Thinking in Streaming Video

Zikang Liu[1,2⋆], Longteng Guo[1⋆], Handong Li[1,2⋆], Ru Zhen[3], Xingjian He[1], Ruyi Ji[1], Xiaoming Ren[3], Yanhao Zhang[3], Haonan Lu[3], and Jing Liu[1,2⋆⋆]

[1] Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] OPPO AI Center, OPPO Inc.
{liuzikang2023}@ia.ac.cn, {longteng.guo,jliu}@nlpr.ia.ac.cn

Abstract. Real-time understanding of continuous video streams is essential for interactive assistants and multimodal agents operating in dynamic environments. However, most existing video reasoning approaches follow a batch paradigm that defers reasoning until the full video context is observed, resulting in high latency and growing computational cost that are incompatible with streaming scenarios. In this paper, we introduce **ThinkStream**, a framework for streaming video reasoning based on a **Watch–Think–Speak** paradigm that enables models to incrementally update their understanding as new video observations arrive. At each step, the model performs a short reasoning update and decides whether sufficient evidence has accumulated to produce a response. To support long-horizon streaming, we propose Reasoning-Compressed Streaming Memory (RCSM), which treats intermediate reasoning traces as compact semantic memory that replaces outdated visual tokens while preserving essential context. We further train the model using a Streaming Reinforcement Learning with Verifiable Rewards scheme that aligns incremental reasoning and response timing with the requirements of streaming interaction. Experiments on multiple streaming video benchmarks show that ThinkStream significantly outperforms existing online video models while maintaining low latency and memory usage. Code, models and data are available at `https://github.com/CASIA-IVA-Lab/ThinkStream`.

Keywords: Streaming Video · Incremental Reasoning · Memory

1 Introduction

Real-time understanding of continuous video streams is a core capability for interactive assistants and multimodal agents [25, 31, 32]. Unlike offline video question answering [1, 2, 12, 41], streaming scenarios impose several fundamental constraints: (i) *strict causality*, where reasoning must rely only on observations available up to the current moment; (ii) *low computation and memory usage*, requiring inference latency and resource usage to remain stable even as the stream

⋆ Equal contribution.
⋆⋆ Corresponding Author.

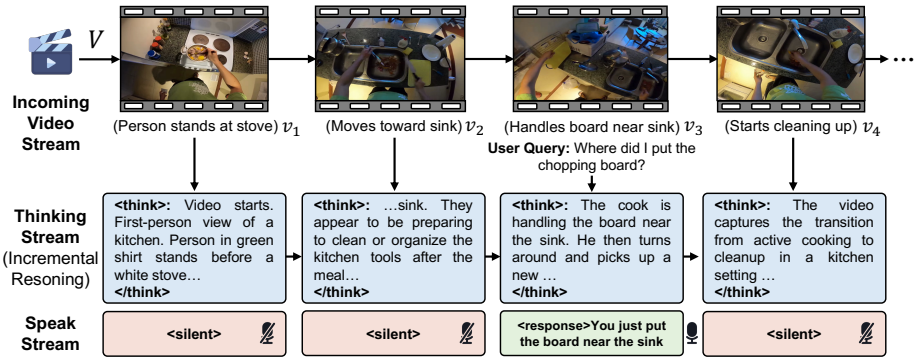

Fig. 1: Illustration of the **Streaming Watch–Think–Speak** paradigm. As video chunks arrive sequentially, the model continuously updates its understanding through incremental reasoning steps (`<think>`). Each update integrates newly observed evidence with accumulated context. Based on this evolving interpretation, the model decides whether sufficient evidence has been gathered to produce a response (`<response>`) or whether it should remain silent (`<silent>`) and continue observing the stream.

grows indefinitely; and (iii) *timely interaction*, which demands that the system maintain an up-to-date interpretation of the evolving scene to respond or act when necessary.

Such constraints commonly arise in environments where visual events unfold continuously, including real-time assistants observing user activities, monitoring systems detecting emerging events, and embodied agents interacting with dynamic physical environments.

However, most existing video reasoning paradigms remain fundamentally *batch*: the model first consumes a long video context and only then performs multi-step reasoning to produce an answer [7, 30]. This design introduces unacceptable latency and weakens the connection between reasoning steps and the evidence that triggered them. Increasing the depth or length of Chain-of-Thought (CoT) reasoning partially mitigates this issue, but long-form decoding further inflates latency and compute, making it incompatible with continuous streams.

In dynamic environments, human cognition often resembles *thinking while observing*: people form provisional interpretations from local evidence, refine them as new signals arrive, and act once sufficient confidence has been accumulated. This observation motivates an incremental perspective on streaming video understanding, where reasoning evolves alongside the incoming stream rather than being deferred until the full context is observed.

In this paper, we propose a **Streaming Watch–Think–Speak** paradigm for video understanding. At each step, the model observes a newly arrived video chunk and performs a short reasoning update that integrates the new evidence with previously accumulated context. Based on this evolving understanding, the model determines whether the available evidence is sufficient to produce a response, or whether it should remain silent and continue observing the stream

until more information arrives. As illustrated in Fig. 1, the model produces a structured output consisting of (i) a reasoning segment enclosed in a `<think>` block and (ii) an interaction action that either emits a response (`<response>`) or indicates that the model remains silent (`<silent>`) and continues observing the stream.

Building on this paradigm, we introduce **ThinkStream**, a framework that enables reasoning-capable multimodal models to operate over long-horizon video streams under low computational resources. The central challenge is to preserve coherent understanding without allowing the visual context, and thus the KV cache, to grow rapidly. To address this, we propose **Reasoning-Compressed Streaming Memory (RCSM)**, which treats intermediate reasoning traces as compact semantic memory. As the stream evolves, outdated visual tokens are evicted while the corresponding reasoning states are retained as long-term semantic anchors, preserving essential historical context while keeping inference cost stable.

