Video Streaming Thinking: VideoLLMs Can Watch and Think Simultaneously

Yiran Guan[1*], Liang Yin[1*], Dingkang Liang[1], Jianzhong Ju[2], Zhenbo Luo[2], Jian Luan[2], Yuliang Liu[1], and Xiang Bai[1†]

[1] Huazhong University of Science and Technology, Wuhan, China
[2] MiLM Plus, Xiaomi Inc., Beijing, China
{yiranguan,liangyin,dkliang,xbai}@hust.edu.cn

Abstract. Online Video Large Language Models (VideoLLMs) play a critical role in supporting responsive, real-time interaction. Existing methods focus on streaming perception, lacking a synchronized logical reasoning stream. However, directly applying test-time scaling methods incurs unacceptable response latency. To address this trade-off, we propose ***V**ideo **S**treaming **T**hinking* (VST), a novel paradigm for streaming video understanding. It supports a *thinking-while-watching* mechanism, which activates reasoning over incoming video clips during streaming. This design improves timely comprehension and coherent cognition while preserving real-time responsiveness by amortizing LLM reasoning latency over video playback. Furthermore, we introduce a comprehensive post-training pipeline that integrates VST-SFT, which structurally adapts the offline VideoLLM to causal streaming reasoning, and VST-RL, which provides end-to-end improvement through self-exploration in a multi-turn video interaction environment. Additionally, we devise an automated training-data synthesis pipeline that uses video knowledge graphs to generate high-quality streaming QA pairs, with an entity–relation grounded streaming Chain-of-Thought to enforce multi-evidence reasoning and sustained attention to the video stream. Extensive evaluations show that VST-7B performs strongly on online benchmarks, *e.g.* 79.5% on StreamingBench and 59.3% on OVO-Bench. Meanwhile, VST remains competitive on offline long-form or reasoning benchmarks. Compared with Video-R1, VST responds 15.7× faster and achieves +5.4% improvement on VideoHolmes, demonstrating higher efficiency and strong generalization across diverse video understanding tasks. Code, data, and models have been released at `https://github.com/1ranGuan/VST`.

Keywords: Streaming Video Understanding · CoT · VideoLLM

1 Introduction

Online video understanding enables Video Large Language Models (VideoLLMs) to interpret streaming visual inputs and respond in real time, making it particularly valuable for embodied intelligence [9,13] and interactive AI assistants [3,8].

* Equal contribution. † Corresponding author.

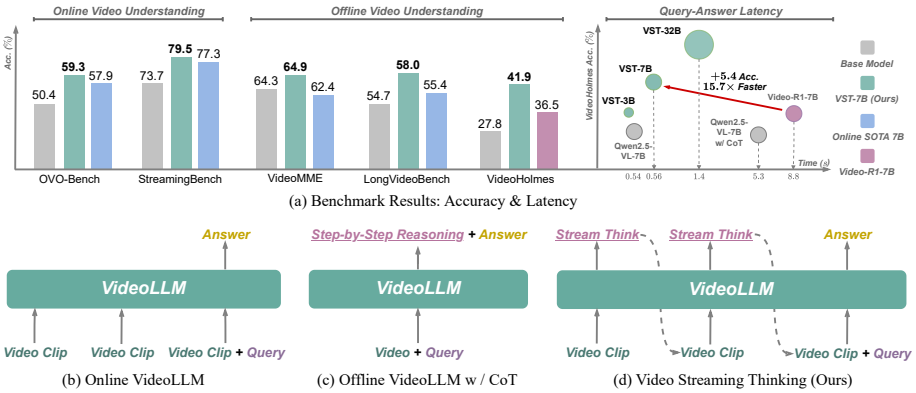


Fig. 1: Benchmark results and paradigm comparison. (a) VST-7B delivers strong performance on online and offline video understanding benchmarks while maintaining low QA latency. (b) Existing streaming VideoLLMs focus on efficient streaming processing, but lack explicit analytical reasoning. (c) VideoLLM with CoT performs heavy post-query step-by-step reasoning to improve performance, but incurs high QA latency. (d) Our method introduces proactive pre-query reasoning, interleaving it with video consumption to achieve both strong performance and efficient responsiveness.

Unlike offline methods that benefit from post-hoc global access to the entire video [1, 22, 24, 28, 54], the core challenges of online video understanding lie in strict temporal causality, real-time processing, and a finite context window.

Several prior methods have been proposed to address the challenges of online video understanding. As shown in Fig. 1(b), they primarily improve context-window efficiency by explicitly managing visual tokens for compression [41, 45, 58, 62] or by retrieving from the KV cache [6, 34, 57]. However, these methods primarily focus on streaming perception and treat the management of visual features as a form of memory, with limited involvement of the LLM itself and no explicit reasoning or analytical deliberation. To fill this missing piece, one promising direction inspired by offline video understanding is to apply test-time scaling via Chain-of-Thought (CoT) to elicit stronger reasoning ability [4, 10, 14, 15, 29, 52, 63, 69], as shown in Fig. 1(c). Nevertheless, directly performing step-by-step reasoning after the user query can significantly increase QA response latency, making it difficult to meet strict real-time requirements in online scenarios.

In this paper, we introduce the ***V**ideo **S**treaming **T**hinking* (VST) to resolve the trade-off between explicit reasoning and real-time responsiveness, shifting the LLM backend from passive waiting to active, intermittent reasoning during video consumption. This design is inspired by insights from human cognition. Findings on *neural coupling* [19,42] suggest that the logical flow in the brain synchronizes closely with the influx of external information, fostering the perception of current signals and their synthesis into a coherent understanding. Similarly, as illustrated in Fig. 1(d), our method continuously processes incoming video clips and produces intermediate thoughts in real time. This eliminates the need to

defer heavy computation until the query arrives, which is a common limitation of offline VideoLLMs with CoT [4, 10, 48]. This *thinking-while-watching* mechanism maintains a coherent internal state over the stream, ensuring that the final response is grounded in a deeply processed understanding of the historical context. By front-loading and amortizing the reasoning cost ahead of query arrival, VST preserves the low QA latency required in streaming scenarios.

We instantiate this paradigm with a dedicated post-training pipeline that combines supervised fine-tuning (VST-SFT) and reinforcement learning (VST-RL). Concretely, we cast streaming thinking as a multi-turn conversation, where the model incrementally writes textual thoughts to an external memory while observing incoming video clips under a constrained visual context window. In the VST-SFT stage, we align the model with the desired streaming reasoning protocol by learning from off-policy demonstrations that strictly respect temporal causality, thereby bootstrapping its basic *thinking-while-watching* capability. Building upon this initialization, the VST-RL stage performs end-to-end reinforcement learning with verifiable rewards, encouraging the model to make intermediate reasoning steps that improve downstream question answering under realistic online conditions.

Due to the scarcity of existing data for video streaming thinking, we develop an automated synthesis pipeline to support our training, particularly the VST-SFT stage that requires high-quality reasoning demonstrations. Specifically, we model entities and their temporal relationships within long videos as knowledge graphs. By sampling paths from these graphs to form evidence chains, we prompt an offline VideoLLM to generate complex QA pairs and their corresponding intermediate CoTs. This design enforces multi-hop reasoning across diverse visual evidence while ensuring strict alignment between the generated thoughts and the video context. Ultimately, we synthesize a large-scale dataset comprising 100K high-quality streaming reasoning samples.

We conducted extensive evaluations across multiple online and offline video understanding benchmarks (see Fig. 1(a)). The results show that our method achieves state-of-the-art performance compared to existing online VideoLLMs, while remaining competitive on offline video understanding benchmarks. Notably, VST performs particularly well on long-form videos that require comprehensive plot comprehension and multi-step reasoning. Moreover, compared to Video-R1, our method achieves higher accuracy while significantly reducing QA latency, demonstrating that VST is a viable test-time scaling approach that meets the requirements of streaming scenarios.

