses. Specifically, we integrate question semantics and historical observations through prompting the anticipatory agent to anticipate the temporal progression of key events, align current observations with the expected future evidence, and subsequently adjust the perception action (*e.g.,* attending to task-relevant regions or continuously tracking in subsequent frames). To enable efficient inference, we design a *streaming KV-cache memory* mechanism that constructs a hierarchical memory structure for selective recall of relevant tokens, enabling efficient semantic retrieval while reducing the overhead of storing all tokens in the traditional KV-cache. Extensive experiments on streaming and long video understanding tasks demonstrate that our method outperforms existing methods in response accuracy and real-time efficiency, highlighting its practical value for real-world streaming scenarios.

1 Introduction

As vision models are increasingly deployed in real-world scenarios (*e.g.,* autonomous driving and intelligent surveillance), research has progressively shifted towards understanding continuous video streams. Streaming video understanding requires continuous processing of incoming frames, efficient extraction of critical information, proactive decision-making, and responsive interaction with dynamically evolving visual content, guided by explicitly defined queries.

Recent works such as VideoStreaming (Qian et al., 2024), Flash-VStream (Zhang et al., 2024a), and ReKV (Di et al., 2025) have explored multimodal large language models (MLLMs) (Touvron et al., 2023; OpenAI, 2023b) for streaming video understanding, leveraging memory-based mechanisms to process long video streams and support real-time scene understanding. However, these models lack a proactive response mechanism to determine whether to respond immediately or continue observing. Therefore, We argue that beyond continuous perception, the key missing capability in existing streaming systems is *anticipation*, the ability to proactively predicting and actively acquiring future task-relevant information, rather than passively consuming incoming frames.

To enable proactive interaction, VideoLLM-online (Chen et al., 2024a) adopts an alternating "perception–reaction" mechanism (Fig. 1a), where a single LLM handles both perception and generation. However, the autoregressive nature of LLMs prevents parallel execution, delaying frame processing and reducing responsiveness. Dispider (Qian et al., 2025) addresses this by decoupling decision and reaction with an asynchronous binary trigger (Fig. 1b), yet still lacks task-driven planning and future anticipation. As a result, it prematurely responds with incomplete evidence (*e.g.,* predicting "`wall`" instead of "`painting`"), a failure rooted in insufficient temporal reasoning.

In this work, we propose a **StreamAgent** that anticipates the temporal intervals and spatial regions expected to contain future task-relevant information to enable proactive and goal-driven responses, as illustrated in Fig. 1 (c). Specifically, the agent integrates question semantics with historical ob-

* Equal contribution.
† Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 1315–1332
October 24-29, 2026 ©2026 Association for Computational Linguistics

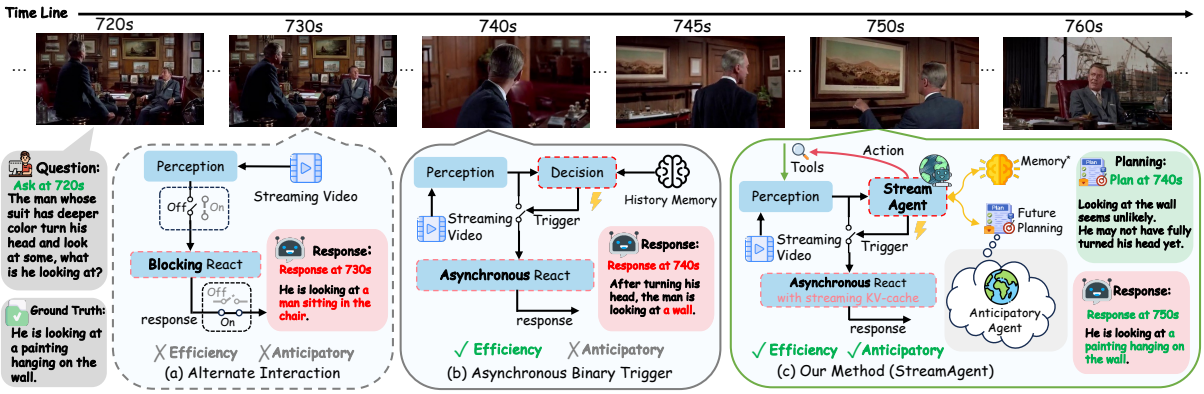

Figure 1: **Comparison between StreamAgent and existing methods.** Prior online methods (Chen et al., 2024a; Qian et al., 2025) enable proactive interactions but rely on either (a) alternating interaction, causing slow processing, or (b) asynchronous binary triggers, leading to inaccurate responses. In contrast, **StreamAgent** integrates continuous perception with task-driven planning and future anticipation, enabling proactive identification of key temporal and spatial cues, and supporting efficient asynchronous reaction through the **streaming KV-cache**.

servations to predict the temporal progression and spatial locations of key events, aligning the current observation with the anticipated trajectory to determine whether sufficient information has been accumulated to trigger a response. If not, StreamAgent proactively refines its perception strategy (e.g., attending to task-relevant regions or persistently tracking targets across subsequent frames). As new video streams arrive, the agent iteratively updates its spatiotemporal focus to gather sufficient evidence for accurate responses, while an incremental memory update mechanism continuously integrates new information to ensure timely and coherent responses throughout the stream.

To address the long-context bottleneck in streaming inference (Di et al., 2025), we further propose a **streaming KV-cache** that constructs a hierarchical memory structure for adaptive context retrieval. Each video clip is encoded into KV-cache via chunk-wise incremental prefill and offloaded to CPU as long-term memory. Upon a query, relevant entries are dynamically retrieved via layer-adaptive selection based on attention patterns, ensuring that only semantically pertinent content is recalled while alleviating GPU memory constraints.

We further validate our method through zero-shot evaluation on both streaming and offline video understanding benchmarks. In summary, our key contributions are as follows:

- We propose **StreamAgent** for streaming video understanding that anticipates future task-relevant temporal intervals and spatial regions to enable proactive responses.
- We propose a **streaming KV-cache** that constructs a hierarchical memory structure, designed to enable efficient semantic retrieval and *selective recall* of relevant tokens.
- Experiments demonstrate that our method achieves state-of-the-art performance on streaming and long video understanding benchmarks.

2 Related Work

Streaming Video Understanding. Unlike traditional video understanding benchmarks (Carreira and Zisserman, 2017; Alamri et al., 2019; Mangalam et al., 2023) that operate on sampled clips, streaming video understanding requires real-time perception, proactive decision-making, and responsive interaction over long video inputs. Early efforts such as VideoLLM-Online (Chen et al., 2024a) and Flash-VStream (Zhang et al., 2024a) explore online models that process frames via memory-based mechanisms. To further disentangle the pipeline, Dispider (Qian et al., 2025) introduces an asynchronous binary trigger that separates decision from reaction. More recent studies extend streaming video understanding along complementary axes, including streaming dialogue frameworks (Chatterjee et al., 2025; Xiong et al., 2025; Liu et al., 2025a; Chen et al., 2025b), continuous perception under evolving content (Wu et al., 2024c; Zhang et al., 2024d; Wang et al., 2025a), and temporal multi-turn interaction benchmarks (Yang et al., 2025b; Wu et al., 2024a; Li et al., 2025a). However, these methods either block perception during generation or rely on binary triggers without task-driven planning, limiting their ability to anticipate task-relevant information.

Agentic Frameworks for Video Understanding. Recent studies have explored agentic approaches

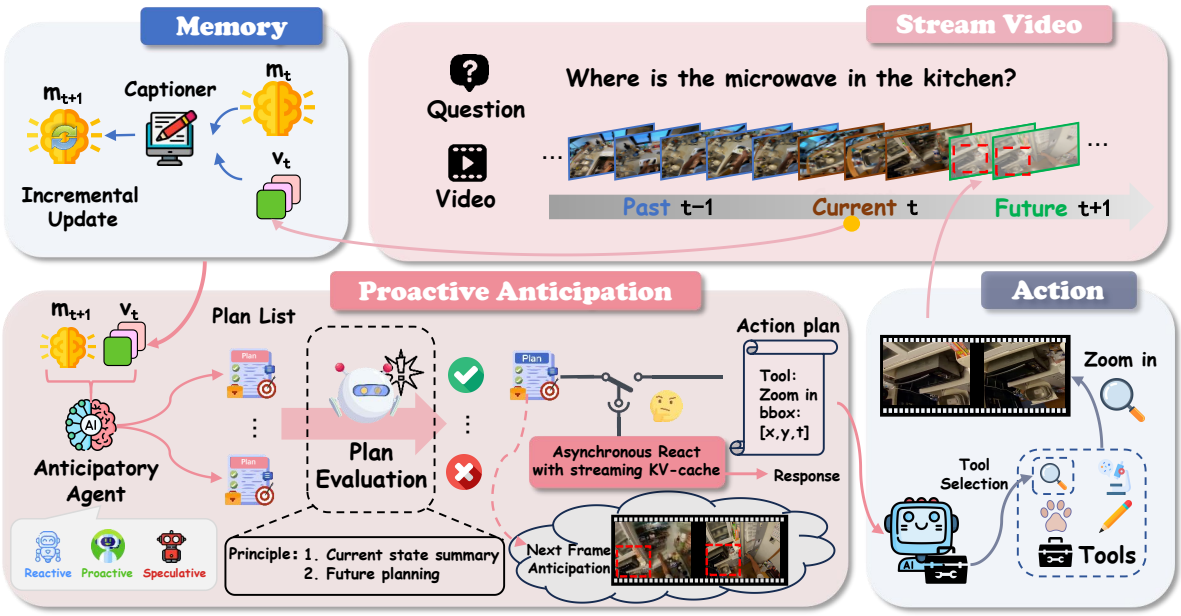

Figure 2: **Overview of the StreamAgent framework:** At each timestep, the system incrementally updates memory based on the current streaming video clip v_t and memory m_t. Conditioned on the updated memory and current observation, the planner generates multiple future-aware plans through proactive anticipation simulating incoming videos, spanning three perspectives, *Reactive, Proactive, and Speculative*. These plans are evaluated jointly considering current situational awareness and predicted future utility. The selected plan either triggers asynchronous reactive behavior or initiates proactive information hunting by invoking tools (*e.g.,* zooming in task-relevant regions in subsequent frames), enabling goal-directed and proactive responses.

that leverage MLLMs to tackle complex video understanding tasks (Zhi et al., 2025; Kugo et al., 2025; Yuan et al., 2025; Chen et al., 2025a; Jeoung et al., 2024; Zhang et al., 2024c). These methods typically lack the capacity for forward-looking anticipation needed in real-time streaming scenarios. In contrast, our StreamAgent integrates continuous perception with anticipatory planning and tool-augmented active perception, enabling goal-directed and efficient responses over streaming video without exhaustive frame-level processing.