To train models that produce such reasoning updates, we introduce a **Streaming Reinforcement Learning with Verifiable Rewards (RLVR)** framework tailored to Watch–Think–Speak. We design simple rule-based rewards that are automatically verifiable and reflect the requirements of streaming interaction, encouraging the model to generate structured reasoning traces, respond only when sufficient evidence has accumulated, and maintain answer correctness throughout the stream.

Finally, we support large-scale RLVR training and inference with an efficient streaming rollout backend that enables high-throughput inference with dynamic context updates. We also construct a large-scale dataset featuring time-grounded reasoning traces for cold-start supervision and a carefully formatted subset for RLVR training.

Extensive experiments validate the effectiveness of our approach. On dedicated streaming video benchmarks, ThinkStream achieves substantially stronger performance than existing online video models, with even a compact 3B-scale model outperforming significantly larger baselines. Meanwhile, our framework maintains low latency and memory usage over long video streams, enabling real-time inference while preserving strong performance on standard offline video benchmarks.

Our contributions are summarized as follows:

- We propose a **Streaming Watch–Think–Speak** paradigm that formulates streaming video understanding as an incremental reasoning and interaction process, enabling models to continuously update their interpretation while deciding when to respond.
- We introduce **ThinkStream**, a unified framework for long-horizon streaming video reasoning, together with Reasoning-Compressed Streaming Memory (RCSM) that treats reasoning traces as compact semantic memory to replace evicted visual tokens while keeping inference cost much lower.

- We develop a **Streaming Reinforcement Learning** with RLVR training scheme that aligns incremental reasoning and response timing through automatically verifiable reward signals.
- We support scalable training and deployment through an efficient streaming inference backend and construct a large-scale dataset with time-grounded reasoning traces. Our code, models, and datasets will be released to facilitate future research on streaming video reasoning.

2 Related Work

2.1 Streaming Video Understanding

Transitioning to streaming video for real-time interaction, early frameworks (e.g., VideoLLM-online [3]) rely on direct perception-response cycles lacking intermediate reasoning. To manage massive visual token influxes, subsequent methods introduced specialized memory architectures [13, 37, 42], KV cache compression [20,35,38] or KV cache sparsification/pruning [4,32]. However, their memory mechanisms remain purely visual, devoid of semantic compression and explicit logical deduction. Furthermore, addressing the proactive output timing ("when to speak") challenge, existing approaches (e.g., EgoSpeak [11], StreamMind [6]) heavily depend on dedicated classification heads or external event triggers. Conversely, our framework eschews external classifiers; by integrating a generative reasoning phase, the model autonomously determines whether to remain silent or respond, seamlessly unifying perception, reasoning, and reaction into a single generative process.

2.2 Reasoning in Multimodal Large Language Models

While integrating Chain-of-Thought and Reinforcement Learning has significantly advanced LLMs [33, 34], video-domain adaptations [5, 22, 23, 27, 28] remain confined to offline batch-processing, incompatible with real-time streaming constraints. Conversely, strictly text-based streaming reasoning frameworks like StreamingThinker [24] are ill-equipped for the concurrent think-and-respond demands and dense visual token bottlenecks of continuous video. Recognizing that reasoning advancements are fundamentally driven by RL, we introduce the first streaming RL approach to tackle these multimodal complexities. Furthermore, existing offline compression methods (e.g., LightThinker [39]) merely reduce cache size without leveraging reasoning as active memory. In contrast, our work explicitly repurposes CoT trajectories as compressed semantic memory. By aggressively evicting dense visual tokens while preserving reasoning chains, we propose a mechanism that simultaneously scales reasoning capabilities and bounds KV cache growth for continuous video streaming.

3 Streaming Watch–Think–Speak Paradigm

Human perception in continuous environments rarely proceeds as a purely reactive process. As new observations arrive, people constantly form provisional

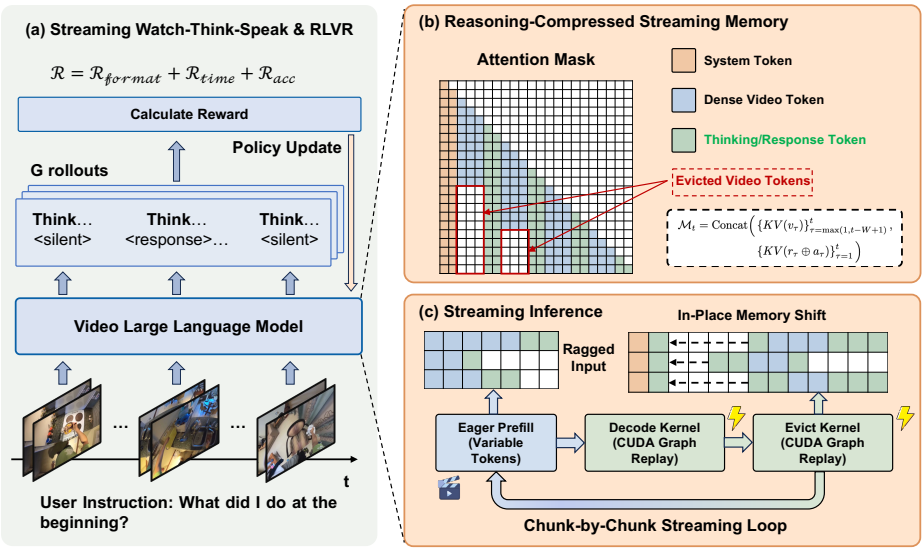

Fig. 2: Overview of the ThinkStream framework. (a) Streaming Watch-Think-Speak Paradigm & RLVR: The model undergoes streaming rollouts and policy updates driven by format, latency, and accuracy rewards. (b) Reasoning-Compressed Streaming Memory: Outdated dense video tokens are dynamically evicted from the KV cache, while highly compressed reasoning and response tokens are retained as long-term semantic anchors. (c) Streaming Inference: A custom backend utilizes Eager Prefill for variable tokens and replayable CUDA Graphs for both the Decode and Evict Kernels, enabling an efficient chunk-by-chunk streaming loop with in-place memory shifting.

interpretations, refine them with additional evidence, and occasionally act when the accumulated understanding becomes sufficiently reliable. This process interleaves perception and reasoning: local observations trigger short reasoning updates, and these incremental updates gradually consolidate into a coherent global interpretation of the scene.