In summary, our main contributions are as follows:

- We propose the VST paradigm to interleave active explicit CoT generation with continuous video streams, enabling amortized test-time scaling with real-time responsiveness.
- A knowledge-graph-based data synthesis pipeline and a dedicated post-training recipe (VST-SFT and VST-RL) are introduced to adapt an offline VideoLLM to streaming settings with strong streaming reasoning capabilities.

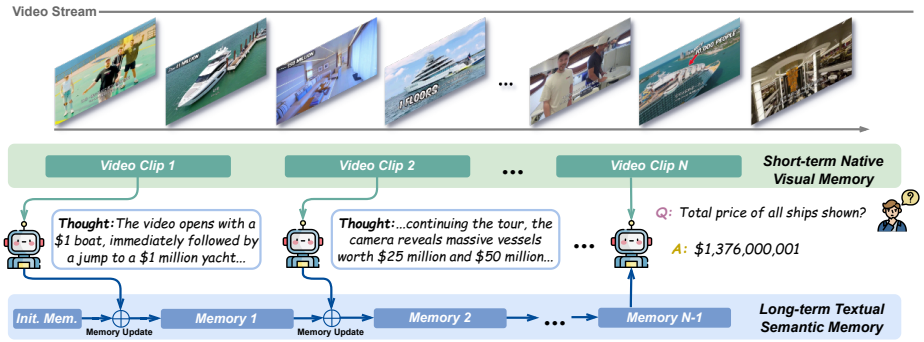


Fig. 2: Illustration of the Video Streaming Thinking pipeline. The model employs a streaming thought mechanism to compress visual dynamics into a long-term textual memory. Combined with the short-term visual buffer, this enables efficient reasoning over indefinite video streams with fixed memory budgets.

- Extensive evaluations across multiple online and offline video understanding benchmarks demonstrate state-of-the-art performance. In addition, compared to offline CoT VideoLLM, our method provides significantly lower QA latency.

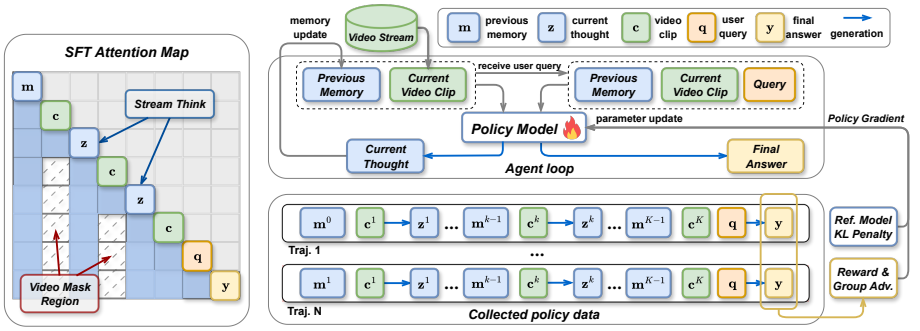
2 Method

2.1 The *Video Streaming Thinking* (VST) Paradigm

We formulate VST as a multi-round video conversation task operating within a constrained context window, as illustrated in Fig. 2. Unlike previous online VideoLLMs, our model leverages streaming intervals before a user query to proactively reason about the content via autoregressive textual generation. This process synthesizes key visual details and event dynamics into a dual-memory system: maintaining a *short-term native video memory* for the current visual context, while accumulating a *long-term textual semantic memory* of past events.

Formally, given a video stream $\mathcal{V}$, let $\mathbf{v}_i$ denote the visual features for the i-th frame. We accumulate these incoming features into discrete clips $\mathbf{c}^k = \{\mathbf{v}_i\}_{i=\tau_{k-1}+1}^{\tau_k}$, where the boundary τ_k is set when the accumulated visual tokens reach the preset capacity L. At each interval k, conditioned on the current clip $\mathbf{c}^k$ and the accumulated memory $\mathbf{m}^{k-1}$, the LLM generates a streaming thought $\mathbf{z}^k$ by sampling from the distribution $\mathbf{z}^k \sim p(\mathbf{z} \mid \mathbf{c}^k, \mathbf{m}^{k-1})$. Here, $\mathbf{z}^k$ summarizes the essential semantics of the current video segment, preserving the continuity of the overall thought process. For the long-term textual memory, we employ a memory update function $\mathbf{m}^k = \text{Update}(\mathbf{m}^{k-1}, \mathbf{z}^k)$, which adopts a simple first-in-first-out strategy to evict the earliest memory entries.

This iterative reasoning process continues until step K, when a user query $\mathbf{q}$ is received. Upon this trigger, the LLM generates the final response $\mathbf{y}$ based on the accumulated thoughts $\{\mathbf{z}^k\}_{k=1}^{K-1}$ and the latest visual context $\mathbf{c}^K$. Consequently,



(a) VST-SFT (b) VST-RL

Fig. 3: Overview of the training pipeline. (a) VST-SFT applies a streaming attention mask to enforce temporal causality, restricting attention to the current visual buffer and history textual context. (b) VST-RL performs on-policy optimization via an agentic loop, improving the quality of streaming thoughts through verifiable rewards computed solely from the final answer.

the joint probability of the thoughts and the final answer is decomposed as:

$$p(\mathbf{y}, \{\mathbf{z}^k\}_{k=1}^{K-1} \mid \mathbf{q}, \mathcal{V}) = \underbrace{p(\mathbf{y} \mid \mathbf{q}, \mathbf{c}^K, \mathbf{m}^{K-1})}_{\text{Direct Answer}} \prod_{k=1}^{K-1} \underbrace{p(\mathbf{z}^k \mid \mathbf{c}^k, \mathbf{m}^{k-1})}_{\text{Streaming Thinking}}. \quad (1)$$

This formulation yields two distinct advantages. 1) It amortizes the computational cost of Chain-of-Thought (CoT) generation over the pre-query phase. This strategy effectively achieves test-time scaling to boost performance without incurring additional latency at the moment of user interaction. 2) The sequential generation of thoughts naturally aligns with the temporal causality inherent in streaming videos. This structure facilitates the adaptation of offline models to online scenarios by mirroring the progressive nature of the video stream.

2.2 Training Method for VST

To instantiate the VST paradigm introduced in Sec. 2.1, we develop a two-stage post-training pipeline that combines supervised fine-tuning (VST-SFT) and reinforcement learning (VST-RL), progressively endowing an offline VideoLLM with streaming thinking capabilities. The VST-SFT stage adapts the offline model to the temporal causality of streaming video, while learning reasoning capabilities from off-policy expert data. Subsequently, VST-RL transitions the model from off-policy imitation to on-policy RL, and refines these learned capabilities for further end-to-end improvement.

Stage 1: VST-SFT. We initiate the training pipeline with SFT to instill the streaming thought mechanism into the offline VideoLLM. For a training instance,

we explicitly formulate the sequence as:

$$\mathcal{S} = \Big(\mathbf{m}^0, (\mathbf{c}^1, \mathbf{z}^1), \ldots, (\mathbf{c}^{K-1}, \mathbf{z}^{K-1}), \mathbf{c}^K, \mathbf{q}, \mathbf{y}\Big). \tag{2}$$

Here, $\mathbf{m}^0$ denotes the initial memory, and $(\mathbf{c}^k, \mathbf{z}^k)$ represent the interleaved video clips and streaming thoughts. The sequence concludes with the final clip $\mathbf{c}^K$, user query $\mathbf{q}$, and ground truth response $\mathbf{y}$.