KV-Cache Optimization for Video Understanding. Processing long video streams incurs substantial KV-cache overhead. Recent methods address this through KV-cache compression (Ning et al., 2025; Chatterjee et al., 2025), visual feature compression (Zhang et al., 2024b; Wu et al., 2024c), or attention-based token pruning (Schneider et al., 2025; Di et al., 2025). However, these approaches rely on static attention patterns for token selection, limiting adaptability across transformer layers. Our streaming KV-cache adopts a layer-adaptive retrieval strategy, allowing each layer to retrieve a variable number of KV entries tailored to its attention behavior.

3 Methodology

3.1 Overview of StreamAgent Framework

To tackle the challenges of Streaming VideoQA, we propose StreamAgent, a decision agent that anticipates future temporal intervals and spatial regions to contain task-relevant information, enabling proactive responses to user queries, as illustrated in Fig. 2. Details of the proposed decision agent are elaborated in the following sections.

Problem Formulation. Streaming VideoQA (Ding et al., 2025) requires answering proactive, context-sensitive, and real-time questions over video streams (Ning et al., 2024; Guo et al., 2025). Let $\mathcal{V}^T = \{v_t\}_{t=1}^T$ denote a video stream up to time T, where v_t is the t-th clip, and let Q_t denote a question posed at timestamp t. The goal is to leverage the current clip v_t and past video $\mathcal{V}_{1:t-1}$ to answer Q_t. When $\mathcal{V}_{1:t}$ is insufficient, the model should proactively defer and respond at a future timestamp t' ($t \leq t' \leq T$) once $\mathcal{V}_{t+1:t'}$ provides sufficient evidence. We define a decision function $D(Q_t, \mathcal{V}_{1:t})$ that determines whether to wait or respond, and a response function f that is invoked only when sufficient information has been accumulated.

Incremental Memory Update. Continuously encoding all incoming video frames results in linearly growing visual tokens, which is prohibitive for real-time decision making. To address this, we

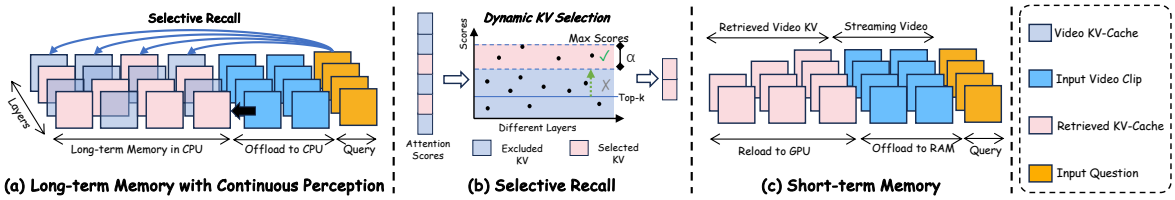

Figure 3: **Overview of streaming KV-caches. (a) Long-term Memory:** As the video stream is continuously encoded, key-value (KV) caches from earlier frames are offloaded to CPU memory as long-term memory. **(b) Selective Recall:** Upon a query, relevant KV-caches are dynamically selected across layers based on attention scores. **(c) Short-term Memory:** Selected KV-caches are reloaded to GPU and combined with streaming inputs as short-term memory for efficient response generation.

compress the visual token sequence $\mathcal{X}_t = \{x_i\}_{i=1}^{N_t}$ of each incoming clip v_t into a compact textual caption $\mathcal{C}_t = \{c_i\}_{i=1}^{n_t}$ ($N_t \gg n_t$), and maintain a memory state m_t that is recurrently updated as:

$$m_t = \mathcal{U}(m_{t-1}, \mathcal{C}_t), \tag{1}$$

where $\mathcal{U}(\cdot)$ denotes the update function parameterized by the captioning model. This recurrence ensures that only the previous state m_{t-1} and the current caption $\mathcal{C}_t$ are needed at each step, bounding memory cost at a constant level regardless of stream length. We note that this compression inevitably loses early fine-grained details; this limitation is compensated by the streaming KV-cache (Sec. 3.2), which preserves original visual tokens for selective recall during response generation.

Proactive Anticipation with Plan Evaluation. Within the StreamAgent framework, we introduce the anticipatory stream agent, which leverages a lightweight MLLM, denoted as $A(\cdot)$, to forecast the temporal dynamics and spatial distribution of task-relevant events. The current state is represented as $S_t = A(m_t, v_t)$, integrating the memory m_t (text) and the current observation v_t (video). From S_t, the agent reasons about potential future event trajectories $E_t^{(h)} = \{e_i\}_{i=t+1}^{t+h}$, where each e_i denotes a predicted event at future timestamp i and h conceptually represents the temporal reasoning depth. To elicit diverse candidate plans that cover different levels of temporal foresight, StreamAgent prompts $A(\cdot)$ from three complementary perspectives. The *Reactive* perspective (h=0) grounds decisions solely in the current state S_t without forecasting, favoring immediate response when evidence appears sufficient. The *Proactive* perspective (h=δ) extrapolates from current observations to anticipate near-future task-relevant events. The *Speculative* perspective (h=Δ, $\Delta \gg \delta$) explores long-range future possibilities for queries requiring distant evidence. Each perspective produces one or more candidate plans, forming:

$$P_j = A\Big(E_t^{(h_j)},\, S_t \mid m_t,\, v_t\Big), \quad \mathcal{P} = \{P_j\}_{j=1}^{k}, \tag{2}$$

where $h_j \in \{0, \delta, \Delta\}$ and k is the total number of candidate plans across all perspectives.

After generating the set of candidate plans, the same agent $A(\cdot)$ evaluates each plan using a separate scoring prompt, following the LLM-as-a-judge paradigm (Zheng et al., 2023). The evaluation is guided by the complementary criteria of assessing whether the currently accumulated evidence is sufficient to answer the query Q and whether further observation of future video frames is likely to provide critical missing information.Based on these considerations, the agent produces a holistic assessment of each plan and selects the one deemed most appropriate for the current decision.

Tool-Augmented Action. Beyond passive frame consumption, StreamAgent acts as a goal-driven information explorer that proactively determines *when*, *where*, and *how* to acquire critical visual evidence. Given the selected plan $\hat{P}$, the current state S_t, and the query Q, the anticipatory agent $A(\cdot)$ outputs a structured action specification indicating which tool to invoke and the corresponding spatial-temporal parameters (*e.g.,* a bounding box for region-of-interest cropping or a target region for continuous tracking). Formally, at each timestamp t, the agent selects a subset of tools $\mathcal{T}_t' \subseteq \mathcal{T}$ and applies each $\phi_j \in \mathcal{T}_t'$ to a designated sub-region $v_{t+1}^j \subseteq v_{t+1}$ of the incoming frame, producing intermediate results $R_j = \phi_j(v_{t+1}^j)$. The perception state is then updated as:

$$S_{t+1} = A\Big(m_t,\, \{R_j\}_{j=1}^{|\mathcal{T}_t'|}\Big). \tag{3}$$

Crucially, tool invocations are not one-shot: the plan $\hat{P}$ may instruct the agent to *persistently monitor* specific spatial regions or track designated objects across future frames $v_{t+1}, v_{t+2}, \ldots$ until the target evidence is captured. By iteratively refining

tool usage and perception targets along the predicted planning trajectory $\hat{P}$, StreamAgent exhibits proactive information hunting behavior, dynamically adapting its sensory focus to accumulate task-relevant evidence in streaming environments.

3.2 Streaming Video KV-caches Memory Mechanism

This section presents the two processes, *Continuous Perception* and *Selective Recall*, that constitute the streaming KV-caches memory mechanism, as illustrated in Fig. 3.

Continuous Perception. Continuous Perception adopts an incremental encoding and prefill strategy, storing the resulting KV caches from each video clip into long-term memory for future retrieval (*i.e.,* Selective Recall). As shown in Fig. 3(a), each video clip is sequentially encoded and prefilled, producing a KV-cache: $\{H^k_{\text{clip}}, H^v_{\text{clip}}\} = \{(\mathbf{k}_j, \mathbf{v}_j)\}_{j=1}^{l_P}$, where l_P denotes the total KV length from prior clips. To minimize peak activation memory usage, we apply chunked prefill within each clip. Given an incoming video clip v_i with token sequence $\mathbf{Z}^{v_i} = \{z_j^{v_i}\}_{j=1}^{n}$, we divide it into C chunks $\mathbf{C}^{v_i} = \{\mathbf{Z}_j^{v_i}\}_{j=1}^{\mu}$, where μ and n denote the total number of chunks and tokens in v_i, respectively, and each token $z_j^{v_i} \in \mathbb{R}^h$ has hidden size h. At each transformer layer, we sequentially prefill the current visual tokens chunk $X = \{z_{j+l_C}\}_{j=1}^{l_X}$ while maintaining the corresponding KV-cache of the past chunks as $\{H^k_{\text{chunk}}, H^v_{\text{chunk}}\} = \{(\mathbf{k}_j, \mathbf{v}_j)\}_{j=1}^{l_C}$, where l_X represents the chunk size, and l_C denotes the length of the previously accumulated KV cache in the current clip. The attention keys and values are then built as:

$$K = [H^k_{\text{clip}}, H^k_{\text{chunk}}, XW_K], V = [H^v_{\text{clip}}, H^v_{\text{chunk}}, XW_V], \tag{4}$$

where W_K, and W_V are the K and V weight matrices, stored for future retrieval, respectively.

