Harnessing Streaming Video in the Wild

Dingyu Yao[1,2*] **Shuhuan Gu**[1,2*] **Qingyi Si**[3*] **Junhao Zhou**[1,2*] **Chenxu Yang**[1,2] **Chuanyu Qin**[1,2]
Naibin Gu[1,2] **Zheng Lin**[1,2†] **Weiping Wang**[1] **Nan Duan**[3] **Jiaqi Wang**[3]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, Chin,
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]JD.COM
{yaodingyu, linzheng}@iie.ac.cn, siqingyi.phoebus@jd.com

Abstract

Vision-Language Models (VLMs) are increasingly required to process unbounded video streams in applications such as video-call assistants, live commentary, and embodied robots. An ideal streaming system should support proactive interaction, long-horizon memory, and real-time processing, while resting on a VLM backbone capable of handling diverse in-the-wild streaming tasks. However, existing VLMs excel at offline video understanding but fall short in streaming capabilities and lack dedicated infrastructure for streaming deployment. We address this gap on three fronts. (i) For backbone capability, we construct **Streaming-Train-248K**, a streaming dataset paired with a novel training objective for adapting VLMs to streaming interaction and understanding. (ii) For real-world deployment, we introduce **Streaming Harness**, a plug-and-play system that endows any VLM with three core abilities: proactive interaction (per-second response decisions), long-term memory (12-hour context retention), and real-time processing (sub-second latency). (iii) To drive community progress on real-streaming capabilities, we design **Streaming-Eval**, a benchmark that reflects models' capabilities across diverse in-the-wild scenarios. Extensive experiments demonstrate consistent gains across all four streaming capabilities. We will open-source our data, harness, and benchmark to accelerate the community's transition from offline video understanding to deployable streaming intelligence. [1]

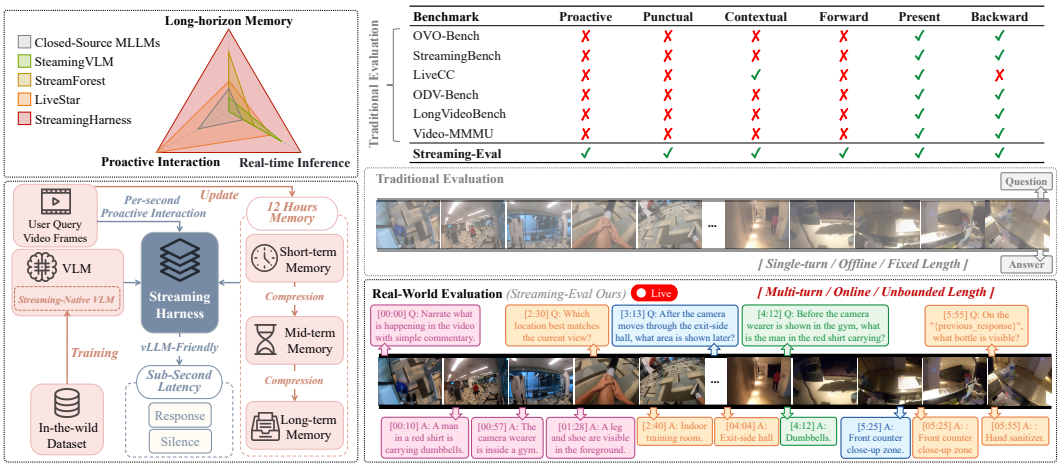
1 Introduction

Vision-Language Models (VLMs) [1–4] have substantially advanced video understanding, achieving strong results on offline tasks [5–8]. These tasks share a common formulation: the model jointly attends over a complete, pre-recorded clip and produces a one-shot response. Yet many real-world applications violate this assumption. In video-call assistants [4], live commentary [9, 10], and embodied robots [11], video arrives as an unbounded stream, and the model is required to respond causally, conditioning solely on previously observed frames under stringent latency constraints.

An ideal streaming system should satisfy four fundamental requirements: *(i) a versatile VLM backbone* capable of handling diverse in-the-wild streaming tasks; *(ii) proactive interaction*, whereby the model determines when to respond and when to remain silent, generating outputs causally without access to future frames; *(iii) long-horizon memory*, which enables the retrieval of historical evidence frames pertinent to a given query across streams of unbounded length; and *(iv) real-time inference*, ensuring that responses are produced under low-latency constraints and remain temporally

*Equal contribution.
†Corresponding author
[1]Code is available at: https://github.com/ydyhello/StreamingHarness

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



	Benchmark	Proactive	Punctual	Contextual	Forward	Present	Backward
Traditional Evaluation	OVO-Bench	✗	✗	✗	✗	✓	✓
	StreamingBench	✗	✗	✗	✗	✓	✓
	LiveCC	✗	✗	✓	✗	✓	✗
	ODV-Bench	✗	✗	✗	✗	✓	✓
	LongVideoBench	✗	✗	✗	✗	✓	✓
	Video-MMMU	✗	✗	✗	✗	✓	✓
	Streaming-Eval	✓	✓	✓	✓	✓	✓