To align with the streaming inference architecture, we apply a streaming video attention mask. As depicted in Fig. 3(a), this mask restricts the model's attention to a fixed-size window of recent visual tokens, mirroring the short-term visual buffer used during inference. Specifically, let M be the additive attention mask. Let $\mathbb{I}_v(j) \in \{0, 1\}$ indicate whether the j-th token is a visual token, and let L denote the visual buffer size. Therefore, the attention mask can be written as:

$$M_{i,j} = \begin{cases} 0, & j \le i \text{ and } \Big(\mathbb{I}_v(j) = 0 \text{ or } \sum_{t=j+1}^{i} \mathbb{I}_v(t) < L\Big) \\ -\infty, & \text{otherwise} \end{cases} \tag{3}$$

In this way, the model can only access a sliding window of the latest L visual tokens, while all non-visual tokens remain fully visible under the causal constraint. Furthermore, to accommodate context length constraints while handling long-form videos, we implement a temporal segmentation strategy. The original sequence $\mathcal{S}$ is sliced into consecutive segments $\{\mathbf{s}_n\}_{n=1}^{M}$, defined as:

$$\mathbf{s}_n = \begin{cases} \Big(\mathbf{m}^{n-1}, \{(\mathbf{c}^k, \mathbf{z}^k)\}_{k=T_{n-1}+1}^{T_n}\Big), & n < M \\ \Big(\mathbf{m}^{n-1}, \{(\mathbf{c}^k, \mathbf{z}^k)\}_{k=T_{n-1}+1}^{K-1}, \mathbf{c}^K, \mathbf{q}, \mathbf{y}\Big), & n = M \end{cases} \tag{4}$$

where T_n denotes the cut-off index for the n-th segment. The memory state is updated recursively across segments following $\mathbf{m}^n = \text{Update}(\mathbf{m}^{n-1}, \{\mathbf{z}^k\}_{k=T_{n-1}+1}^{T_n})$. During SFT, we apply the standard next-token prediction loss exclusively to the streaming thoughts $\{\mathbf{z}^k\}_{k=1}^{K-1}$ and the final response $\mathbf{y}$, treating visual tokens and historical memory as conditioning inputs.

Stage 2: VST-RL. Building upon the supervised foundation, we introduce VST-RL to transition the model from off-policy imitation to on-policy self-improvement. The RL training process consists of two main phases: trajectory rollout and policy gradient optimization.

As shown in the upper part of Fig. 3(b), the rollout phase operates as an agentic loop. The policy model interacts with the streaming environment to generate a trajectory $\mathcal{T}$ following the predefined joint probability in Eq. (1), where the streaming thoughts $\hat{\mathbf{z}}^k$ and the final response $\hat{\mathbf{y}}$ are sequentially sampled from the sampling policy $\pi_{\theta'}$. After collecting a group of N trajectories $\{\mathcal{T}_i\}_{i=1}^{N}$, we employ a GRPO [15, 33, 60, 61] strategy to optimize the policy model. We compute the reward $\mathbf{r_i}$ solely based on the final answer $\mathbf{y_i}$ via verifiable reward functions. To encourage the model to generate useful streaming thoughts, the

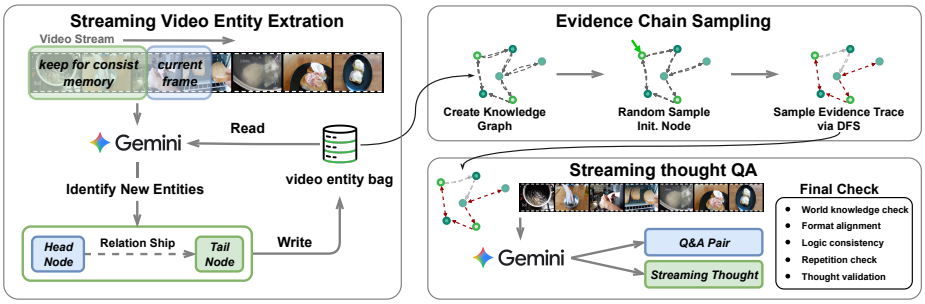


Fig. 4: Stream-Thought QA data curation pipeline. We incrementally extract video entities and relations to build a knowledge graph, sample multi-hop evidence chains, and use Gemini to generate streaming QA pairs with grounded streaming thoughts, followed by automatic filtering.

calculated advantage is assigned to all generated tokens within the entire trajectory $\mathcal{T}_i$. The policy gradient objective is calculated as:

$$\mathcal{J}_{\mathrm{RL}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{\mathcal{T}_i\}_{i=1}^{N}\sim\pi_{\theta'}(\cdot|q)} \left[\frac{1}{\sum_{i=1}^{N}|\mathcal{T}_i|} \sum_{i=1}^{N}\sum_{t=1}^{|\mathcal{T}_i|} \left(\mathcal{L}_{i,t}^{\mathrm{clip}}(\theta) - \beta D_{\mathrm{KL}}(\pi_\theta||\pi_{\mathrm{ref}}) \right) \right], \quad (5)$$

$$\mathcal{L}_{i,t}^{\mathrm{clip}}(\theta) = \min\left[\gamma_t(\theta)\hat{A}_i, \mathrm{clip}\left(\gamma_t(\theta), 1-\epsilon_{\mathrm{low}}, 1+\epsilon_{\mathrm{high}}\right)\hat{A}_i \right]. \quad (6)$$

Where $|\mathcal{T}_i|$ denotes the total number of generated tokens in trajectory $\mathcal{T}_i$, $\gamma_t(\theta)$ represents the probability ratio between π_θ and the sampling policy $\pi_{\theta'}$ at step t, $\hat{A}_i = r_i - \mathrm{mean}(R)$ is the group relative advantage, and $\epsilon_{\mathrm{low}}, \epsilon_{\mathrm{high}}$ are the clipping hyperparameters follow DAPO [61].

2.3 Data Synthesis Pipeline for VST

We generate a set of video streaming thought data to support VST training, motivated by the fact that most existing chain-of-thought (CoT) datasets target offline VideoLLMs with a global, hindsight view of the entire video, making it difficult to avoid information leakage under causal streaming constraints. To this end, we introduce an automated data generation pipeline grounded in knowledge graphs. As illustrated in Fig. 4, the pipeline produces high-quality training examples with explicit reasoning paths through streaming video entity extraction, evidence chain sampling, and streaming thought QA synthesis.

Streaming Video Entity Extraction. To build a temporally consistent knowledge graph, we maintain an *entity bank* and extract triples from a sliding window over the video stream. We segment the video into N scene clips with PySceneDetect. For each incoming clip, an offline VideoLLM (*e.g.* Gemini 3.0 flash) updates the entity bank by adding newly observed entities and relations

as (**head**, relation, **tail**). When the window exceeds size W, we drop the oldest clip and retain the most recent $W-1$ overlapping clips to preserve temporal continuity. The entity bank thus serves as a lightweight memory for consistent entity tracking and timeline-aligned graph construction.

Evidence Chain Sampling. After processing the whole video, the complete entity bank is refined using an LLM to filter out noise entities, such as duplicates and subtitles. Subsequently, NetworkX [16] is used to construct the knowledge graph, which represents the logical relationships between events in the video. To mine long-term causal dependencies, an initial node is randomly selected, and a depth-first search (DFS) is used to extract evidence chains. Each node in these chains contains detailed information about the head and tail entities, their relationship, timestamps, and scene descriptions, facilitating comprehensive reasoning over the video content. For each video, we sample multiple evidence chains, enforcing that the entity overlap between any two chains is below 10% to promote diversity.