Selective Recall. Selective Recall dynamically retrieves question-relevant KV-caches from long-term memory into GPU-resident short-term memory, adapting to the attention behavior of each transformer layer for efficient and accurate inference.

• *Retrieving with Dynamic Attention Pattern.* Recent advances in long-sequence modeling with LLMs show that the number of influential keys and values varies across different layers (Lee et al., 2024b). Therefore, unlike ReKV (Di et al., 2025), which retains a fixed number of KV entries per layer, we propose a layer-adaptive retrieval strategy that dynamically selects variable numbers of KV entries based on attention patterns (see Fig. 3(b)). This allows layers with broader attention to access more entries, while layers with narrower focus retrieve fewer. In detail, during video stream prefill, we first compute a representative feature per frame to enable rapid scoring: $\frac{1}{T_f}\sum_{j=1}^{T_f} \mathbf{k}_j \in \mathbb{R}^{D'}$, where T_f is the number of tokens per frame, and $\mathbf{k}_j$ is the j-th key vector. Subsequently, to match frame relevance to the question, we compute the query-level attention descriptor by averaging over all question tokens: $\frac{1}{T_q}\sum_{j=1}^{T_q} \mathbf{q}_j \in \mathbb{R}^{D'}$, where T_q is the number of query tokens, and $\mathbf{q}_j$ is the score vector of the j-th token across each attention head. Since softmax exponentially amplifies higher attention scores, we filter out less relevant frames by retaining only those with scores within a margin α of the maximum, *i.e.,* score $\geq \max - \alpha$. Thus, the set of important keys can be formulated as:

$$\mathcal{J}_h = \left\{ j \in \{1, \ldots, L_k\} \mid \max_{j'}(\mathbf{S}_h) - \mathbf{S}_{h,j} \leq \alpha \right\}, \tag{5}$$

where $\mathbf{S}_h \in \mathbb{R}^{L_k}$ and L_k denote the attention scores for head h and total number of stored keys, respectively. For example, consider l frames, with n frames having attention scores above the threshold α. The softmax-normalized importance of the i-th frame is given by:

$$\beta_i = \frac{e^{\mathbf{S}_{h,i}}}{\sum_{j=1}^{l} e^{\mathbf{S}_{h,j}}}, \tag{6}$$

where $\mathbf{S}_{h,i}$ denote the attention scores of i-th frame. If $\alpha = 6$ and the score of the i-th frame falls below $\max(\mathbf{S}_h) - 6$, its softmax weight β_i is at most $1/e^6 \approx 1/403.4$ of the maximum. Such negligible weights justify its exclusion, contributing minimally to the final attention distribution.

• *Answering with Retrieved KV.* The retrieved KV-caches serve as the contextual input for video understanding. As illustrated in Fig. 3(c), the KV entries corresponding to the current question are concatenated with the KV-caches retrieved from the long-term memory. In this case, the K and V used for attention computation are represented as:

$$K = [HX^k_{\text{out}}, X_q \cdot W_K], \quad V = [H^v_{\text{out}}, X_q \cdot W_V], \tag{7}$$

where H^k_{out} and H^v_{out} represent the key and value cache outputs which are retrieved from the long-term memory, respectively, and X_q denotes either

Model	#Frames	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 Agents	-	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 Multimodal Models																	
Gemini 1.5 Pro (Team et al., 2023)	1fps	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
GPT-4o (OpenAI, 2024)	64	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
Open-source Offline VideoLLMs																	
LLaVA-NeXT-Video-7B (?)	64	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
LLaVA-OneVision-7B(Li et al., 2024a)	64	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
Qwen2-VL-7B (Wang et al., 2024a)	64	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
InternVL-V2-8B (Chen et al., 2024b)	64	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
LongVU-7B (Shen et al., 2024)	1fps	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
Open-source Online VideoLLMs																	
Flash-VStream-7B (Zhang et al., 2024a)	1fps	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
VideoLLM-online-8B (Chen et al., 2024a)	2fps	8.1	23.9	12.1	14.0	45.5	21.2	20.8	22.2	18.8	12.2	17.7	-	-	-	-	-
Dispider (Qian et al., 2025)	1fps	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-7B (Yao et al., 2025)	1fps	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-7B (Ours)	1fps	**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**

Table 1: Evaluation results on OVO-Bench (Li et al., 2025b) comprising three categories: i) *Real-Time Visual Perception* (OCR: Optical Character Recognition, ACR: Action Recognition, ATR: Attribute Recognition, STU: Spatial Understanding, FPD: Future Prediction, OJR: Object Recognition), ii) *Backward Tracing* (EPM: Episodic Memory, ASI: Action Sequence Identification, HLD: Hallucination Detection), and iii) *Forward Active Responding* (REC: Repetition Event Count, SSR: Sequential Steps Recognition, CRR: Clues Reveal Responding).

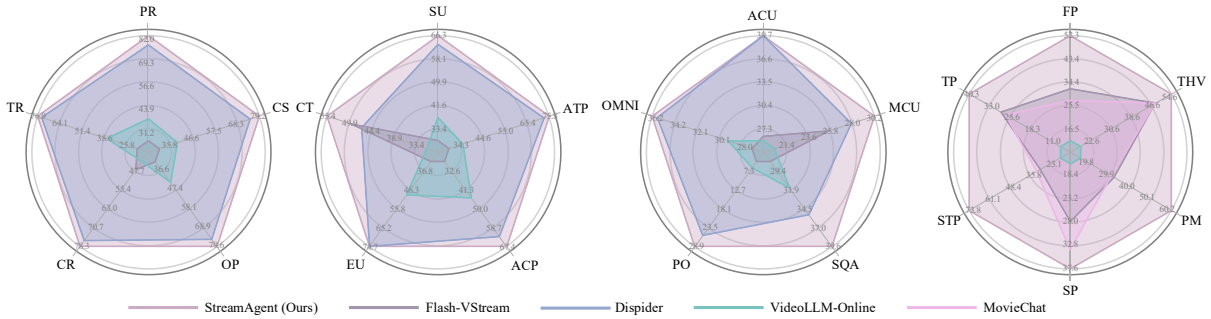

Figure 4: Comparative radar plots of StreamAgent and existing online video LLMs on diverse benchmarks, StreamingBench (first three) and OV-Bench (last), demonstrating StreamAgent's superior unified video representation capabilities.

the current input question or the next token that will be generated.

4 Experiments

4.1 Implementation Details

We employ the lightweight Qwen2.5VL-3B (Bai et al., 2025) as the agent's core for planning and tool coordination, and the larger Qwen2.5VL-7B (Bai et al., 2025) for precise interactions. Input video frames are resized to 224×224 at 1 FPS, following Dispider (Qian et al., 2025). For long-term memory, we first store the KV-cache in GPU memory, and offload to CPU memory once a predefined threshold is reached. The default value of α is set to 3. All experiments are conducted on NVIDIA A800 (80GB) GPUs with FP16 precision. Further setup details are provided in the Appendix C.

4.2 Results on Streaming Video Benchmarks

We evaluate StreamAgent on three streaming video benchmarks tailored to different aspects of streaming video understanding, including StreamingBench (Lin et al., 2024), OVO-Bench (Li et al., 2025b) and OVbench (Huang et al., 2025b), under strict real-time settings. Table 1 reports StreamAgent's performance on OVO-Bench which consists of 12 diverse tasks grouped into three core capabilities: Backward Tracing, Real-Time Visual Perception, and Forward Active Responding. StreamAgent achieves the highest overall performance among online models and even outperforms most offline. Notably, it improves Forward Active Responding by 10.7% over Dispider (Qian et al., 2025), and closely matches the top offline model, highlighting the effectiveness of proactive planning and streaming-aware design. To further assess generalization, we evaluate on StreamingBench and OVBench. While StreamingBench focuses on real-time perception with fine-grained, time-stamped queries, OVBench emphasizes long-range temporal reasoning across 16 subtasks spanning past, present, and future. As illustrated in Fig. 4, StreamAgent achieves state-of-the-art results on both benchmarks, significantly outperforming previous

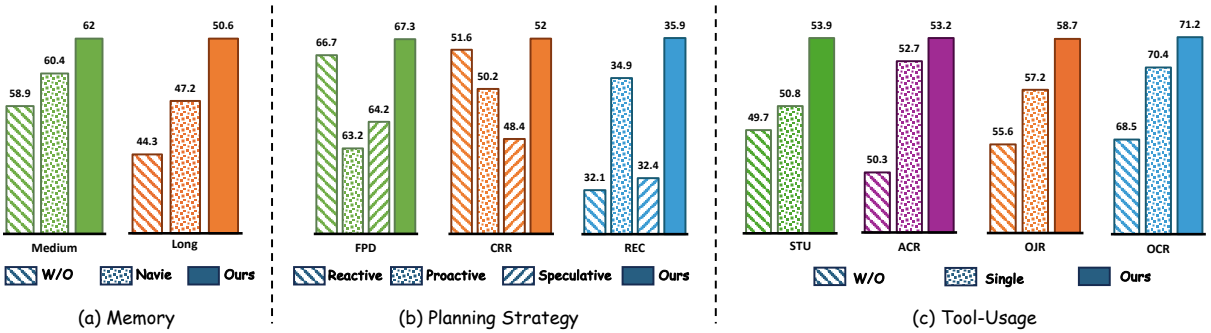

Figure 5: **Ablation Study** on three core components: (a) The effectiveness of the memory module is evaluated on the *medium* and *long* subsets of the VideoMME benchmark. (b) The impact of different planning strategies is evaluated on an OVOBench (Li et al., 2025b) subset designed to test temporal foresight. (c) Tool usage is assessed on a separate subset to examine how proactive interaction enhances performance.

online models and approaching offline baselines. These gains are driven by our proactive StreamAgent, which accurately predicts task-relevant temporal and spatial cues, and our efficient streaming KV-cache mechanism, which enhances long-form understanding by selective recall while reducing computational overhead.