3.1 Paradigm Design

Inspired by this cognitive pattern, we formulate streaming video understanding as a Watch–Think–Speak loop. Instead of separating perception and reasoning into two independent stages, the model performs incremental reasoning alongside the incoming stream. Each newly observed video segment triggers a short reasoning update that integrates the latest evidence with previously accumulated understanding. Over time, these updates progressively transform local perceptual signals into a structured interpretation of the evolving scene.

Within this paradigm, reasoning operates as a lightweight but continuous process. Upon observing a new video chunk, the model may (1) summarize newly observed events, (2) update hypotheses about ongoing activities or (3) refine

causal or temporal relations inferred earlier. These reasoning steps are intentionally short and localized, ensuring that they can be generated within strict latency constraints while still allowing the model to maintain a coherent internal narrative of the stream.

Following each reasoning update, the model determines whether to respond to the user. When sufficient evidence has accumulated, the model produces an explicit answer; otherwise it remains silent and continues observing the stream.

3.2 Formal Definition

Let the incoming video stream be represented as a sequence of temporal chunks $\mathcal{V} = \{v_1, v_2, \ldots, v_t, \ldots\}$,where v_t denotes the video segment observed at step t. Let $\mathcal{I}$ denote the user instruction or dialogue context. The model maintains a historical state $\mathcal{H}_{t-1}$ summarizing the accumulated context from previous steps.

Operationally, each streaming step produces a structured output consisting of a reasoning segment followed by an interaction decision. The output format is defined as $\langle\texttt{think}\rangle\ r_t\ \langle/\texttt{think}\rangle\ a_t$. Here r_t denotes the reasoning tokens generated at step t, and a_t represents the action taken by the model. The action either produces a response or explicitly indicates that the model remains silent.

Formally, given the incoming chunk v_t, the generation process follows

$$p(r_t, a_t \mid \mathcal{H}_{t-1}, v_t, \mathcal{I}) = \prod_{i=1}^{|r_t \oplus a_t|} \pi_\theta\big(y_i \mid y_{<i}, \mathcal{H}_{t-1}, v_t, \mathcal{I}\big), \tag{1}$$

where π_θ denotes the policy model and $\oplus$ represents sequence concatenation.

The action space is restricted to two possibilities:

$$a_t \in \big\{\langle\text{silent}\rangle,\ \langle\text{response}\rangle \oplus c_t\big\}, \tag{2}$$

where c_t denotes the generated response content. Emitting $\langle$silent$\rangle$ indicates that the model continues observing the stream without responding, while emitting $\langle$response$\rangle$ signals that the model has accumulated sufficient evidence to produce an answer.

This formulation allows reasoning to unfold incrementally along the temporal stream. Each reasoning segment r_t updates the model's internal understanding based solely on observations available up to time t, ensuring strict streaming causality. Meanwhile, the speaking decision a_t transforms the evolving reasoning state into an interaction policy that determines *when* the model should respond and when it should continue watching.

4 ThinkStream

Building upon our Watch-Think-Speak loop, we introduce **ThinkStream**, a unified framework that enables reasoning-capable multimodal models to operate under the strict computational constraints of continuous video streams. The

central challenge lies in reconciling two competing requirements: maintaining coherent long-horizon understanding over the video stream while preserving real-time inference efficiency.

4.1 Reasoning-Compressed Streaming Memory

Continuous video streams naturally produce a large number of visual tokens. If all tokens are retained in the KV cache, both memory usage and attention complexity grow monotonically with time, eventually rendering real-time inference infeasible. However, not all past visual observations are equally important. Early visual details can often be replaced by higher-level semantic summaries once the model has formed a stable interpretation of the scene.

Motivated by this observation, we introduce Reasoning-Compressed Streaming Memory (**RCSM**). The core idea of RCSM is to treat reasoning tokens as compressed semantic representations of earlier visual observations. Instead of storing the entire history of dense visual tokens, the model gradually replaces outdated visual features with the reasoning traces generated during the streaming reasoning process.

Reasoning-as-Compression State. In the Watch-Think-Speak paradigm, each reasoning segment r_t reflects the model's interpretation of the newly observed video chunk v_t conditioned on prior context. These reasoning tokens encode semantic abstractions such as events, temporal relations, and causal hypotheses derived from the raw visual input.

We therefore interpret the reasoning state as a *compression operator* that transforms dense perceptual evidence into compact symbolic representations. Formally, let $KV(\cdot)$ denote the cached key-value representations of tokens. Instead of preserving all visual tokens $\{v_1, \ldots, v_t\}$, the model maintains a mixed memory state consisting of recent visual tokens and accumulated reasoning tokens.