Stream Thought QA Synthesis. The final phase leverages Gemini 3.0 flash as a data synthesizer. Conditioned on the video knowledge graph, the model first generates a streaming CoT rationale to actively reason over video events and dynamic content. Subsequently, aligned with a sampled evidence chain $\{\mathbf{z}^k\}_{k=1}^{K}$, it synthesizes a query $\mathbf{q}$ and the final answer $\mathbf{y}$, necessitating multi-evidence reasoning that integrates the CoT with visual context. To ensure data fidelity, we apply a strict post-generation filtering rubric, including: world-knowledge check, format alignment, logical consistency, repetition check, and thought validation.

Curation of VST training set. Following the above procedure, we generate 100K streaming-thought examples with videos from LLaVA-Vid [67] and Video-Marathon [31]. In addition, our full supervised fine-tuning corpus for VST-SFT includes 50K open-ended QA instances randomly sampled from LLaVA-Vid. For VST-RL, we train on 11K sampled questions, including multiple-choice questions from LLaVA-Vid, Video-Marathon, and Onethinker [11], as well as counting questions from RepCount [20].

3 Experiment

3.1 Implementation Details

We adopt Qwen2.5-VL [1] as our base offline VideoLLM, processing input videos at 2 fps. Both VST-SFT and VST-RL (7B model) training stages are conducted on 32 × 80GB VRAM GPUs, utilizing the datasets detailed in Sec. 2.3. The visual encoder and projection layer are frozen throughout the entire training process. For VST-SFT, each training sample follows a 128 second time limit,

and overlong raw videos are segmented into clips following Eq. (4). For VST-RL, we employ verl [40] with vLLM [25] and FSDP [68] backend. We configure the rollout batch size to 256 with a group size of $N = 8$, and define the reward function based on the correctness of the final answer. Additionally, following LongVILA-R1 [4], we leverage the paralleled encoding strategy during rollout to pre-compute video embeddings. During testing, following StreamingForest [62], we cap each inference step (including streaming-think and the final answer) at 8,192 video tokens and limit the max thinking times to 4 for efficient evaluation. We conduct all evaluations using the lmms-eval framework [65].

Table 1: Comparison of offline and online VideoLLMs on StreamingBench Real-Time understanding tasks.

Model	Venue	OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	**Overall**
Proprietary Models												
Gemini 1.5 pro [43]	-	79.0	80.5	83.5	79.7	80.0	84.7	77.8	64.2	72.0	48.7	75.7
GPT-4o [36]	-	77.1	80.5	83.9	76.5	70.2	83.8	66.7	62.2	69.1	49.2	73.3
Open-source Offline Models												
VILA-1.5-8B [30]	CVPR'24	53.7	49.2	71.0	56.9	53.4	53.9	54.6	48.8	50.1	17.6	52.3
LongVA-7B [66]	TMLR'25	70.0	63.3	61.2	70.9	62.7	59.5	61.1	53.7	54.7	34.7	60.0
MiniCPM-v2.6-7B [21]	COLM'24	71.9	71.1	77.9	75.8	64.6	65.7	70.4	56.1	62.3	53.4	67.4
LLaVA-OV-7B [26]	TMLR'25	80.4	74.2	76.0	80.7	72.7	71.7	67.6	65.5	65.7	45.1	71.1
Qwen2.5-VL-7B [1]	-	78.3	80.5	78.9	80.5	76.7	78.5	79.6	63.4	66.2	53.2	73.7
Open-source Online Models												
Flash-VStream-7B [64]	ICCV'25	25.9	43.6	24.9	23.9	27.3	13.1	18.5	25.2	23.9	48.7	23.2
VideoLLM-online-8B [2]	CVPR'24	39.1	40.1	34.5	31.1	46.0	32.4	31.5	34.2	42.5	27.9	36.0
Dispider-8B [37]	CVPR'25	74.9	75.5	74.1	73.1	74.4	59.9	76.1	62.9	62.2	45.8	67.6
TimeChatOnline-7B [58]	MM'25	80.2	82.0	79.5	83.3	76.1	78.5	78.7	64.6	69.6	58.0	75.4
Streamforest-7B [62]	NeurIPS'25	83.1	82.8	82.7	84.3	77.5	78.2	76.9	69.1	75.6	54.4	77.3
VST-7B (ours)	ECCV'26	85.4	82.0	86.4	89.1	74.2	87.2	82.4	73.1	73.9	47.3	79.5

3.2 Benchmarks

To demonstrate the effectiveness of our method, we conducted a comprehensive evaluation across five video understanding benchmarks. Specifically, StreamingBench [32] and OVO-Bench [35] are utilized for online video understanding, focusing on the model's online reasoning capabilities and temporal awareness. VideoMME [12] serves as a comprehensive offline benchmark covering diverse domains and varying video durations. LongVideoBench [51] is designed to evaluate the long-form video understanding capabilities, while Video-Holmes [5] emphasizes logical reasoning within video content.

3.3 Online Video Benchmark Results

As shown in Tabs. 1 and 2, we evaluate our model on StreamingBench and OVO-Bench. VST-7B achieves 79.5% on StreamingBench and 59.3% on OVO-

Bench, clearly outperforming prior open-source streaming SOTA models, including Streamforest [62] (77.3%) on StreamingBench and Streamo [53] (57.9%) on OVO-Bench. Notably, despite being much smaller than proprietary models, our method surpasses GPT-4o and Gemini 1.5 pro on StreamingBench by 6.2% and 3.8%, respectively, and achieves comparable performance with GPT-4o on OVO-Bench. Beyond the overall scores, VST-7B is particularly strong on OVO-Bench's Backward Tracing task, where it achieves 56.7%, outperforming Streamforest by +4.7%. This result indicates that our model can retain and retrieve historical information effectively, supporting sustained memory over streaming inputs. These results highlight the strength of our approach for streaming video understanding. We believe the gains stem from our VST paradigm and a tailored post-training recipe, which together improve the model's ability.

Table 2: Comparison of offline and online VideoLLMs on OVO-Bench.

Model	Venue	Real-Time							Backward				Forward				**Overall**
		OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.	REC	SSR	CRR	Avg.	Avg.
Proprietary Models																	
Gemini 1.5 pro [43]	-	85.9	67.0	79.3	58.4	63.4	62.0	69.3	58.6	76.4	52.6	62.5	35.5	74.2	61.7	57.2	63.0
GPT-4o [36]	-	69.8	64.2	71.6	51.1	70.3	59.8	64.5	57.9	75.7	48.7	60.8	27.6	73.2	59.4	53.4	59.5
Open-source Offline Models																	
Qwen2-VL-72B [47]	-	65.8	60.6	69.8	51.7	69.3	54.4	61.9	52.5	60.8	57.5	57.0	38.8	64.1	45.0	49.3	56.3
LLaVA-Video-7B [67]	TMLR'25	69.1	58.7	68.8	49.4	74.3	59.8	63.5	56.2	57.4	7.5	40.4	34.1	70.0	60.4	54.8	52.9
LLaVA-OV-7B [26]	TMLR'25	66.4	57.8	73.3	53.4	71.3	62.0	64.0	54.2	55.4	21.5	43.7	25.6	67.1	58.8	50.5	52.7
LongVU-7B [39]	ICML'25	53.7	53.2	62.9	47.8	68.3	59.8	57.6	40.7	59.5	4.8	35.0	12.2	69.5	60.8	47.5	46.7
Open-source Online Models																	
VideoLLM-online-8B [2]	CVPR'24	8.1	23.9	12.1	14.0	45.5	21.2	20.8	22.2	18.8	12.2	17.7	-	-	-	-	-
Dispider-8B [37]	CVPR'25	57.7	49.5	62.1	44.9	61.4	51.6	54.6	48.5	55.4	4.3	36.1	18.1	37.4	48.8	34.7	41.8
TimeChatOnline-7B [58]	MM'25	75.2	46.8	70.7	47.8	69.3	61.4	61.9	55.9	59.5	9.7	41.7	31.6	38.5	40.0	36.7	46.7
Streamforest-7B [62]	NeurIPS'25	68.5	53.2	71.6	47.8	65.4	60.9	61.2	58.9	64.9	32.3	52.0	32.8	70.6	57.1	52.5	55.6
Streamo-7B [53]	CVPR'26	77.2	66.1	76.7	45.5	66.3	72.8	67.4	55.6	58.1	33.9	49.2	30.8	57.6	82.5	57.0	57.9
VST-7B (ours)	ECCV'26	80.5	55.1	72.4	55.1	76.2	64.1	67.2	56.9	64.9	48.4	56.7	33.0	66.9	62.1	54.0	59.3