4.3 Ablation Study

Effectiveness of Agent Design. To evaluate the contribution of each core component in our StreamAgent framework, we conduct a comprehensive ablation study along three key axes: *incremental memory, planning strategy, and tool-based active perception*. Each experiment is designed to isolate the effect of the corresponding module on the final performance.

1. Memory. Specifically, we compare the performance of our memory mechanism against approaches without memory and those using traditional segmentation-based captioning, where captions are generated for each segment and then concatenated. Our method shows significant gains on longer videos (Fig. 5 (a)), which confirms that Markov memory not only reduces redundancy but also enhances long-term context modeling crucial for high-quality video question answering.

2. Planning. For planning, we compare *Reactive, Proactive, and Speculative*. strategies with our heuristic-based planning. As shown in Fig. 5 (b), our approach consistently outperforms fixed strategies across all time scales. Notably, the performance of speculative planning remains relatively limited across videos of all lengths, likely due to the inherent uncertainty of long-horizon predictions. Additionally, the nature of the dataset may contribute to this trend, as many questions tend to favor short-term reasoning and do not require extensive temporal foresight.

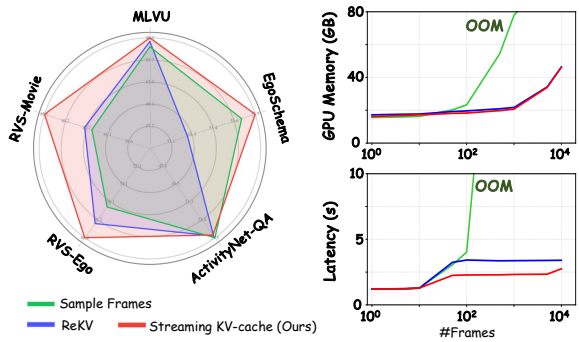

Figure 6: Streaming KV-cache boosts Streaming VideoQA accuracy and achieves over 30% faster inference than ReKV when evaluated with Qwen2.5vl-7B on an A800 (80GB).

Model	#Frames	MLVU	VideoMME			
			Short	Medium	Long	Overall
VideoAgent	87	57.8	63.6	55.4	49.0	56.0
VideoMemAgent	72	58.2	55.3	**64.2**	**52.7**	57.4
ReAgent-V	35	60.7	**73.5**	58.2	49.8	60.7
StreamAgent-7B	1fps	**67.2**	73.4	62.0	50.6	**62.9**

Table 2: Ablation of different video understanding agents.

3. Tool-Use. Finally, we ablate external tools to evaluate the impact of proactive information hunting. As shown in Fig. 5 (c), tool-based perception significantly improves accuracy with minimal overhead, underscoring the value of targeted, strategic perception beyond passive observation.

Effectiveness of Streaming KV-cache. To evaluate the effectiveness of our streaming KV-cache mechanism, we follow the experimental protocol of ReKV (Di et al., 2025) and test on five diverse benchmarks covering different kinds of traditional scenarios: *MLVU* (Zhou et al., 2024), *EgoSchema* (Mangalam et al., 2023), *ActivityNet-QA* (Yu et al., 2019), *RVS-Ego* and *RVS-Movie* (Zhang et al., 2024a). We compare against representative baselines: Sampled Frames, and ReKV. As shown in Fig. 6, our hybrid architec-

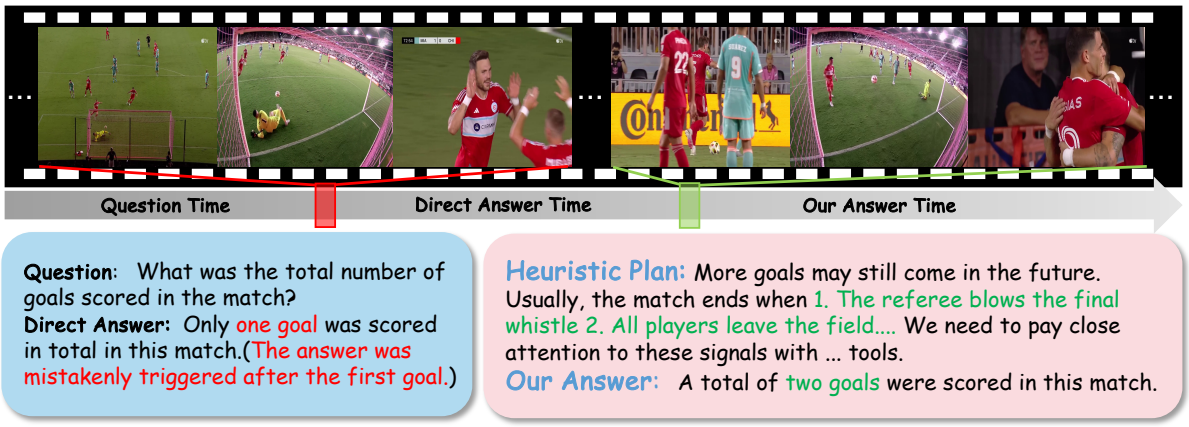

Figure 7: **Proactive Anticipation examples.** The video stream is processed frame by frame. On the left, a Direct Answer is triggered prematurely right after the first goal, resulting in an incorrect response due to the lack of future planning. On the right, the proposed Heuristic Plan waits for clear signs that the game has ended, such as the referee's final whistle, before answering.

Model	#Frames	MLVU	LongVideoBench	VideoMME	
				overall	long
Video Length	-	3∼120 min	8 sec∼60 min	1∼60 min	30∼60 min
Open-Source Offline VideoLLMs					
LLaMA-VID-7B	1fps	33.2	-	-	-
MovieChat-7B	2048	25.8	-	38.2	33.4
LLaVA-Next-Video-7B	32	-	43.5	46.6	-
VideoChat2-7B	16	47.9	39.3	39.5	33.2
LongVA-7B	128	56.3	-	52.6	46.2
Qwen2.5-VL-7B	1fps	66.9	61.5	63.2	50.4
Open-source Online VideoLLMs					
Dispider-7B	1fps	61.7	-	57.2	-
VideoChat-Online-8B	2fps	-	-	52.8	44.9
TimeChat-Online-7B	1fps	65.4	57.7	62.5	49.2
StreamAgent-7B	1fps	**67.2**	**57.9**	**62.9**	**50.6**

Table 3: Results on offline long video benchmarks. We report the accuracy on the MLVU (Zhou et al., 2024), LongVideobench (Wu et al., 2024b) and VideoMME (Fu et al., 2024) without subtitles.

ture enables efficient communication, achieving up to 30% lower latency than ReKV as frame counts increase. Moreover, GPU memory usage remains on par with Sampled Frames, while our method consistently outperforms baselines. More details can be found in Appendix C.

Ablation Study on Video Understanding Agent. In the offline setting, we further compare the performance of various video understanding agents, VideoAgent (Wang et al., 2024b), VideoMemAgent (Fan et al., 2024) and ReAgent-V (Zhou et al., 2025a). Benefiting from our memory mechanism and streaming KV-cache, our method processes videos at 1 FPS and shows outperformance (Table 2) on video benchmarks.

Results on Offline Long Video Tasks. In addition to the streaming setting, we also evaluate StreamAgent on three offline long-form video understanding benchmarks: VideoMME (Fu et al., 2024), MLVU (Zhou et al., 2024), and LongVideoBench (Wu et al., 2024b). In the offline setting, the entire video is provided as input to the VideoLLMs. Table 3 demonstrates that StreamAgent exhibits superior offline video understanding capabilities compared to recent state-of-the-art VideoLLMs, including LLaMA-VID (Li et al., 2024b), MovieChat (Song et al., 2024), LLava-next-video (Zhang et al., 2024f), VideoChat (Li et al., 2023), LongVA (Zhang et al., 2024e), Qwen2.5-VL (Wang et al., 2024a), Dispider (Qian et al., 2025), VideoChat-Online (Huang et al., 2025b). Thanks to incremental memory module and KV cache retrieval mechanism, StreamAgent handles conventional scenario well and achieves competitive performance with offline models.

Case Study. To evaluate StreamAgent's capability in *Proactive Event Anticipation*, we present a case from StreamingBench (Lin et al., 2024) dataset involving the question "What was the total number of goals scored in the match?", a scene from a football match, as shown in Fig. 7. When the model lacked a plan, it prematurely triggered a response after the first goal and incorrectly answered "1 goal.". Through heuristic planning, we avoid such premature answers by using tools to focus on key signals and responding after those signals are observed. This case highlights the value of future-aware planning and tool integration in proactive video understanding.

5 Conclusion

In this paper, we propose StreamAgent for streaming video understanding, which proactively anticipates the temporal intervals and spatial regions of future task-relevant information, enabling proactive and accurate responses. Furthermore, we introduce streaming KV-cache that constructs a hierarchical memory structure and supports selective recall of relevant tokens for efficient semantic retrieval. Experiments demonstrate that our proposed method

achieves state-of-the-art performance on streaming and long video understanding benchmarks in both response accuracy and real-time efficiency.

Limitations

Although our proposed framework and dataset significantly enhance the streaming capabilities of existing offline video large language models (Video-LLMs), there are still some noteworthy limitations.

First, although the decision model is compact and decoupled from the main language model, its decision quality is still constrained by the capacity of the small activation model. While tool usage helps mitigate this issue, the model may still struggle to accurately predict the optimal response timing in highly complex or ambiguous scenarios.

Second, while StreamAgent improves response accuracy through collaborative interactions among multiple agents and proactive acquisition of future information, the lack of large-scale streaming data for training remains a challenge that limits its full potential.

Third, although our streaming KV-cache effectively handles long-term memory, we aim to further reduce computational overhead and improve efficiency by removing redundant information before it enters the MLLM, through various redundancy elimination techniques.