The memory state at step t is defined as

$$\mathcal{M}_t = \mathrm{Concat}\Big(\{KV(v_\tau)\}_{\tau=\max(1,t-W+1)}^{t}, \{KV(r_\tau \oplus a_\tau)\}_{\tau=1}^{t}\Big), \tag{3}$$

where W denotes the size of the visual sliding window. In this design, reasoning tokens act as long-term semantic anchors that preserve the essential interpretation of earlier observations. When the stream length exceeds the window W, the earliest visual tokens are evicted from the cache. Let v_{t-W} denote the oldest visual chunk within the window. When a new chunk v_{t+1} arrives, the tokens corresponding to v_{t-W} are removed from the KV cache. This design establishes a natural transition from dense perceptual memory to compressed semantic memory as the stream evolves.

Specifically, the number of retained visual tokens is upper-bounded by the window size, while reasoning tokens grow at a much slower rate due to their compact representation. Consequently, the effective context length remains stable even for long video streams. This property allows ThinkStream to sustain

real-time inference over continuous video input while preserving the semantic information required for long-horizon reasoning.

4.2 Streaming-Context RLVR Training

RCSM specifies *what* state is retained under streaming constraints; the remaining question is *how* to train the model to produce reasoning traces that are simultaneously useful for solving the current query and informative enough to serve as compressed long-term memory once dense visual tokens are evicted. To this end, we adopt a Streaming Reinforcement Learning with Verifiable Rewards (RLVR) setup tailored to the Watch-Think-Speak loop.

Streaming Rollout with Partial Reasoning. Let the video stream be represented as a sequence of chunks $\mathcal{V} = \{v_1, v_2, \dots\}$ with user instruction $\mathcal{I}$. At each step t, the policy observes the current chunk v_t alongside the accumulated streaming history $\mathcal{H}_{t-1}$ to produce an output tuple $o_t = (r_t, a_t)$.

The streaming history, maintained via the RCSM memory mechanism in Sec. 4.1, is denoted as:

$$\mathcal{H}_{t-1} = \{(v_\tau, r_\tau, a_\tau)\}_{\tau=1}^{t-1} \tag{4}$$

Since the model only has access to the temporal prefix $\mathcal{V}_{\leq t}$ at any given moment, the joint probability of the streaming rollout trajectory factorizes step-by-step as follows:

$$\pi_\theta(o_{1:T}|\mathcal{V}_{1:T}, \mathcal{I}) = \prod_{t=1}^{T} \pi_\theta(r_t, a_t|\mathcal{H}_{t-1}, v_t, \mathcal{I}) \tag{5}$$

RL Optimization. We adopt Group Relative Policy Optimization (GRPO) for model training. We sample G trajectories $\{o_{1:T}^{(1)}, \dots, o_{1:T}^{(G)}\}$ for each streaming instance, and optimize the policy using the standard clipped objective with KL regularization:

$$\mathcal{J}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G} \min\left(\rho_i(\theta)\hat{A}_i, \text{clip}(\rho_i(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i\right) - \beta D_{KL}(\pi_\theta \| \pi_{\text{ref}})\right]. \tag{6}$$

Reward Design. The reward function provides the essential training signal to align the policy with the Watch-Think-Speak loop under stringent streaming constraints. To this end, we formulate a rule-based reward system comprising three integral components: an accuracy reward, a format reward, and a time reward. The *accuracy reward* ($\mathcal{R}_{acc}$) assigns positive credit to responses that successfully match the ground-truth answer. To facilitate reliable automated verification, our RLVR samples are meticulously constructed utilizing deterministic formats, including multiple-choice, binary (Yes/No), and counting queries. Concurrently, the *format reward* ($\mathcal{R}_{format}$) enforces strict adherence to the structured interaction protocol mandated by the loop, yielding a positive

value strictly when the generated output conforms to the prescribed structural constraints, and zero otherwise.

Additionally, because streaming interactions necessitate temporally appropriate responses, we introduce a *time reward* ($\mathcal{R}_{time}$). This component quantifies the temporal discrepancy between the model's response step t_{resp} and the ground-truth step t_{gt}. We apply a linear decay within a predefined tolerance window w, formulated as $\mathcal{R}_{time} = \max(0, 1 - |t_{\text{resp}} - t_{gt}|/w)$. This mechanism effectively penalizes both premature inferences and excessive latency.

Ultimately, the total reward is computed as the linear summation of these three components:

$$\mathcal{R} = \mathcal{R}_{format} + \mathcal{R}_{time} + \mathcal{R}_{acc}. \tag{7}$$

Collectively, this composite reward structure rigorously incentivizes the model to generate well-structured reasoning, actuate responses at optimal temporal junctures, and maintain factual fidelity throughout the duration of the streaming interaction.

4.3 High-Efficiency Streaming Inference

The online rollout phase of RLVR requires high-throughput streaming inference, where the model repeatedly performs chunk-level decoding while dynamically updating the KV cache. In particular, the RCSM memory mechanism requires pruning stale visual tokens during decoding, which demands explicit control over KV cache manipulation.

Existing frameworks are not well suited for this setting. Native `transformers` implementations allow manual KV cache operations but suffer from significant kernel launch overhead, while optimized engines such as `vLLM` and `SGLang` achieve high decoding throughput but provide limited support for customized KV cache updates required by RCSM.

To address this limitation, we design a streaming inference backend based on *CUDA Graphs*, which is illustrated in Fig. 2(c). The execution pipeline consists of two stages: **(1) Prefill Phase:** Executed in eager mode to process newly arrived visual tokens and update the KV cache. **(2) Decode-and-Prune Phase:** Captured as a CUDA graph that performs token decoding together with KV cache pruning in replayable execution graphs. More implementation details can be found in the Appendix.

5 ThinkStream Dataset

To train the model for real-time streaming reasoning and interaction, we construct a large-scale, high-quality dataset featuring time-grounded CoT and verifiable question-answering pairs. Our data generation pipeline consists of three sequential stages.