3.4 Offline Video Benchmark Results

In Tab. 3, we evaluate VST-7B on three offline video benchmarks, including VideoMME, LongVideoBench, and VideoHolmes. The results show that VST-7B delivers competitive performance across all three datasets, with particularly strong gains on long-video understanding and complex reasoning. On long-video benchmarks, VST-7B achieves 55.3% on VideoMME-long, outperforming TimeChat-Online by +6.9%, and 58.0% on LongVideoBench, exceeding it by +2.6%. On the reasoning benchmark VideoHolmes, VST-7B reaches 41.9%, surpassing Video-R1 by +5.4%. We attribute these improvements to our streaming thinking framework, which enables dynamic thinking over long videos to build long-term memory, and leverages both historical memory and current visual context for deep reasoning.

Table 3: Comparison of offline and online VideoLLMs on VideoMME (without subtitles), LongVideoBench, and VideoHolmes.

Model	Venue	**VideoMME** w/o sub.		**LongVideoBench**	**VideoHolmes**
		Long	Overall		
Proprietary Models					
Gemini 1.5 pro [43]	-	67.4	75.0	64.0	45.7
GPT-4o [36]	-	65.3	71.9	66.7	42.0
Open-source Offline Models					
LongVA-7B [66]	TMLR'25	47.6	54.3	56.3	-
Video-R1-7B [10]	NeurIPS'25	-	61.4	-	36.5
LongVILA-R1-7B [4]	NeurIPS'25	55.2	65.1	58.0	-
REVISOR-7B [27]	CVPR'26	56.2	65.7	57.5	-
Open-source Online Models					
Dispider-7B [37]	CVPR'25	-	57.2	-	-
Streamforest-7B [62]	NeurIPS'25	-	61.4	-	-
TimeChatOnline-7B [58]	MM'25	48.4	62.4	55.4	-
VST-7B (Ours)	ECCV'26	55.3	64.9	58.0	41.9

Table 4: Ablation study on VST training schedule.

Model & Config	**OVO-Bench**			**VideoMME**
	Backward	Forward	Overall	w/o sub. Overall
Qwen2.5-VL-7B (Base model)	47.5	41.9	50.5	62.9
Ablation on VST-SFT training data				
+LLava-Vid (50K)	49.9	42.4	52.3	61.8
+LLava-Vid (30K) & VST (20K)	52.0	50.1	56.8	62.5
+LLava-Vid (20K) & VST (30K)	53.3	50.0	57.1	63.1
Ablation on different training stage				
+VST-SFT	56.7	48.5	57.4	63.0
+VST-RL	49.3	54.6	56.8	62.8
+VST-SFT & VST-RL	56.7	54.0	59.3	64.9

3.5 Ablation Study

Ablation on training schedule. As shown in Tab. 4, we first analyze the composition of the SFT training data. Mixing our VST data with the LLaVA-Vid QA dataset consistently improves online video understanding. Specifically, compared to using 50K LLaVA-Vid data alone, the mix of 20K LLaVA-Vid and 30K VST data achieves a +4.8% gain on the OVO-Bench. Furthermore, the ablation on different training stages demonstrates that our training strategies effectively enhance online video capabilities. Interestingly, we find that VST-SFT primarily benefits the model's backward memory capacity (+9.2% improvement on Backward track), while VST-RL is advantageous for forward prediction capabilities (improving the Forward score of +12.7%). Finally, combining both stages (VST-SFT & VST-RL) yields the highest overall performance on both OVO-Bench (59.3%) and VideoMME (64.9%).

Table 5: Ablation study on the scale of the base offline VideoLLM. All results in this table are obtained with the VST inference pipeline, where Qwen is evaluated as a zero-shot inference baseline.

Size	Model	OVOB. Overall	StreamingB. Realtime	V-MME w/o sub. Overall	LongVideoB.	VideoHolmes
3B	Qwen2.5-VL	53.1	67.8	57.9	53.3	30.7
	VST	56.2	75.5	59.5	54.1	36.1
7B	Qwen2.5-VL	55.0	71.7	62.3	54.7	32.9
	VST	59.3	79.5	64.9	58.0	41.9
32B	Qwen2.5-VL	60.1	71.5	65.8	59.8	40.1
	VST	63.5	80.7	67.2	60.7	45.1

Ablation on Streaming Thinking Times at Inference. Fig. 5 analyzes the impact of maximum streaming thinking times on OVO-Bench. For the Backward task, accuracy increases from 53.3% and grows continuously from 1 to 16 steps, ultimately reaching 57.5%. This demonstrates that additional thinking steps help generate precise memories for backward tracing. For the Real-Time and Forward tasks, initial thinking steps clearly aid in understanding visual information. However, performance reaches a plateau for ≥ 4 steps, as excessive memory details introduce redundancy.

Ablation on Base Model Size. Tab. 5 examines the impact of the base model capacity. Note that the base Qwen rows here are evaluated under the VST streaming inference pipeline (zero-shot), and thus differ from the native offline base in Tab. 4; this isolates the contribution of training from that of the inference protocol. We apply our two-stage training recipe (VST-SFT and VST-RL) to the Qwen2.5-VL-Instruct models at 3B, 7B, and 32B scales. The Video Streaming Thinking paradigm yields consistent improvements across all online and offline benchmarks regardless of the model size. For instance, on StreamingBench, VST achieves absolute accuracy gains of +7.7%, +7.8%, and +9.2% over the 3B, 7B, and 32B base models, respectively. Similar consistent enhancements are observed on complex tasks like VideoHolmes (+5.4%, +9.0%, and +5.0%). These results demonstrate that our proposed method is highly parameter-scalable.

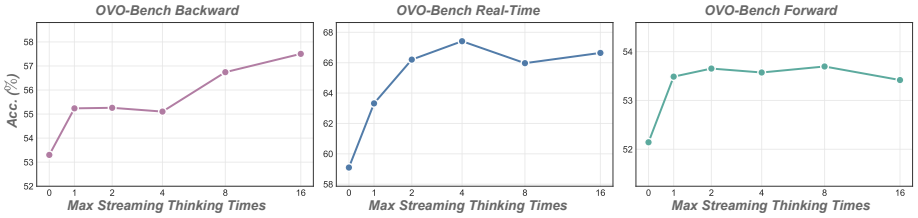


Fig. 5: Ablation study on max thinking times.

3.6 Analysis

Efficiency Analysis. We compare the QA latency of several offline and online methods under the same experimental setup. All measurements are conducted on VideoHolmes, as shown in Tab. 6. Models without CoT directly output the final answer. Benefiting from our query-ahead streaming think mechanism, VST maintains significantly lower response latency. Moreover, streaming think is executed asynchronously before the query and finishes within the clip inter-arrival interval, so its computation is amortized over playback rather than added after the query. As a result, it does not increase the real-world end-to-end inference time.

Table 6: Inference Latency.