References

Huda Alamri, Vincent Cartillier, Abhishek Das, Jue Wang, Anoop Cherian, Irfan Essa, Dhruv Batra, Tim K Marks, Chiori Hori, Peter Anderson, and 1 others. 2019. Audio visual scene-aware dialog. In *CVPR*, pages 7558–7567.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.

Dibyadip Chatterjee, Edoardo Remelli, Yale Song, Bugra Tekin, Abhay Mittal, Bharat Bhatnagar, Necati Cihan Camgöz, Shreyas Hampali, Eric Sauser, Shugao Ma, Angela Yao, and Fadime Sener. 2025. Memory-efficient streaming videollms for real-time procedural video understanding. *Preprint*, arXiv:2504.13915.

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

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 *CVPR*.

Joya Chen, Ziyun Zeng, Yiqi Lin, Wei Li, Zejun Ma, and Mike Zheng Shou. 2025b. Livecc: Learning video llm with streaming speech transcription at scale. *Preprint*, arXiv:2504.16030.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, pages 24185–24198.

Joonmyung Choi, Sanghyeok Lee, Jaewon Chu, Minhyuk Choi, and Hyunwoo J. Kim. 2024. vid-tldr: Training free token merging for light-weight video transformer. *Preprint*, arXiv:2403.13347.

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*.

Xin Ding, Hao Wu, Yifan Yang, Shiqi Jiang, Donglin Bai, Zhibo Chen, and Ting Cao. 2025. Streammind: Unlocking full frame rate streaming video dialogue through event-gated cognition. *Preprint*, arXiv:2503.06220.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. 2024. Videoagent: A memory-augmented multimodal agent for video understanding. *Preprint*, arXiv:2403.11481.

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. *Preprint*, arXiv:2501.03230.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. 2025. Video-r1: Reinforcing video reasoning in mllms. *Preprint*, arXiv:2503.21776.

Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2024. Video-mme: The first-ever comprehensive evaluation

benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*.

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

Dong Guo, Faming Wu, Zhuofan Zheng, Zihao Wang, Zilong Huang, Ziyu Zhu, and Zuquan Song. 2025. Seed1.5-vl technical report. *Preprint*, arXiv:2505.07062.

Kai Hu, Feng Gao, Xiaohan Nie, Peng Zhou, Son Tran, Tal Neiman, Lingyun Wang, Mubarak Shah, Raffay Hamid, Bing Yin, and Trishul Chilimbi. 2025. M-llm based video frame selection for efficient video understanding. *Preprint*, arXiv:2502.19680.

De-An Huang, Subhashree Radhakrishnan, Zhiding Yu, and Jan Kautz. 2025a. Frag: Frame selection augmented generation for long video and long document understanding. *Preprint*, arXiv:2504.17447.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. 2020. Movienet: A holistic dataset for movie understanding. In *ECCV*.

Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. 2025b. Online video understanding: Ovbench and videochat-online. *Preprint*, arXiv:2501.00584.

Sullam Jeoung, Goeric Huybrechts, Bhavana Ganesh, Aram Galstyan, and Sravan Bodapati. 2024. Adaptive video understanding agent: Enhancing efficiency with dynamic frame sampling and feedback-driven reasoning. *arXiv preprint arXiv:2410.20252*.

Peng Jin, Ryuichi Takanobu, Caiwan Zhang, Xiaochun Cao, and Li Yuan. 2023. Chat-univi: Unified visual representation empowers large language models with image and video understanding. *arXiv preprint arXiv:2311.08046*.

Bruno Korbar, Yongqin Xian, Alessio Tonioni, Andrew Zisserman, and Federico Tombari. 2024. Text-conditioned resampler for long form video understanding. *Preprint*, arXiv:2312.11897.

Noriyuki Kugo, Xiang Li, Zixin Li, Ashish Gupta, Arpandeep Khatua, Nidhish Jain, Chaitanya Patel, Yuta Kyuragi, Yasunori Ishii, Masamoto Tanabiki, and 1 others. 2025. Videomultiagents: A multi-agent framework for video question answering. *arXiv preprint arXiv:2504.20091*.

Seon-Ho Lee, Jue Wang, Zhikang Zhang, David Fan, and Xinyu Li. 2024a. Video token merging for long-form video understanding. *Preprint*, arXiv:2410.23782.

Wonbeom Lee, Jungi Lee, Junghwan Seo, and Jaewoong Sim. 2024b. Infinigen: Efficient generative inference of large language models with dynamic kv cache management. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pages 155–172.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

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

Ruanjun Li, Yuedong Tan, Yuanming Shi, and Jiawei Shao. 2025a. Videoscan: Enabling efficient streaming video understanding via frame-level semantic carriers. *Preprint*, arXiv:2503.09387.

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

Yifei Li, Junbo Niu, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. 2025b. Ovo-bench: How far is your video-llms from real-world online video understanding? *Preprint*, arXiv:2501.05510.

Yunxin Li, Xinyu Chen, Baotian Hu, Longyue Wang, Haoyuan Shi, and Min Zhang. 2024c. Videovista: A versatile benchmark for video understanding and reasoning. *Preprint*, arXiv:2406.11303.

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

Jihao Liu, Zhiding Yu, Shiyi Lan, Shihao Wang, Rongyao Fang, Jan Kautz, Hongsheng Li, and Jose M. Alvare. 2025a. Streamchat: Chatting with streaming video. *Preprint*, arXiv:2412.08646.

Shuming Liu, Chen Zhao, Tianqi Xu, and Bernard Ghanem. 2025b. Bolt: Boost large vision-language model without training for long-form video understanding. *Preprint*, arXiv:2503.21483.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2024. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *ACL*.

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

Zhenyu Ning, Guangda Liu, Qihao Jin, Wenchao Ding, Minyi Guo, and Jieru Zhao. 2025. Livevlm: Efficient online video understanding via streaming-oriented kv cache and retrieval. *arXiv preprint arXiv:2505.15269*.

Zhenyu Ning, Jieru Zhao, Qihao Jin, Wenchao Ding, and Minyi Guo. 2024. Inf-mllm: Efficient streaming inference of multimodal large language models on a single gpu. *Preprint*, arXiv:2409.09086.

OpenAI. 2023a. Gpt-4.

OpenAI. 2023b. Gpt-4 technical report. Technical report.

OpenAI. 2023c. Gpt-4v.

OpenAI. 2024. Gpt-4o.

Sam Pollard and Michael Wray. 2025. Video, how do your tokens merge? *Preprint*, arXiv:2506.03885.

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. *arXiv preprint arXiv:2501.03218*.

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

Benjamin Schneider, Dongfu Jiang, Chao Du, Tianyu Pang, and Wenhu Chen. 2025. Quickvideo: Real-time long video understanding with system algorithm co-design. *arXiv preprint arXiv:2505.16175*.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. 2024. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*.

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

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, and 1 others. 2024. Moviechat: From dense token to sparse memory for long video understanding. In *CVPR*, pages 18221–18232.

Core Team, Zihao Yue, Zhenru Lin, Yifan Song, Weikun Wang, Shuhuai Ren, Shuhao Gu, Shicheng Li, Peidian Li, Liang Zhao, Lei Li, Kainan Bao, Hao Tian, Hailin Zhang, Gang Wang, Dawei Zhu, Cici, Chenhong He, Bowen Ye, and 55 others. 2025a. Mimo-vl technical report. *Preprint*, arXiv:2506.03569.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Kwai Keye Team, Biao Yang, Bin Wen, Changyi Liu, Chenglong Chu, Chengru Song, Chongling Rao, Chuan Yi, Da Li, Dunju Zang, Fan Yang, Guorui Zhou, Hao Peng, Haojie Ding, Jiaming Huang, Jiangxia Cao, Jiankang Chen, Jingyun Hua, Jin Ouyang, and 41 others. 2025b. Kwai keye-vl technical report. *Preprint*, arXiv:2507.01949.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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. *Preprint*, arXiv:2505.05467.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024a. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. 2024b. Videoagent: Long-form video understanding with large language model as agent. *arXiv preprint arXiv:2403.10517*.

Yuxuan Wang, Yueqian Wang, Pengfei Wu, Jianxin Liang, Dongyan Zhao, Yang Liu, and Zilong Zheng. 2024c. Efficient temporal extrapolation of multimodal large language models with temporal grounding bridge. *Preprint*, arXiv:2402.16050.

Yuxuan Wang, Cihang Xie, Yang Liu, and Zilong Zheng. 2024d. Videollamb: Long-context video understanding with recurrent memory bridges. *Preprint*, arXiv:2409.01071.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. 2025b. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. *Preprint*, arXiv:2405.19209.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. 2024. Longvlm: Efficient long video understanding via large language models. *Preprint*, arXiv:2404.03384.

Cheng-Kuang Wu, Zhi Rui Tam, Chieh-Yen Lin, Yun-Nung Chen, and Hung yi Lee. 2024a. Streambench: Towards benchmarking continuous improvement of language agents. *Preprint*, arXiv:2406.08747.

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

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. 2025. Mmsearch-r1: Incentivizing lmms to search. *Preprint*, arXiv:2506.20670.

Shiwei Wu, Joya Chen, Kevin Qinghong Lin, Qimeng Wang, Yan Gao, Qianli Xu, Tong Xu, Yao Hu, Enhong Chen, and Mike Zheng Shou. 2024c. Videollm-mod: Efficient video-language streaming with mixture-of-depths vision computation. *Preprint*, arXiv:2408.16730.

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. *Preprint*, arXiv:2501.13468.

Yibin Yan, Jilan Xu, Shangzhe Di, Yikun Liu, Yudi Shi, Qirui Chen, Zeqian Li, Yifei Huang, and Weidi Xie. 2025. Learning streaming video representation via multitask training. *Preprint*, arXiv:2504.20041.