Video Segmentation and Dense Captioning. We employ PySceneDetect to segment continuous videos into coherent temporal scenes. We then utilize Qwen3-VL-235B-A22B-Instruct [1] to densely caption each segment, yielding a sequence of timestamped semantic descriptions that serve as the foundational factual grounding for all subsequent generation phases.

Diverse Instruction Synthesis. We construct highly diverse instruction data by exploring three key scenario dimensions: interaction modes (*Real-time Dialogue*, *Event Trigger*, *Continuous Output*), temporal scopes (*Past*, *Current*, *Future*), and seven fine-grained content semantics (including *entity attributes*, *action semantics*, *spatial relationships*, *causal reasoning*, *procedural states*, *global scenes*, and *optical character recognition*). By computing the Cartesian product of these dimensions and filtering for practical validity, we derive *39* meaningful scenario combinations. Guided by these filtered combinations, we synthesize diverse question-answering pairs that span multiple formats, specifically open-ended, multiple-choice, binary (Yes/No), and counting questions.

Time-Grounded CoT Generation. Leveraging the timestamped captions and the generated instruction pairs, we simulate the internal reasoning traces to synthesize a dense, continuous, and strictly time-grounded CoT. This ensures that every thought and response is causally constrained to its specific timestamp, effectively acting as an uninterrupted *stream of consciousness*.

Through this unified pipeline, we ultimately construct 110K Cold Start instances paired with detailed reasoning traces, alongside 9K RLVR instances strictly formatted for verifiable reward optimization. Comprehensive details regarding the video source, dataset distributions and prompt templates are provided in the Appendix.

6 Experiments

To evaluate the effectiveness of the proposed ThinkStream framework, we conduct extensive experiments across streaming and offline video benchmarks. We also provide detailed ablation studies and efficiency profiling to validate our architectural designs and the real-time capabilities of our custom streaming inference engine.

Implementation Details. We initialize our framework based on the Qwen2.5-VL-3B [2] model. During the Cold Start phase, we train the model with a batch size of 64 and a learning rate of 1×10^{-5}. In the subsequent Reinforcement Learning (RL) phase, we employ a batch size of 8, a GRPO group size of 8, and a learning rate of 2×10^{-7}. The AdamW [17] optimizer is utilized across all training stages. For video processing, frames are sampled at a rate of 2 FPS and encoded using native dynamic resolution. All training experiments are conducted on a single node equipped with 8 NVIDIA H20 GPUs.

Table 1: Performance comparison of ThinkStream and existing open-source offline and online models on the OVO-Bench streaming video benchmark. ThinkStream-3B achieves a strong average score, significantly surpassing its base model and competing online models.

Model	Real-Time Visual Perception							Backward Tracing				Avg.
	OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.	
Open-source Offline Models												
LLaVA-Video-7B [41]	69.13	58.72	68.83	**49.44**	**74.26**	59.78	63.52	56.23	57.43	7.53	40.4	51.96
Qwen2-VL-7B [26]	60.4	50.46	56.03	47.19	66.34	55.43	55.98	47.81	35.48	56.08	46.46	51.22
LongVU-7B [21]	53.69	53.21	62.93	47.75	68.32	59.78	57.61	40.74	59.46	4.84	35.01	46.31
Qwen2.5-VL-3B [2]	76.51	44.03	67.24	42.13	68.31	61.96	60.03	50.50	53.38	22.04	41.98	51.00
Qwen2.5-VL-7B [2]	67.79	55.05	67.24	42.13	66.34	60.87	59.90	51.52	58.78	23.66	44.65	52.28
Qwen2.5-VL-32B [2]	77.18	58.72	68.10	50.56	74.26	57.61	64.40	58.59	62.84	29.57	50.33	57.37
Open-source Online Models												
Flash-VStream-7B [37]	24.16	29.36	28.45	33.71	25.74	28.8	28.37	39.06	37.16	5.91	27.38	27.88
VideoLLM-online-8B [3]	8.05	23.85	12.07	14.04	45.54	21.2	20.79	22.22	18.8	12.18	17.73	19.26
Dispider-7B [19]	57.72	49.54	62.07	44.94	61.39	51.63	54.55	48.48	55.41	4.3	36.06	45.31
StreamForest-7B [36]	68.46	53.21	**71.55**	47.75	65.35	60.87	61.20	**58.92**	**64.86**	32.26	52.02	56.61
Streamo-3B [31]	78.52	52.29	67.24	44.38	55.45	**71.20**	61.51	51.18	57.43	16.67	41.76	51.64
ThinkStream-3B (Ours)	**85.23**	**64.22**	69.82	**49.43**	**69.31**	64.13	**67.03**	53.87	59.46	**43.55**	**52.30**	**59.66**

6.1 Main Results

Streaming Video Benchmarks. We comprehensively evaluate ThinkStream on specialized streaming video benchmarks [15, 18]. As shown in Tab. 1 and Tab. 2, our models consistently outperform existing models. Specifically, in Tab. 1, ThinkStream-3B achieves an overall average score of 59.66, significantly surpassing both its base model Qwen2.5-VL-3B (51.00) and competing open-source online models such as Streamo-3B (51.64) on OVO-Bench. Furthermore, on the StreamingBench Real-Time detailed in Tab. 2, ThinkStream-3B attains an average score of 75.00. This not only vastly exceeds other open-source online MLLMs like Dispider-7B (67.63) but also demonstrates highly competitive performance against proprietary models such as GPT-4o (73.28). The results demonstrate that ThinkStream achieves strong performance on streaming video tasks, showing the effectiveness of our Watch-Think-Speak loop.