Online / Offline Method	**QA Latency**
Qwen2.5-VL-7B	0.54s
Qwen2.5-VL-7B w/CoT	5.30s
Video-R1 w/CoT	8.80s
VideoLLM-online-8B	0.38s
Dispider-7B	1.10s
VST-3B (Ours)	0.53s
VST-7B (Ours)	0.56s
VST-32B (Ours)	1.40s

In our implementation, streaming thoughts are generated asynchronously every 16–32 seconds and take 7.0 seconds on average, with a P99 latency of 11.2 seconds. Since this is shorter than the minimum trigger interval, the overhead is amortized over playback. If interrupted by a query or a new step, VST uses the latest completed memory state. Therefore, VST adds background computation but not post-query response latency.

Case Study. Fig. 6 presents a case study from VideoHolmes. The query requires temporal reasoning over disjoint segments, specifically aligning repeated visualizations of a wall clock with the subsequent appearance of a "blurred-face man". The baseline Video-R1-7B, which relies on post-query thinking, fails to capture these dispersed temporal cues due to the difficulty of attending to specific evidence across a long context. Consequently, it hallucinates a spurious correlation involving object interactions, leading to a logical error. Furthermore, this retrospective reasoning incurs a significant latency of 9.53s. In contrast, VST-7B employs streaming thinking to continuously update its evidence (e.g., timestamps and event triggers) as the video memories. This pre-query evidence accumulation allows VST to correctly deduce the time-based rule and, by shifting the reasoning burden to the streaming phase, drastically reduces the response latency to 0.51s. This comparison demonstrates that pre-query streaming thinking simultaneously enhances reasoning robustness and system responsiveness.

We further examine typical failure cases of VST during inference. While streaming thinking improves evidence accumulation, the remaining errors are mainly associated with imperfect memory construction. Specifically, VST may store visually salient but query-irrelevant details, compress early evidence too aggressively, miss fine-grained temporal spans, or fail to associate weak cues across distant events. These observations suggest that more fine-grained and faithful memory construction, especially for preserving subtle temporal evidence and cross-event relations, is a promising direction for future work.

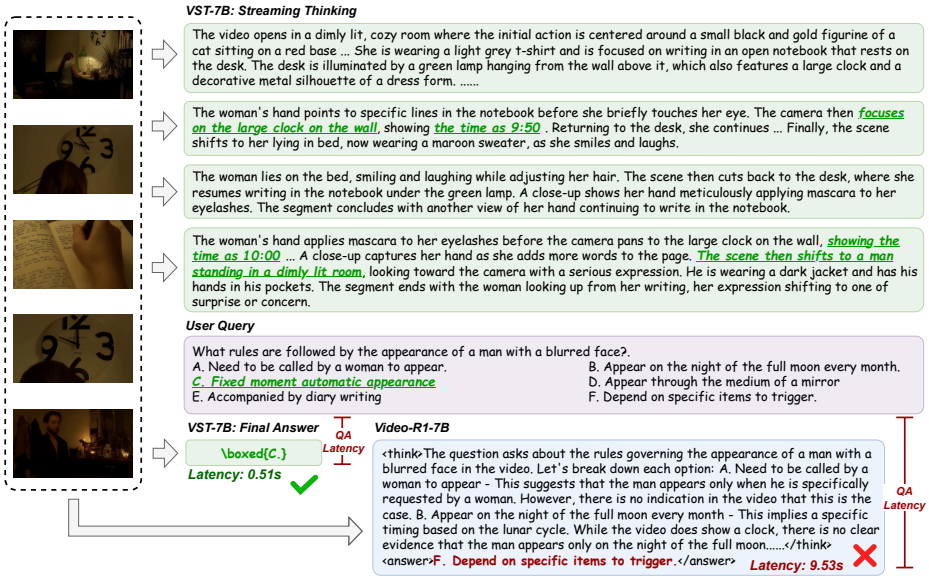


Fig. 6: Case Study from VideoHolmes. We compare VST-7B with Video-R1-7B. VST-7B processes the video stream and performs streaming thinking before the query, then answers directly once the query arrives. In contrast, Video-R1-7B generates CoT after the query, resulting in higher QA latency. VST-7B achieves better performance with lower QA latency in this example.

4 Related Work

Streaming Video Understanding. Streaming video understanding processes continuous visual inputs of indeterminate length. Unlike offline methods, the lack of global sampling and restricted context windows poses significant challenges for VideoLLMs. Some existing methods attempt to retain extended video information within limited context lengths through real-time visual token compression [2,38,41,55,58,62]. For instance, rLiVS reduces streaming token redundancy by recurrently pruning attention-uninformative visual tokens, echoing broader attention-reuse strategies for efficient visual processing [7, 59]. Others incorporate external memory mechanisms to recall historical information via similarity ranking [6,37,57,64]. However, these methods rely on static heuristics, lacking autonomous memory management and the ability to perform complex, multi-step reasoning. To bridge this gap, recent studies have introduced streaming and long-context reasoning methods into video understanding [46,50]. Along this line, this paper proposes ***V**ideo **S**treaming **T**hinking* (VST), which introduces an online thinking process that evolves with the video stream. By coupling autonomous memory management with in-depth instruction analysis, VST enables models to transcend short-range perception and achieve robust streaming intelligence.

VideoLLMs Test-Time Scaling. Following the breakthrough of test-time scaling and chain-of-thought in LLMs [15,44,49], recent VideoLLMs have adopted supervised fine-tuning (SFT) to mimic expert reasoning trajectories [17, 18] or utilized R1-style reinforcement learning (RL) to enhance task performance [4, 10, 23, 27, 48, 56]. Despite these advances, existing post-training research remains predominantly confined to offline video understanding. The exploration of reasoning within streaming contexts, particularly regarding long-horizon cognitive capabilities, remains a critical yet neglected frontier. In this paper, we introduce a unified SFT and RL framework for streaming video understanding. Our method achieves a synergistic balance between real-time responsiveness and sophisticated reasoning, enabling autonomous memory management and in-depth analysis of evolving video streams.

5 Conclusion

In this paper, we propose ***V**ideo **S**treaming **T**hinking* (VST), a new paradigm for streaming video understanding that introduces a synchronized stream of logical inference with real-time responsiveness. VST enables a *thinking-while-watching* mechanism that performs reasoning over incoming clips during streaming. We further develop a post-training recipe (VST-SFT and VST-RL) and an automated data synthesis pipeline based on video knowledge graphs to produce streaming-thought supervision. Empirically, VST delivers strong performance across multiple online and offline video understanding benchmarks, and yields consistent gains across VideoLLMs ranging from 3B to 32B parameters, indicating that the approach generalizes across model scales. Overall, our study establishes VST as a practical test-time scaling approach for streaming scenarios, simultaneously enabling explicit CoT generation and real-time responsiveness.

Limitation and Future Works. While the computation of streaming thoughts can be scheduled in parallel with incoming video clips, the additional LLM token consumption is still non-negligible. A promising direction is to explore *latent reasoning* to enable more token-efficient streaming thinking. Moreover, VST primarily focuses on text-guided memory management, which is orthogonal to existing streaming visual memory mechanisms. Investigating their combination and potential synergy is an interesting avenue for future work.

Acknowledgements

This work was done during the research internship of Yiran Guan and Liang Yin at Xiaomi Inc.
This work is supported by the NSFC (62225603).