Haolin Yang, Feilong Tang, Ming Hu, Qingyu Yin, Yulong Li, Yexin Liu, Zelin Peng, Peng Gao, Junjun He, Zongyuan Ge, and Imran Razzak. 2025a. Scaling-noise: Scaling inference-time search for generating infinite videos. *Preprint*, arXiv:2503.16400.

Zhenyu Yang, Yuhang Hu, Zemin Du, Dizhan Xue, Shengsheng Qian, Jiahong Wu, Fan Yang, Weiming Dong, and Changsheng Xu. 2025b. Svbench: A benchmark with temporal multi-turn dialogues for streaming video understanding. *Preprint*, arXiv:2502.10810.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, Lingpeng Kong, Qi Liu, Yuanxing Zhang, and Xu Sun. 2025. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. *Preprint*, arXiv:2504.17343.

Fangxu Yu, Lai Jiang, Haoqiang Kang, Shibo Hao, and Lianhui Qin. 2025. Flow of reasoning: Training llms for divergent reasoning with minimal examples. *Preprint*, arXiv:2406.05673.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. 2019. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *AAAI*.

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

Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding. *Preprint*, arXiv:2306.02858.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. 2024a. Flash-vstream: Memory-based real-time understanding for long video streams. *arXiv preprint arXiv:2406.08085*.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. 2024b. Flash-vstream: Memory-based real-time understanding for long video streams. *arXiv preprint arXiv:2406.08085*.

Lu Zhang, Tiancheng Zhao, Heting Ying, Yibo Ma, and Kyusong Lee. 2024c. Omagent: A multi-modal agent framework for complex video understanding with task divide-and-conquer. *Preprint*, arXiv:2406.16620.

Pan Zhang, Xiaoyi Dong, Yuhang Cao, Yuhang Zang, Rui Qian, Xilin Wei, Lin Chen, Yifei Li, Junbo Niu, Shuangrui Ding, Qipeng Guo, Haodong Duan, Xin Chen, Han Lv, Zheng Nie, Min Zhang, Bin Wang, Wenwei Zhang, Xinyue Zhang, and 10 others. 2024d. Internlm-xcomposer2.5-omnilive: A comprehensive multimodal system for long-term streaming video and audio interactions. *Preprint*, arXiv:2412.09596.

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

Shaojie Zhang, Jiahui Yang, Jianqin Yin, Zhenbo Luo, and Jian Luan. 2025. Q-frame: Query-aware frame selection and multi-resolution adaptation for video-llms. *Preprint*, arXiv:2506.22139.

Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. 2024f. Llava-next: A strong zero-shot video understanding model.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang,

Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Zhuo Zhi, Qiangqiang Wu, Wenbo Li, Yinchuan Li, Kun Shao, Kaiwen Zhou, and 1 others. 2025. Videoagent2: Enhancing the llm-based agent system for long-form video understanding by uncertainty-aware cot. *arXiv preprint arXiv:2504.04471*.

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*.

Yiyang Zhou, Yangfan He, Yaofeng Su, Siwei Han, Joel Jang, Gedas Bertasius, Mohit Bansal, and Huaxiu Yao. 2025a. Reagent-v: A reward-driven multi-agent framework for video understanding. *arXiv preprint arXiv:2506.01300*.

Yiyang Zhou, Linjie Li, Shi Qiu, Zhengyuan Yang, Yuyang Zhao, Siwei Han, Yangfan He, Kangqi Li, Haonian Ji, Zihao Zhao, Haibo Tong, Lijuan Wang, and Huaxiu Yao. 2025b. Glimpse: Do large vision-language models truly think with videos or just glimpse at them? *Preprint*, arXiv:2507.09491.

Appendix Table of Contents

- **A Additional Related Work**
- **B Benchmark and Baseline**
- **C Memory Analysis of Streaming KV cache**
- **D Detail of Evaluation**
- **E Pseudocode**
- **F Future Work**

A Additional Related Work.

Long Video Understanding Inspired by the powerful reasoning capabilities of LLMs (Yu et al., 2025), recent works have explored using LLMs to address complex video-related tasks. Since LLMs primarily process text, various methods (Korbar et al., 2024; Weng et al., 2024; Zhang et al., 2023; Jin et al., 2023; Wang et al., 2024c) have been developed to efficiently train multimodal projectors to connect the visual encoder and LLMs or leverage caption-centric information. With the success of VideoLLM (Team et al., 2025b; Shu et al., 2025; Yan et al., 2025; Team et al., 2025a), attention has shifted to the more complex task of long video understanding (Wang et al., 2025b; Zhou et al., 2025b; Wang et al., 2024d), which is typically facilitated through token compression (Pollard and Wray, 2025; Lee et al., 2024a; Choi et al., 2024) and frame selection (Liu et al., 2025b; Zhang et al., 2025; Hu et al., 2025; Huang et al., 2025a). This approach effectively reduces the token sequence length, allowing for better handling of longer videos. Recent work has enhanced the model's capabilities by using inference (Yang et al., 2025a; Feng et al., 2025; Li et al., 2024c; Fei et al., 2024; Tian et al., 2025; Wu et al., 2025), often searching through long videos to obtain more granular details and achieve more accurate grounding during video understanding.

B Benchmark and Baseline.

MLVU$_{\texttt{dev-mc}}$ (Zhou et al., 2024) is the multiple-choice subset of the MLVU-dev benchmark. It focuses on evaluating the long-form video understanding of MLMs. The question-answer pairs are manually labeled and can be divided into 3 groups: single-detail, multi-detail, and holistic. The evaluation metric is Accuracy.

LongVideoBench (Wu et al., 2024b) is a video question-answering benchmark specifically designed for LMMs. It aims to evaluate LMMs' ability to process long, interleaved video and subtitle inputs up to an hour in length. The benchmark includes: 3,763 varying-length web-collected videos with their subtitles, covering diverse themes. A novel "referring reasoning" video question-answering task, where questions contain referring queries that point to specific video contexts, requiring the model to reason over relevant video details. 6,678 human-annotated multiple-choice questions, categorized into 17 fine-grained categories.

VideoMME (Fu et al., 2024) is the first comprehensive, full-spectrum evaluation benchmark for LMMS specifically designed for video analysis. It addresses the current gap in assessing LMMs' ability to process sequential visual data, moving beyond their traditional focus on static image understanding. Here are its key features: Diverse Video Types: Video-MME covers 6 primary visual domains and 30 subfields, ensuring broad applicability across various scenarios. Varied Temporal Durations: It includes short, medium, and long videos, ranging from 11 seconds to 1 hour, to evaluate robust contextual understanding. Broad Data Modalities: Beyond video frames, the benchmark integrates other modalities like subtitles and audio to assess the all-around capabilities of MLMs. High-Quality Annotations: It uses rigorous manual

labeling by expert annotators to ensure precise and reliable model assessment.

EgoSchema (Mangalam et al., 2023) is a diagnostic benchmark for long VideoQA, featuring over 5000 multiple-choice questions and long temporal certificate length. It challenges AI models with long-term understanding, as current state-of-the-art models achieve significantly lower accuracy compared to human performance.

ActivityNet-QA (Yu et al., 2019) encompasses human-annotated QA pairs on 5,800 videos derived from the ActivityNet (Caba Heilbron et al., 2015) dataset. This benchmark is designed to assess the capabilities of VideoQA models in long-term spatiotemporal reasoning. Our evaluation methodology aligns with that of Video-ChatGPT (Maaz et al., 2024), employing `GPT-3.5-turbo-0613` to judge the accuracy of the open-ended VideoQA responses.

RVS-Ego and **RVS-Movie** (Zhang et al., 2024a) are Streaming VideoQA benchmarks, constructed using 10 long videos from the Ego4D dataset (Grauman et al., 2022) and 22 long videos from the MovieNet dataset (Huang et al., 2020), respectively. These benchmarks feature open-ended questions paired with timestamps, which are initially generated by GPT-4V (OpenAI, 2023c) and GPT-4 (OpenAI, 2023a), and subsequently refined through manual filtering.

OVO-Bench (Li et al., 2025b) is a dataset specifically designed to evaluate Video Large Language Models (Video-LLMs) on their ability to understand online, streaming video content. Unlike traditional offline benchmarks that assess models after they have seen the entire video, OVO-Bench focuses on real-time reasoning, requiring models to answer questions at any given timestamp while the video is playing. OVO-Bench defines a comprehensive suite of tasks to evaluate the temporal reasoning and visual understanding capabilities of Video Large Language Models (Video-LLMs) in online settings. These tasks are grouped into three core categories. The first, Backward Tracing, assesses a model's ability to recall and reason about past events, including retrieving key moments (EPM), identifying the correct sequence of actions (ASI), and detecting hallucinated responses to irrelevant questions (HLD). The second category, Real-Time Visual Perception, focuses on understanding the present visual context through spatial reasoning (STU), object recognition (OJR), attribute identification (ATR), action recognition (ACR), text recognition (OCR), and even predicting what might happen next (FPD). The third category, Forward Active Responding, goes beyond passive perception by requiring models to delay responses until sufficient evidence is available—such as recognizing repeated events (REC), detecting procedural transitions (SSR), and waiting for critical clues before answering (CRR). Together, these tasks provide a rigorous evaluation of how well a model can perceive, remember, and anticipate in dynamic, real-world video environments.

StreamingBench (Lin et al., 2024) introduces the first comprehensive benchmark tailored for evaluating how well multimodal large language models (MLLMs) understand streaming video in realistic scenarios. Comprising 900 videos with 4,500 human-curated QA pairs spread across 18 task types, it simulates continuous video interactions by posing five time-staggered questions per video. The benchmark assesses three key understanding dimensions: real-time visual understanding (recognizing objects, actions, or text in the moment), omni-source understanding (integrating synchronized visual and audio inputs), and contextual understanding (maintaining continuity across interactions, detecting anomalies, filtering misleading cues, and responding proactively based on preceding context).