Offline Video Benchmarks. To ensure the general ability of our streaming-optimized architecture, we also evaluate on standard offline video benchmarks [8, 29]. Despite aggressively evicting visual tokens, ThinkStream-3B achieves highly competitive performance as demonstrated in Tab. 3. Specifically, our model achieves a score of 61.9 on VideoMME and 56.4 on Long VideoBench. With an overall average score of 59.4 across the evaluated metrics, ThinkStream-3B significantly outperforms its base model Qwen2.5-VL-3B (with a 54.4 average). This proves that our ThinkStream effectively preserves understanding capabilities on offline video tasks.

Table 2: Performance comparison of various MLLMs on the StreamingBench Real-Time benchmark. ThinkStream-3B demonstrates highly competitive performance against proprietary models such as GPT-4o and vastly exceeds other open-source online MLLMs.

Model	OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	Avg.
Human	89.47	92.00	93.60	91.47	95.65	92.52	88.00	88.75	89.74	91.30	91.46
Proprietary MLLMs											
Gemini 1.5 pro [9]	79.02	80.47	83.54	79.67	80.00	84.74	77.78	64.23	71.95	48.70	75.69
GPT-4o [10]	77.11	80.47	83.91	76.47	70.19	83.80	66.67	62.19	69.12	49.22	73.28
Claude 3.5 Sonnet	73.33	80.47	84.09	82.02	75.39	79.53	61.11	61.79	69.32	43.09	72.44
Open-source Offline MLLMs											
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
LongVA-7B [40]	70.03	63.28	61.20	70.92	62.73	59.50	61.11	53.66	54.67	34.72	59.96
LLaVA-NeXT-Video-32B [16]	78.20	70.31	73.82	76.80	63.35	69.78	57.41	56.10	64.31	38.86	66.96
Qwen2.5-VL-3B [2]	67.21	68.75	75.39	79.17	72.96	72.27	71.30	61.38	71.59	26.06	67.96
Qwen2.5-VL-7B [2]	77.93	76.56	78.55	80.86	76.73	76.95	80.56	65.45	65.72	52.85	73.31
Qwen2.5-VL-32B [2]	76.29	79.69	78.55	83.50	76.10	79.44	80.56	61.38	68.27	59.07	74.27
Open-source Online MLLMs											
Flash-VStream-7B [37]	25.89	43.57	24.91	23.87	27.33	13.08	18.52	25.20	23.87	48.70	23.23
VideoLLM-online-8B [3]	39.07	40.06	34.49	31.05	45.96	32.40	31.48	34.16	42.49	27.89	35.99
Dispider-7B [19]	74.92	**75.53**	74.10	73.08	74.44	59.92	76.14	62.91	62.16	**45.80**	67.63
ThinkStream-3B (Ours)	**83.74**	70.31	**76.03**	**82.69**	**76.73**	**78.19**	**76.85**	**70.73**	**74.43**	45.21	**75.00**

6.2 Ablation Study

Reasoning Token Budget. We ablate the maximum reasoning token budget allocated per second of video. As illustrated in Tab. 4, increasing the reasoning token budget up to 20 tokens per second yields substantial performance improvements, as the model gains sufficient capacity for logical deduction. Specifically, scaling the budget from 0 to 20 tokens significantly improves the OVO-Backward score from 41.8 to 52.3. However, allocating beyond 20 tokens per second results in marginal performance gains (the OVO-Backward score only reaches 52.6 at 30 tokens) while inflating computational overhead and decoding latency, which jumps from 380 ms at 20 tokens to 505 ms at 30 tokens. Thus, 20 tokens per second strikes the optimal balance between reasoning capacity and efficiency.

Video KV Cache Window Size. We investigate the impact of the dense visual token retention window, which is shown in Tab. 5. Setting the window size to 20 seconds achieves the best performance, peaking at 75.0 on StreamingBench Real-Time and 52.3 on OVO-Backward. By contrast, narrower windows such as 5s and 10s result in lower OVO-Backward scores of 46.7 and 50.1, respectively, whereas expanding the window to 30s slightly degrades the OVO-Backward performance to 51.6. This confirms the validity of our assumption that short-term, high-resolution visual context combined with long-term semantic reasoning tokens is sufficient for continuous video understanding.

Table 3: Model performance comparison on standard offline video benchmarks, including VideoMME and Long VideoBench. Despite aggressively evicting visual tokens, ThinkStream-3B achieves highly competitive performance, demonstrating that it effectively preserves understanding capabilities on offline tasks.

Model	OVO Real-Time	OVO Backward	VideoMME	LongVideoBench	Avg.
Flash-VStream-7B	28.4	27.4	61.2	-	-
VideoLLM-online-8B	20.8	17.7	26.9	-	-
Dispider-7B	54.6	36.1	57.2	-	-
Streamo-3B	61.5	41.8	61.8	56.2	55.3
Qwen2.5-VL-3B	60.0	41.9	61.5	54.2	54.4
ThinkStream-3B (Ours)	**67.0** (↑7.0)	**52.3** (↑10.4)	**61.9** (↑0.8)	**56.4** (↑1.8)	**59.4** (↑5.0)

Table 4: Ablation on Reasoning Token Budget. Allocating 20 tokens per second strikes the optimal balance between reasoning capacity and efficiency.

#Token	Streaming.	OVO-BW	Latency (ms) ↓
0	69.6	41.8	**130**
5	70.2	46.9	193
10	72.3	49.7	255
20	**75.0**	52.3	380
30	**75.0**	**52.6**	505

Table 5: Ablation on Visual KV Cache Window Size. Setting the window size to 20 seconds achieves the best trade-off between accuracy and efficiency.