References

1. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
2. Chen, J., Lv, Z., Wu, S., Lin, K.Q., Song, C., Gao, D., Liu, J.W., Gao, Z., Mao, D., Shou, M.Z.: Videollm-online: Online video large language model for streaming video. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 18407–18418 (2024)
3. Chen, J., Zeng, Z., Lin, Y., Li, W., Ma, Z., Shou, M.Z.: Livecc: Learning video llm with streaming speech transcription at scale. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 29083–29095 (2025)
4. Chen, Y., Huang, W., Shi, B., Hu, Q., Ye, H., Zhu, L., Liu, Z., Molchanov, P., Kautz, J., Qi, X., et al.: Scaling rl to long videos. In: Proc. of Advances in Neural Information Processing Systems (2025)
5. Cheng, J., Ge, Y., Wang, T., Ge, Y., Liao, J., Shan, Y.: Video-holmes: Can mllm think like holmes for complex video reasoning? arXiv preprint arXiv:2505.21374 (2025)
6. Di, S., Yu, Z., Zhang, G., Li, H., Cheng, H., Li, B., He, W., Shu, F., Jiang, H., et al.: Streaming video question-answering with in-context video kv-cache retrieval. In: Proc. of Intl. Conf. on Learning Representations (2025)
7. Dorovatas, E., Seifi, S., Gupta, G., Aljundi, R.: Recurrent attention-based token selection for efficient streaming video-llms. Proc. of Advances in Neural Information Processing Systems **38**, 144088–144114 (2025)
8. Driess, D., Xia, F., Sajjadi, M.S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al.: Palm-e: an embodied multimodal language model. In: Proc. of Intl. Conf. on Machine Learning. pp. 8469–8488 (2023)
9. Fang, H., Li, S., Wang, S., Xi, X., Liang, D., Bai, X.: Towards generalizable robotic manipulation in dynamic environments. arXiv preprint arXiv:2603.15620 (2026)
10. Feng, K., Gong, K., Li, B., Guo, Z., Wang, Y., Peng, T., Wu, J., Zhang, X., Wang, B., Yue, X.: Video-r1: Reinforcing video reasoning in mllms. In: Proc. of Advances in Neural Information Processing Systems (2025)
11. Feng, K., Zhang, M., Li, H., Fan, K., Chen, S., Jiang, Y., Zheng, D., Sun, P., Zhang, Y., Sun, H., et al.: Onethinker: All-in-one reasoning model for image and video. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (2026)
12. Fu, C., Dai, Y., Luo, Y., Li, L., Ren, S., Zhang, R., Wang, Z., Zhou, C., Shen, Y., Zhang, M., et al.: Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 24108–24118 (2025)
13. Fu, H., Zhang, D., Zhao, Z., Cui, J., Xie, H., Wang, B., Chen, G., Liang, D., Bai, X.: Minddrive: A vision-language-action model for autonomous driving via online reinforcement learning. arXiv preprint arXiv:2512.13636 (2025)
14. Guan, Y., Tu, S., Liang, D., Zhu, L., Ju, J., Luo, Z., Luan, J., Liu, Y., Bai, X.: Thinkomni: Lifting textual reasoning to omni-modal scenarios via guidance decoding. In: Proc. of Intl. Conf. on Learning Representations (2026)
15. Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., et al.: Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. Nature **645**(8081), 633–638 (2025)

16. Hagberg, A., Swart, P.J., Schult, D.A.: Exploring network structure, dynamics, and function using networkx. Tech. rep., Los Alamos National Laboratory (LANL) (2007)
17. Han, S., Huang, W., Shi, H., Zhuo, L., Su, X., Zhang, S., Zhou, X., Qi, X., Liao, Y., Liu, S.: Videoespresso: A large-scale chain-of-thought dataset for fine-grained video reasoning via core frame selection. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 26181–26191 (2025)
18. Hannan, T., Islam, M.M., Gu, J., Seidl, T., Bertasius, G.: Revisionllm: Recursive vision-language model for temporal grounding in hour-long videos. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 19012–19022 (2025)
19. Hasson, U., Nir, Y., Levy, I., Fuhrmann, G., Malach, R.: Intersubject synchronization of cortical activity during natural vision. Science **303**(5664), 1634–1640 (2004)
20. Hu, H., Dong, S., Zhao, Y., Lian, D., Li, Z., Gao, S.: Transrac: Encoding multi-scale temporal correlation with transformers for repetitive action counting. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 19013–19022 (2022)
21. Hu, S., Tu, Y., Han, X., Cui, G., He, C., Zhao, W., Long, X., Zheng, Z., Fang, Y., Huang, Y., Zhang, X., Thai, Z.L., Wang, C., Yao, Y., Zhao, C., Zhou, J., Cai, J., Zhai, Z., Ding, N., Jia, C., Zeng, G., dahai li, Liu, Z., Sun, M.: MiniCPM: Unveiling the potential of small language models with scalable training strategies. In: Conference on Language Modeling (2024)
22. Huang, M., Liu, Y., Liang, D., Jin, L., Bai, X.: Mini-monkey: Alleviating the semantic sawtooth effect for lightweight mllms via complementary image pyramid. In: Proc. of Intl. Conf. on Learning Representations. vol. 2025, pp. 85804–85823 (2025)
23. Jiang, T., Wu, L., Xia, S., Li, S., Yan, Z., Yang, H., Qiao, Y., Wang, Y.: Imagine before you predict: Interleaved latent visual reasoning for video event prediction. arXiv preprint arXiv:2606.05769 (2026)
24. Jiang, T., Xia, S., Xu, Y., Wu, L., Zeng, X., Wang, L., Qiao, Y., Wang, Y.: Vknowu: Evaluating visual knowledge understanding in multimodal llms. arXiv preprint arXiv:2511.20272 (2025)
25. Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C.H., Gonzalez, J.E., Zhang, H., Stoica, I.: Efficient memory management for large language model serving with pagedattention. In: Proc. of the ACM SIGOPS 29th Symposium on Operating Systems Principles (2023)
26. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., Li, C.: LLaVA-onevision: Easy visual task transfer. Transactions on Machine Learning Research (2025)
27. Li, J., Yin, H., Tan, W., Chen, J., Xu, B., Qu, Y., Chen, Y., Ju, J., Luo, Z., Luan, J.: Revisor: Beyond textual reflection, towards multimodal introspective reasoning in long-form video understanding. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (2026)
28. Li, Z., Yang, B., Liu, Q., Ma, Z., Zhang, S., Yang, J., Sun, Y., Liu, Y., Bai, X.: Monkey: Image resolution and text label are important things for large multi-modal models. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 26763–26773 (2024)
29. Liang, D., Zhang, C., Xu, X., Ju, J., Luo, Z., Bai, X.: Cook and clean together: Teaching embodied agents for parallel task execution. In: Proc. of the AAAI Conf. on Artificial Intelligence (2026)