OVBench (Huang et al., 2025b) are Streaming VideoQA benchmarks, denoting the unique challenges and characteristics of online video understanding compared to traditional offline settings. First, it emphasizes the online temporal perspective, where questions are grounded in specific moments, past, present, or future, enabling much finer time sensitivity than offline approaches. It highlights how time-dependent contexts cause answers to evolve dynamically, meaning that the same question may yield different responses as the stream progresses. Third, it points to the need for real-time spatio-temporal interaction in applications like AR glasses or autonomous driving, where immediate and accurate responses to live environments are crucial.

To rigorously assess the performance of StreamAgent, we compare it against a broad array of baseline models that span different capabilities and design philosophies. Proprietary closed-source models include GPT-4o, known for its advanced multimodal reasoning, and Gemini-1.5-Pro, optimized for long-context multimodal input. Among open-source video-language models, the evaluation

Model	Params	Frames	Real-Time Visual Understanding											Omni-Source Understanding					Contextual Understanding					Overall
			OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	**All**	ER	SCU	SD	MA	**All**	ACU	MCU	SQA	PO	**All**	
Human																								
Human‡	-	-	89.47	92.00	93.60	91.47	95.65	92.52	88.00	88.75	89.74	91.30	91.46	88.00	88.24	93.60	90.27	90.26	88.80	90.40	95.00	100	93.55	91.66
Proprietary MLLMs																								
Gemini 1.5 pro	-	1 fps	79.02	80.47	83.54	79.67	80.00	84.74	77.78	64.23	71.95	48.70	75.69	46.80	39.60	74.90	80.00	60.22	51.41	40.73	54.80	45.10	48.73	67.07
GPT-4o	-	64	77.11	80.47	83.91	76.47	70.19	83.80	66.67	62.19	69.12	49.22	73.28	41.20	37.20	43.60	56.00	44.50	41.20	38.40	32.80	56.86	38.70	60.15
Claude 3.5 Sonnet	-	20	80.49	77.34	82.02	81.73	72.33	75.39	61.11	61.79	69.32	43.09	72.44	31.60	34.00	32.80	48.80	36.80	38.40	34.80	34.40	64.71	37.70	57.68
Open-Source Video MLLMs																								
LLaVA-OneVision	7B	32	80.38	74.22	76.03	80.72	72.67	71.65	67.59	65.45	65.72	45.08	71.12	40.80	37.20	33.60	44.80	38.40	35.60	36.00	27.27	29.55	32.74	56.36
Qwen2-VL	7B	0.2-1 fps	75.20	82.81	73.19	77.45	68.32	71.03	72.22	61.19	61.47	46.11	69.04	41.20	22.00	32.80	43.60	34.90	31.20	26.00	39.60	22.73	31.66	54.14
MiniCPM-V 2.6	8B	32	71.93	71.09	77.92	75.82	64.60	65.73	70.37	56.10	62.32	53.37	67.44	40.80	24.00	34.00	41.20	35.00	34.00	31.60	41.92	22.22	34.97	53.85
LLaVA-NeXT-Video	32B	64	78.20	70.31	73.82	76.80	63.35	69.78	57.41	56.10	64.31	38.86	66.96	37.69	24.80	34.40	42.80	34.90	29.20	30.40	35.35	18.18	30.79	52.77
InternVL-V2	8B	16	68.12	60.94	69.40	77.12	67.70	62.93	59.26	53.25	54.96	56.48	63.72	37.60	26.40	37.20	42.00	35.80	32.00	31.20	32.32	40.91	32.42	51.40
Kangaroo	7B	64	71.12	84.38	70.66	73.20	67.08	61.68	56.48	55.69	62.04	38.86	64.60	37.60	31.20	28.80	39.20	34.20	32.80	26.40	33.84	16.00	30.06	51.10
LongVA	7B	128	70.03	63.28	61.20	70.92	62.73	59.50	61.11	53.66	54.67	34.72	59.96	39.60	32.40	28.00	41.60	35.40	32.80	29.60	30.30	15.91	29.95	48.66
VILA-1.5	8B	14	53.68	49.22	70.98	56.86	53.42	53.89	54.63	48.78	50.14	17.62	52.32	41.60	26.40	28.40	36.00	33.10	26.80	34.00	23.23	17.65	27.35	43.20
Video-CCAM	14B	96	56.40	57.81	65.30	62.75	64.60	51.40	42.59	47.97	49.58	31.61	53.96	33.60	22.00	28.40	34.80	29.70	27.60	24.40	16.67	22.73	22.88	42.53
Video-LLaMA2	7B	32	55.86	55.47	57.41	58.17	52.80	43.61	39.81	42.68	45.61	35.23	49.52	30.40	32.40	30.40	36.00	32.40	24.80	26.80	18.67	0.00	21.93	40.40
Streaming MLLMs																								
Flash-VStream	7B	-	25.89	43.57	24.91	23.87	27.33	13.08	18.52	25.20	23.87	48.70	23.23	25.91	24.90	25.60	28.40	26.00	24.80	25.20	26.80	1.96	24.12	24.04
VideoLLM-online	8B	2 fps	39.07	40.06	34.49	31.05	45.96	32.40	31.48	34.16	42.49	27.89	35.99	31.20	**26.51**	24.10	32.00	28.45	24.19	29.20	30.80	3.92	26.55	32.48
Dispider	7B	1 fps	74.92	75.53	74.10	73.08	74.44	59.92	76.14	62.91	62.16	45.80	67.63	35.46	25.26	38.57	43.34	35.66	39.62	27.65	34.80	25.34	33.61	53.12
StreamAgent	7B	1 fps	**79.63**	**78.31**	**79.28**	**75.87**	**74.74**	**76.92**	**82.94**	**66.31**	**73.69**	**55.40**	**74.28**	**35.86**	26.26	**38.87**	**44.04**	**36.26**	**39.72**	**30.25**	**39.60**	**28.90**	**34.62**	**57.02**

Table 4: Performance comparison on StreamingBench on Omni-source Understanding, Contextual Understanding, and Real-Time Visual Understanding. Omni-source Understanding includes Emotion Recognition (ER), Scene Understanding (SCU), Source Discrimination (SD), and Multimodal Alignment (MA). Contextual Understanding includes Misleading Context Understanding (MCU), Anomaly Context Understanding (ACU), Sequential Question Answering (SQA) and Proactive Output (PO). Real-Time Visual Understanding includes Object Perception (OP), Causal Reasoning (CR), Clips Summarization (CS), Attribute Perception (ATP), Event Understanding (EU), Text-Rich Understanding (TR), Prospective Reasoning (PR), Spatial Understanding (SU), Action Perception (ACP), and Counting (CT). Results are categorized into Human, Proprietary MLLMs, and Open-Source MLLMs for a comprehensive evaluation.

includes:

VideoChat2: A chat-centric video-language model built to support multi-turn, conversational interactions grounded in video content. It emphasizes interactive understanding, with capabilities for dialogue continuity, temporal grounding, and multimodal alignment, making it ideal for assistive agents and education scenarios.

MiniCPM-V: MiniCPM-V is a lightweight, vision-language model developed for efficient multimodal understanding and generation tasks. Designed with a compact architecture, it achieves strong performance on image captioning, visual question answering, and other vision-language benchmarks, while maintaining low computational requirements. This makes MiniCPM-V ideal for deployment in resource-constrained environments such as mobile devices and edge computing scenarios.

VILA-1.5: VILA-1.5 is a state-of-the-art vision-language model that excels in multimodal understanding and generation tasks. It integrates visual and textual information through a unified architecture, enabling high performance on benchmarks such as image captioning, visual question answering, and image-text retrieval. With improved alignment between visual and linguistic modalities, VILA-1.5 offers both accuracy and efficiency, making it suitable for a wide range of real-world applications.

Qwen2 and **Qwen2.5-VL** families: Developed by Alibaba, these models exhibit high performance on both image and video understanding benchmarks through advanced multi-modal alignment. Qwen2.5-VL particularly excels in handling long-context and dense visual-textual reasoning tasks, supported by a unified visual-language architecture.

InternVL-2: A high-performing open-source model that scales both the architecture and training data to improve generalization. It incorporates vision-language alignment techniques, dense region grounding, and optimized pretraining routines to support both short and long video tasks with high efficiency.

Kangaroo: Designed specifically for long-context video input, Kangaroo adopts hierarchical attention and memory-efficient tokenization strategies to process hundreds of frames. It is particularly effective for document-level video understanding, such as meeting summarization or sports analysis.

Long-LLaVA: A variant of LLaVA optimized for efficient multi-frame processing. It integrates temporal coherence constraints and cross-frame attention mechanisms, making it capable of capturing

nuanced motion patterns and temporal dependencies for improved video QA and description.

LLaVA-OneVision: LLaVA-OneVision is an open-source large multimodal model that excels across single-image, multi-image, and video understanding tasks. Built on a SigLIP vision encoder and Qwen-2 language backbone, it uses a unified visual token strategy—including AnyRes-9 for high-res images and frame-level pooling for video—to balance representation across modalities.

MovieChat: MovieChat is an innovative long-video understanding framework that combines vision foundation models and large language models using a dual-memory mechanism inspired by the Atkinson–Shiffrin model. It processes video frames with a sliding-window tokenization approach and maintains a rapidly updated short-term memory buffer along with a compact long-term memory.

LLaVA-NeXT-Video: LLaVA-NeXT-Video is an open-source video-language assistant built by fine-tuning the LLaVA-NeXT image model (based on Qwen-1.5 32B) with high-quality video instruction data (∼830K samples) mixed with image data. It leverages the AnyRes representation—originally designed for high-resolution images—to naturally process videos as sequences of frames, enabling strong zero-shot video understanding. With techniques like length generalization and DPO fine-tuning, the model achieves top open-source performance on benchmarks like Video-MME and NextQA-MC, while remaining efficient in inference and deployment.

ReAgent-V: ReAgent-V is an agent-based video-language framework that enhances large vision-language models with dynamic frame selection, multi-step reasoning, and self-correction. By integrating tool-using agents and a feedback-driven refinement process, it achieves superior performance on video understanding tasks with improved efficiency and accuracy.