Window	Streaming.	OVO-RT	OVO-BW
5s	73.9	65.2	46.7
10s	74.5	66.3	50.1
20s	**75.0**	**67.0**	**52.3**
30s	74.9	66.8	51.6

Stream Memory Representation and RLVR Optimization. To validate the necessity of our RLVR pipeline and the specific memory representation, we compare several variants: (1) no memory, (2) discrete caption tokens as memory, (3) Cold-start CoT memory, and (4) our RLVR-optimized CoT memory. As is shown in Tab. 6, initializing the model with constructed cold-start CoT data yields a significant gain, improving the average score from 56.9 (no memory) to 60.5. Interestingly, utilizing discrete caption tokens actually diminishes the overall capability, dropping the average score to 48.7. Finally, post-training the model with RLVR further boosts average performance by 4.3 points (reaching 64.8), demonstrating that reasoning tokens learn to act as a far superior, highly compressed long-term memory compared to naive discrete captions.

6.3 Efficiency and Real-Time Analysis

A core contribution of our work is ensuring that the Streaming Watch-Think-Speak paradigm strictly adheres to real-time constraints. We profile our CUDA Graph-based Streaming Inference Engine against standard eager transformers implementations.

As shown in Fig. 3, our custom engine achieves a massive speedup in token decoding across various batch sizes compared to the Qwen2.5-VL-3B model. For instance, at a batch size of 1, our engine delivers 154.07 tokens/s compared to the baseline's 30.06 tokens/s, representing a more than 5× speedup. Furthermore,

Table 6: Ablation on Stream Memory Representation and RLVR Optimization. Post-training with RLVR demonstrates that reasoning tokens act as a highly compressed long-term memory representation.

Memory Variant	Streaming.	OVO-BW	OVO-RT	Avg.
Qwen2.5-VL-3B	68.9	41.9	60.0	56.9
Discrete caption as memory	59.0	36.9	50.1	48.7
No memory	69.6	41.8	59.2	56.9
Cold-start CoT memory	70.6	47.6	63.3	60.5
RLVR-optimized CoT memory	**75.0**	**52.3**	**67.0**	**64.8**

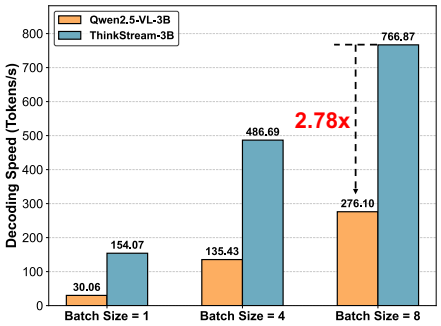

Fig. 3: Token decoding speed comparison across different batch sizes. The custom CUDA Graph-based streaming inference engine achieves a massive speedup compared to the standard Qwen2.5-VL-3B baseline, maintaining high throughput while preserving flexible KV cache control.

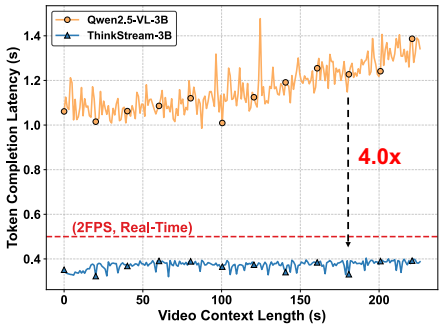

Fig. 4: Real-time latency scaling with processed video length. ThinkStream successfully bounds the end-to-end inference latency below the 0.5s real-time threshold (required for 2 FPS inputs) as the video context grows, whereas the baseline model scales poorly and consistently violates the threshold.

at a batch size of 8, our decoding speed scales efficiently to 766.87 tokens/s, significantly outperforming the baseline's 276.10 tokens/s. This confirms that our custom backend maintains high throughput while preserving the flexible, manual KV cache control required for our eviction strategy.

Crucially, as illustrated in Fig. 4, our framework successfully bounds latency as the processed video length increases. While the baseline model's latency scales poorly and consistently violates the real-time threshold—fluctuating between 1.0s and 1.4s per second of video—our combination of algorithmic KV cache eviction and engineering optimizations ensures that the end-to-end inference latency remains flat. It consistently stays below the 0.5s real-time threshold required for 2 FPS inputs. The total processing delay remains bounded, proving that our approach can be deployed efficiently in streaming video environments.

7 Conclusion

We study streaming video reasoning, where models must continuously interpret incoming observations under strict latency and memory constraints. We introduce the Watch–Think–Speak paradigm and propose ThinkStream, a framework that enables incremental reasoning over long video streams. Through Reasoning-Compressed Streaming Memory (RCSM) and streaming reinforcement learning with verifiable rewards, our approach maintains bounded memory usage while preserving coherent long-horizon understanding. Experiments show that ThinkStream achieves strong performance on streaming video benchmarks with low latency while retaining competitive capability on standard offline video tasks.

Acknowledgements

This research is supported by Artificial Intelligence-National Science and Technology Major Project (2023ZD0121200), and the National Natural Science Foundation of China (62437001, 62436001,62531026), and the Natural Science Foundation of Jiangsu Province under Grant BK20243051,and the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDB1350103.