30. Lin, J., Yin, H., Ping, W., Molchanov, P., Shoeybi, M., Han, S.: Vila: On pre-training for visual language models. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 26689–26699 (2024)
31. Lin, J., Wu, J., Sun, X., Wang, Z., Liu, J., Su, Y., Yu, X., Chen, H., Luo, J., Liu, Z., et al.: Unleashing hour-scale video training for long video-language understanding. Proc. of Advances in Neural Information Processing Systems **38**, 17523–17552 (2025)
32. Lin, J., Fang, Z., Chen, C., Cheng, H., Wan, Z., Luo, F., Wang, Z., Li, P., Liu, Y., Sun, M.: Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. In: Proc. of IEEE Intl. Conf. on Acoustics, Speech and Signal Processing. pp. 12147–12151 (2026)
33. Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W.S., Lin, M.: Understanding r1-zero-like training: A critical perspective. In: Conference on Language Modeling (2025)
34. Ning, Z., Liu, G., Jin, Q., Ding, W., Guo, M., Zhao, J.: Livevlm: Efficient online video understanding via streaming-oriented kv cache and retrieval. arXiv preprint arXiv:2505.15269 (2025)
35. Niu, J., Li, Y., Miao, Z., Ge, C., Zhou, Y., He, Q., Dong, X., Duan, H., Ding, S., Qian, R., et al.: Ovo-bench: How far is your video-llms from real-world online video understanding? In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 18902–18913 (2025)
36. OpenAI: Gpt-4o system card (2024)
37. Qian, R., Ding, S., Dong, X., Zhang, P., Zang, Y., Cao, Y., Lin, D., Wang, J.: Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 24045–24055 (2025)
38. Qian, R., Dong, X., Zhang, P., Zang, Y., Ding, S., Lin, D., Wang, J.: Streaming long video understanding with large language models. In: Proc. of Advances in Neural Information Processing Systems. vol. 37, pp. 119336–119360 (2024)
39. Shen, X., Xiong, Y., Zhao, C., Wu, L., Chen, J., Zhu, C., Liu, Z., Xiao, F., Varadarajan, B., Bordes, F., et al.: Longvu: Spatiotemporal adaptive compression for long video-language understanding. In: Proc. of Intl. Conf. on Machine Learning (2025)
40. Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., Wu, C.: Hybridflow: A flexible and efficient rlhf framework. arXiv preprint arXiv: 2409.19256 (2024)
41. Song, E., Chai, W., Wang, G., Zhang, Y., Zhou, H., Wu, F., Chi, H., Guo, X., Ye, T., Zhang, Y., et al.: Moviechat: From dense token to sparse memory for long video understanding. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 18221–18232 (2024)
42. Stephens, G.J., Silbert, L.J., Hasson, U.: Speaker–listener neural coupling underlies successful communication. Proc. of the National Academy of Sciences **107**(32), 14425–14430 (2010)
43. Team, G., Georgiev, P., Lei, V.I., Burnell, R., Bai, L., Gulati, A., Tanzer, G., Vincent, D., Pan, Z., Wang, S., et al.: Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530 (2024)
44. Tong, J., Fan, Y., Zhao, A., Ma, Y., Shen, X.: Streamingthinker: Large language models can think while reading. In: Proc. of Intl. Conf. on Learning Representations (2026)

45. Wang, C., Tan, X., Cao, J., Li, K., Chen, T.: Curvestream: Boosting streaming video understanding in mllms via curvature-aware hierarchical visual memory management. arXiv preprint arXiv:2603.19571 (2026)
46. Wang, L., Jin, Z., Hao, Y., Chen, Y., Liu, K., Ao, Y., Zhao, J.: Think while watching: Online streaming segment-level memory for multi-turn video reasoning in multimodal large language models. arXiv preprint arXiv:2603.11896 (2026)
47. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024)
48. Wang, Q., Yu, Y., Yuan, Y., Mao, R., Zhou, T.: VideoRFT: Incentivizing video reasoning capability in MLLMs via reinforced fine-tuning. In: Proc. of Advances in Neural Information Processing Systems (2025)
49. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. In: Proc. of Advances in Neural Information Processing Systems. vol. 35, pp. 24824–24837 (2022)
50. Wen, S., Wang, Z., Zhang, X., Huang, L., Wu, W.: Eventmemagent: Hierarchical event-centric memory for online video understanding with adaptive tool use. arXiv preprint arXiv:2602.15329 (2026)
51. Wu, H., Li, D., Chen, B., Li, J.: Longvideobench: A benchmark for long-context interleaved video-language understanding. In: Proc. of Advances in Neural Information Processing Systems. vol. 37, pp. 28828–28857 (2024)
52. Wu, X., Liang, D., Feng, T., Xia, K., Zhang, Y., Li, X., Tan, X., Bai, X.: Generation models know space: Unleashing implicit 3d priors for scene understanding. arXiv preprint arXiv:2603.19235 (2026)
53. Xia, J., Chen, P., Zhang, M., Sun, X., Zhou, K.: Streaming video instruction tuning. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (2026)
54. Xiaomi, L.C.T.: Mimo-vl technical report (2025), https://arxiv.org/abs/2506.03569
55. Xu, R., Xiao, G., Chen, Y., He, L., Peng, K., Lu, Y., Han, S.: StreamingVLM: Real-time understanding for infinite video streams. In: Proc. of Intl. Conf. on Learning Representations (2026)
56. Yan, Z., He, Y., Li, X., Yue, Z., Zeng, X., Wang, Y., Qiao, Y., Wang, L., Wang, Y.: Videochat-r1.5: Visual test-time scaling to reinforce multimodal reasoning by iterative perception. In: Proc. of Advances in Neural Information Processing Systems (2025)
57. Yang, Y., Zhao, Z., Shukla, S.N., Singh, A., Mishra, S.K., Zhang, L., Ren, M.: Streammem: Query-agnostic kv cache memory for streaming video understanding. arXiv preprint arXiv:2508.15717 (2025)
58. Yao, L., Li, Y., Wei, Y., Li, L., Ren, S., Liu, Y., Ouyang, K., Wang, L., Li, S., Li, S., et al.: Timechat-online: 80% visual tokens are naturally redundant in streaming videos. In: Proc. of ACM Intl. Conf. on Multimedia. pp. 10807–10816 (2025)
59. Yin, L., Xie, X., Li, Z., Bai, X., Liu, Y.: Mstar: Box-free multi-query scene text retrieval with attention recycling. Proc. of Advances in Neural Information Processing Systems **38**, 57856–57881 (2025)
60. Yu, H., Chen, T., Feng, J., Chen, J., Dai, W., Yu, Q., Zhang, Y.Q., Ma, W.Y., Liu, J., Wang, M., et al.: Memagent: Reshaping long-context LLM with multi-conv RL-based memory agent. In: Proc. of Intl. Conf. on Learning Representations (2026)
61. Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al.: Dapo: An open-source llm reinforcement learning system at scale. In: Proc. of Advances in Neural Information Processing Systems (2025)

62. Zeng, X., Qiu, K., Zhang, Q., Li, X., Wang, J., Li, J., Yan, Z., Tian, K., Tian, M., Zhao, X., et al.: Streamforest: Efficient online video understanding with persistent event memory. In: Proc. of Advances in Neural Information Processing Systems (2025)
63. Zeng, X., Zhang, Z., Zhu, Y., Li, X., Wang, Z., Ma, C., Zhang, Q., Huang, Z., Ouyang, K., Jiang, T., et al.: Video-o3: Native interleaved clue seeking for long video multi-hop reasoning. In: Proc. of Intl. Conf. on Machine Learning (2026)
64. Zhang, H., Wang, Y., Tang, Y., Liu, Y., Feng, J., Jin, X.: Flash-vstream: Efficient real-time understanding for long video streams. In: Proc. of IEEE Intl. Conf. on Computer Vision. pp. 21059–21069 (2025)
65. Zhang, K., Li, B., Zhang, P., Pu, F., Cahyono, J.A., Hu, K., Liu, S., Zhang, Y., Yang, J., Li, C., Liu, Z.: Lmms-eval: Reality check on the evaluation of large multimodal models (2024), https://arxiv.org/abs/2407.12772
66. Zhang, P., Zhang, K., Li, B., Zeng, G., Yang, J., Zhang, Y., Wang, Z., Tan, H., Li, C., Liu, Z.: Long context transfer from language to vision. Transactions on Machine Learning Research (2025)
67. Zhang, Y., Wu, J., Li, W., Li, B., MA, Z., Liu, Z., Li, C.: LLaVA-video: Video instruction tuning with synthetic data. Transactions on Machine Learning Research (2025)
68. Zhao, Y., Gu, A., Varma, R., Luo, L., Huang, C.C., Xu, M., Wright, L., Shojanazeri, H., Ott, M., Shleifer, S., et al.: Pytorch fsdp: experiences on scaling fully sharded data parallel. arXiv preprint arXiv:2304.11277 (2023)
69. Zhu, L., Guan, Y., Liang, D., Ju, J., Luo, Z., Qin, B., Luan, J., Liu, Y., Bai, X.: Shuffle-r1: Efficient rl framework for multimodal large language models via data-centric dynamic shuffle. In: Proc. of Intl. Conf. on Learning Representations (2026)