VideoAgent: An agent framework for long-form video understanding that mimics human cognition through LLM-guided reasoning, CLIP-based retrieval, and VLM-driven state updates.

VideoMemAgent: An agent framework for structured long-form video understanding that integrates unified temporal and object memory with tool-augmented reasoning, enabling multi-round chain-of-thought inference across complex video content.

C Memory Analysis of Streaming KV cache

In this section, we analyze the memory consumption of our approach, focusing on activation memory and key-value (KV) cache memory.

Activation Memory Analysis. The activation memory of modern LLM architecture mainly comes from two components of each transformer layer: 1) Attention layer and 2) MLP layer. We analyze the potential activation memory usage in formulas in the followings and show that chunked-prefill can approximately reduce the activation memory by C times, where C is the number of chunks.

Attention Layer. For an inference computation, the input tensor of the i-th attention layer (X^i) is first layer-normalized to produce X_a^i, which is then fed into the attention module. The output of the attention module after a residual add and layer normalization forms the X_O^i. The X_O^i can be calculated by:

$$Q^i = X_a^i \cdot W_Q^i, \quad K^i = X_a^i \cdot W_K^i, \quad V^i = X_a^i \cdot W_V^i \tag{8}$$

$$X_O^i = f_{\text{norm}}\left(f_{\text{Soft}}\left(\frac{Q^i {K^i}^T}{\sqrt{h}}\right) \cdot V^i \cdot w_O^i + X^i\right) \tag{9}$$

For simplicity, we ignore memory fragmentation; analyzing the computational data flow of the attention layer, the total activation memory with half precision can be expressed as:

$$\mathcal{M}_{\text{attn}} \approx 2B \cdot S(n_h \cdot d_{\text{head}} + n_{kv} \cdot d_{\text{head}}) \cdot 2 \text{ bytes} \tag{10}$$

The first term accounts for storing the input hidden_states X^i and **Q** tensors (Assume that the model's hidden dimension is equal to $n_h \times d_{head}$), while the second term accounts for the K and V tensors (i.e., KV caches). Assuming $B = 1$, $S \approx 921600$, $d_{\text{model}} = 3584$, $d_{\text{head}} = 128$, $n_h = 28$, $n_{kv} = 4$, we compute:

$$\mathcal{M}_{\text{attn}} = 2 \cdot 921600 \cdot 128 \cdot (28 + 4) \cdot 2 \tag{11}$$

$$= 15{,}099{,}494{,}400 \text{ bytes} \tag{12}$$

$$\approx \boxed{14.1 \text{ GiB}} \tag{13}$$

With chunked-prefill using $C = 4096$ chunk size, we can reduce the sequence length S by $C = 225$

times, reducing $\mathcal{M}_{\text{attn}}$ from 17.6 GB to approximately 0.08 GB. This dramatic reduction enables the processing of extremely long sequences that would otherwise be infeasible.

MLP Layer. The SwiGLU (Swish-Gated Linear Unit) enhances transformer models through improved gating mechanisms and has been adopted as the default MLP architecture in many popular LLMs including InternVL2.5 and Qwen2.5 series. For input tensor X^i, the SwiGLU operation is defined as:

$$X^{i+1} = f_{\text{silu}}\left(X_O^i \cdot W_1\right) * \left(W_2 \cdot X_O^i\right) \cdot W_3 + X_O^i \tag{14}$$

For a batch of sequences, activation memory analysis reveals requirements at each computational step. With batch size B, sequence length S, hidden dimension d_{model}, intermediate dimension d_{ff}, and data type float16, the total activation memory for a single SwiGLU layer is:

$$\mathcal{M}_{\text{act}} = (B \cdot S \cdot (2d_{\text{model}} + 3d_{\text{ff}})) \cdot 2 \text{ bytes} \tag{9}$$

For a one-hour video sampled with 1 FPS (3600 frames in total), parameters can be set $B = 1$, $S \approx 921600$, $d_{\text{model}} = 4096$, and $d_{\text{ff}} = 14336$:

$$\mathcal{M}_{\text{act}} = (921600 \cdot (2 \cdot 4096 + 3 \cdot 14336)) \cdot 2 \tag{15}$$

$$= 94,371,840,000 \text{ bytes} \tag{16}$$

$$\approx \boxed{87.9 \text{ GB}} \tag{17}$$

This substantial memory requirement highlights the computational challenges in deploying SwiGLU-based models for high-resolution inputs with extended sequence lengths. However, if we prefill the tokens chunk by chunk, we can reduce the S by C times, and thus reduce the activation memory $\mathcal{M}_{\text{act}}$ by C times. Assuming each chunk contains 4096 visual tokens , then $C = \frac{921600}{4096} = 225$ and we can reduce $\mathcal{M}_{\text{act}}$ from 87.9 GB to 0.4 GB.

KV Cache Memory Analysis. When using Qwen2.5-VL-7B, with $|V| = 921600$ visual tokens, $|Q| = 256$ text tokens, $L = 28$ layers, $n_{kv} = 4$ heads, and $d_h = 128$, the total memory required to store the KV cache in half precision is:

$$\begin{aligned}\text{Memory} &= 2 \times L \times (|V| + |Q|) \times n_{\text{kv}} \times d_h \times 2 \\ &= 52,862,910,464 \text{ bytes} \\ &\approx \boxed{49.2 \text{ GB}}.\end{aligned} \tag{13}$$

Hyper-parameter	**Value**
Visual Encoder	
Frame Sampling Rate	FPS=1
Input Resolution	224*224
Visual Tokens per Image	32
Patch Size	14x14
Large Language Model	
Number of Layers	28
Hidden Size	3584
Vocabulary Size	152064
Number of Attention Heads	28
Number of KV Heads	4
Streaming KV-cache	
α	3
Maximum number of retrievals	256
GPU-Memory(A100 SXM4)	80GB

Table 5: The implementation settings of the evaluation.

D Detail of Evaluation

In this section, we analyze the memory consumption of our approach, focusing on activation memory and key-value (KV) cache memory.

Implement Setting

We present the detail of the hyperparameters during the inference in Table 6.

Evaluation Result

We present the detail of the evaluation results on StreamingBench in Table 4. StreamAgent achieves state-of-the-art performance among open-source models with an overall score of 57.02, outperforming the recent online model Dispider-7B by 3.90 points (57.02 vs. 53.12). While proprietary model Gemini 1.5 pro leads with 67.07.

Ablation Setting

In the experiment of the ablation in streaming KV-cache, we use FPS=0.5 following the setting in ReKV. Notably, we evaluate the streaming KV-cache in offline setting aglining with ReKV, the detail of the hyperparameters are presented in Table 5.

Hyper-parameter	Value
Visual Encoder	
Frame Sampling Rate	FPS=0.5
Input Resolution	448*448
Visual Tokens per Image	128
Patch Size	14x14
Large Language Model	
Number of Layers	28
Hidden Size	3584
Vocabulary Size	152064
Number of Attention Heads	28
Number of KV Heads	4
Streaming KV-cache	
α	3
Maximum number of retrievals	256
GPU-Memory(A100 SXM4)	80GB
ReKV	
Top-k	64
Block Size	1
GPU-Memory(A100 SXM4)	80GB

Table 6: The implementation settings of the streaming KV-cache ablation study.

E Pseudocode

E.1 Algorithm of the Streaming KV-cache.

We present the algorithm of the streaming KV-cache in Algorithm 1

Algorithm 1 Streaming KV-cache Algorithm

Require: Incoming video stream $\{v_1, v_2, \ldots, v_T\}$, query Q_t, threshold margin α

Ensure: Output answer with retrieved memory-aware KV-cache

Initialize long-term memory in CPU $\mathcal{M}_{\text{long}} \leftarrow \emptyset$
for each video clip v_i **do**
  Divide v_i into chunks $\mathbf{C}^{v_i} = \{\mathbf{Z}_j^{v_i}\}_{j=1}^{\mu}$
  Initialize short-term KV-cache $H_{\text{chunk}}^{k,v} \leftarrow \emptyset$
  for each chunk X in $\mathbf{C}^{v_i}$ **do**
    Compute current K, V using Eq. (8)
    Prefill current chunk X
    Append (K, V) to $H_{\text{chunk}}^{k,v}$
  end for
  Store $H_{\text{chunk}}^{k,v}$ into long-term memory: $\mathcal{M}_{\text{long}} \leftarrow \mathcal{M}_{\text{long}} \cup H_{\text{chunk}}^{k,v}$
end for

Selective Recall
Compute average query attention descriptor: $\mathbf{q}_{\text{avg}} = \frac{1}{T_q} \sum_{j=1}^{T_q} \mathbf{q}_j$
Initialize retrieved KV-cache $H_{\text{out}}^{k,v} \leftarrow \emptyset$
for each layer l **do**
  Compute frame-wise attention scores $\mathbf{S}_h$ between $\mathbf{q}_{\text{avg}}$ and stored $\mathbf{k}_j$
  Select top frames: $\mathcal{J}_h = \{j \mid \max(\mathbf{S}_h) - \mathbf{S}_{h,j} \leq \alpha\}$
  Append corresponding k_j, v_j to $H_{\text{out}}^{k,v}$
end for
Compute final K, V with $H_{\text{out}}^{k,v}$ and X_q
return Answer generated via attention over (K, V)

F Future Work

In the future, we aim to use this agent-based workflow to construct data, and then fine-tune models using the constructed data to enhance their tool-using capabilities and their ability to plan for future events. We also intend to apply this approach in real-world scenarios. This planning-then-tool-invocation framework can naturally transfer to practical applications, for example, in intelligent surveillance systems, cameras can proactively adjust their monitoring areas or focal lengths based on the agent's planning to focus on specific regions. Additionally, we hope to leverage more tools specifically designed for video understanding. For instance, when counting repetitions, MLLMs often tend to overcount or undercount. We can use optical flow or other techniques to assist large models in achieving more accurate counting.