References

1. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
2. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
3. 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18407–18418 (2024)
4. Chen, Y., Bai, X., Wang, Z., Bai, C., Dai, Y., Lu, M., Zhang, S.: Streamkv: Streaming video question-answering with segment-based kv cache retrieval and compression. arXiv preprint arXiv:2511.07278 (2025)
5. 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: The Thirty-ninth Annual Conference on Neural Information Processing Systems
6. Ding, X., Wu, H., Yang, Y., Jiang, S., Zhang, Q., Bai, D., Chen, Z., Cao, T.: Streammind: Unlocking full frame rate streaming video dialogue through event-gated cognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13448–13459 (2025)
7. 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. arXiv preprint arXiv:2503.21776 (2025)

8. 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: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 24108–24118 (2025)
9. Gemini, G.T.: 1.5: unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530 (2024)
10. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)
11. Kim, J., Kim, M.S., Chung, J., Cho, J., Kim, J., Kim, S., Sim, G., Yu, Y.: Egospeak: learning when to speak for egocentric conversational agents in the wild. In: Findings of the Association for Computational Linguistics: NAACL 2025. pp. 2990–3005 (2025)
12. Li, H., Zhang, Y., Guo, L., Yue, X., Liu, J.: Breaking the encoder barrier for seamless video-language understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 23167–23176 (2025)
13. Li, W., Hu, B., Shao, R., Shen, L., Nie, L.: Lion-fs: Fast & slow video-language thinker as online video assistant. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3240–3251 (2025)
14. Lin, J., Yin, H., Ping, W., Molchanov, P., Shoeybi, M., Han, S.: Vila: On pretraining for visual language models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 26689–26699 (2024)
15. Lin, J., Fang, Z., Chen, C., Wan, Z., Luo, F., Li, P., Liu, Y., Sun, M.: Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. arXiv preprint arXiv:2411.03628 (2024)
16. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llavanext: Improved reasoning, ocr, and world knowledge (2024)
17. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
18. 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: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 18902–18913 (2025)
19. 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: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 24045–24055 (2025)
20. Qian, R., Dong, X., Zhang, P., Zang, Y., Ding, S., Lin, D., Wang, J.: Streaming long video understanding with large language models. Advances in Neural Information Processing Systems **37**, 119336–119360 (2024)
21. 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. arXiv preprint arXiv:2410.17434 (2024)
22. Tang, C., Han, Z., Sun, H., Zhou, S., Zhang, X., Wei, X., Yuan, Y., Xu, J., Sun, H.: Tspo: Temporal sampling policy optimization for long-form video language understanding. arXiv preprint arXiv:2508.04369 (2025)
23. Tian, S., Wang, R., Guo, H., Wu, P., Dong, Y., Wang, X., Yang, J., Zhang, H., Zhu, H., Liu, Z.: Ego-r1: Chain-of-tool-thought for ultra-long egocentric video reasoning. arXiv preprint arXiv:2506.13654 (2025)
24. Tong, J., Fan, Y., Zhao, A., Ma, Y., Shen, X.: Streamingthinker: Large language models can think while reading. arXiv preprint arXiv:2510.17238 (2025)

25. Wang, H., Feng, B., Lai, Z., Xu, M., Li, S., Ge, W., Dehghan, A., Cao, M., Huang, P.: Streambridge: Turning your offline video large language model into a proactive streaming assistant. arXiv preprint arXiv:2505.05467 (2025)
26. 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)
27. Wang, S., Jin, J., Wang, X., Song, L., Fu, R., Wang, H., Ge, Z., Lu, Y., Cheng, X.: Video-thinker: Sparking" thinking with videos" via reinforcement learning. arXiv preprint arXiv:2510.23473 (2025)
28. Wang, Z., Yoon, J., Yu, S., Islam, M.M., Bertasius, G., Bansal, M.: Video-rts: Rethinking reinforcement learning and test-time scaling for efficient and enhanced video reasoning. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 28114–28128 (2025)
29. Wu, H., Li, D., Chen, B., Li, J.: Longvideobench: A benchmark for long-context interleaved video-language understanding. Advances in Neural Information Processing Systems **37**, 28828–28857 (2024)
30. Wu, T., Yang, L., Zhan, G., Zhang, Y., Liao, Y., Li, J., Fu, D., Zhang, L., Wang, L.: Tempr1: Improving temporal understanding of mllms via temporal-aware multi-task reinforcement learning. arXiv preprint arXiv:2512.03963 (2025)
31. Xia, J., Chen, P., Zhang, M., Sun, X., Zhou, K.: Streaming video instruction tuning. arXiv preprint arXiv:2512.21334 (2025)
32. Xu, R., Xiao, G., Chen, Y., He, L., Peng, K., Lu, Y., Han, S.: Streamingvlm: Real-time understanding for infinite video streams. arXiv preprint arXiv:2510.09608 (2025)
33. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al.: Qwen3 technical report. arXiv preprint arXiv:2505.09388 (2025)
34. Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., Qiu, Z.: Qwen2.5 technical report. arXiv preprint arXiv:2412.15115 (2024)
35. 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)
36. 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. arXiv preprint arXiv:2509.24871 (2025)
37. Zhang, H., Wang, Y., Tang, Y., Liu, Y., Feng, J., Jin, X.: Flash-vstream: Efficient real-time understanding for long video streams. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 21059–21069 (2025)
38. Zhang, H., Yang, S., Fu, J., Ng, S.K., Qiu, X.: Hermes: Kv cache as hierarchical memory for efficient streaming video understanding. arXiv preprint arXiv:2601.14724 (2026)
39. Zhang, J., Zhu, Y., Sun, M., Luo, Y., Qiao, S., Du, L., Zheng, D., Chen, H., Zhang, N.: Lightthinker: Thinking step-by-step compression. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 13318–13339 (2025)
40. 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. arXiv preprint arXiv:2406.16852 (2024)

41. Zhang, Y., Wu, J., Li, W., Li, B., Ma, Z., Liu, Z., Li, C.: Llava-video: Video instruction tuning with synthetic data. arXiv preprint arXiv:2410.02713 (2024)
42. Zhao, Z., Wang, K., Li, S., Qian, R., Lin, W., Liu, H.: Cogstream: Context-guided streaming video question answering. arXiv preprint arXiv:2506.10516 (2025)