Figure 1: **Top-left:** StreamingHarness jointly delivers *proactive interaction*, *long-horizon memory*, and *real-time inference*—capabilities that no prior baseline satisfies simultaneously. **Bottom-left:** Overview of the StreamingHarness. **Top-right:** Comparative overview of Streaming-Eval and existing benchmarks. **Bottom-right:** Unlike single-turn offline benchmarks, Streaming-Eval assesses six in-the-wild capabilities (*Proactive*, *Punctual*, *Contextual*, *Forward*, *Present*, *Backward*).

synchronized with the incoming input. A growing body of work has sought to advance streaming video understanding along these axes [12–21]; however, as illustrated in Figure 1 (top-left), most existing methods address only one or two of them in isolation, rather than jointly satisfying all of them within a unified framework. To close this gap, we present a unified stack across three fronts (model, harness and benchmark) toward an ideal streaming system.

For backbone capability, we train a Streaming-Native VLM for proactive interaction, capable of handling diverse ***in-the-wild streaming tasks***. We introduce two special tokens [17, 20], `</response>` and `</silence>`, which enable the model to decide whether to respond based on the content observed so far. To support this, we carefully construct **Streaming-Train-248K**, a high-quality dataset spanning a rich variety of real-world task types. The dataset provides per-second frame-text alignment, pairing each one-second segment with either an explicit response or a silence token. Together with a tailored loss function that mitigates the silence-response imbalance, we achieve more stable training.

For real-world deployment, we propose **StreamingHarness**—a plug-and-play system that instantly equips any VLM with strong instruction-following capabilities for streaming video processing. Concretely, as illustrated in Figure 1 (bottom-left), the harness manages ***long-horizon memory*** through a three-tier memory management module, supporting up to 12 hours of context. In parallel, a prefix-aware Key-Value (KV) cache design makes the harness compatible with native `vLLM`, allowing KV cache reuse to preserve sub-second latency and thus deliver ***real-time processing***. Furthermore, to support ***proactive interaction***, user queries may be issued at any moment, and the harness invokes the backbone every second, letting the model decide whether to respond at each tick. The dedicated system prompt can be extended to any VLM that is not natively trained for streaming tasks.

Model capability and deployment infrastructure are necessary, yet without a principled way to measure whether they jointly meet genuine streaming requirements, progress is hard to track. As shown in Figure 1 (top-right), existing benchmarks [14, 22–28] reduce evaluation to single-turn, offline-style queries over short clips, overlooking core streaming behaviors, such as recalling distant evidence, deferring responses pending future evidence, and maintaining coherence across the stream. To this end, we design **Streaming-Eval**, a benchmark spanning two in-the-wild axes: Streaming Interaction, which evaluates whether (Proactive), when (Punctual), and how coherently across interdependent turns (Contextual) a model responds; and Streaming Understanding, which evaluates reasoning over past (Backward), ongoing (Present), and future (Forward) visual evidence. We further propose SW-F1 (Streaming Weighted F1), a metric that jointly captures answer correctness and response timeliness.

We evaluate our stack on Streaming-Eval, spanning narration and question answering. Our training (Section 3) transforms an offline VLM into a streaming-native model, yielding substantial gains

in proactive interaction. Equipping off-the-shelf VLMs with StreamingHarness (Section 4) further endows them with long-horizon memory, particularly improving performance on backward tasks. Combining our streaming-native 8B VLM with StreamingHarness achieves state-of-the-art results, surpassing most strong closed-source models. Moreover, thanks to our vLLM-friendly design, our system sustains coherent narration over two-hour live sports broadcasts at stable sub-second latency. Below are the key contributions of our work:

- **Streaming-Train-248K**. We construct a large-scale streaming dataset with per-second frame-text alignment, paired with a tailored training objective, to adapt VLMs to *in-the-wild* streaming interaction and yield streaming-native VLMs.
- **StreamingHarness**. We introduce a plug-and-play framework that equips any VLM with three core streaming abilities: *proactive interaction* (per-second response decisions), *long-horizon memory* (12-hour context), and *real-time inference* (sub-second latency).
- **Streaming-Eval**. We design a benchmark targeting in-the-wild streaming tasks, which captures the capabilities required for real-world deployment, encouraging the community to shift from offline video understanding toward deployable streaming intelligence.
- **Excellent Performance**. Experiments show that our 8B streaming-native model equipped with StreamingHarness achieves state-of-the-art results, significantly outperforming Claude Opus 4.6, GPT 5.4, Gemini 3.1 Pro, and Doubao Seed 2.0 Pro.

2 Problem Formulation: Streaming as Multi-turn Online Interaction

We formulate the streaming setting as follows. A video is revealed sequentially as a sequence of temporal segments $\mathcal{V} = \{v_1, v_2, \ldots, v_T\}$, where each segment v_t spans a brief fixed-length interval (e.g., one second) and is anchored by a timestamp token τ_t of the form `<ts-(t+1)s>`. The user may issue textual queries at sparsely distributed moments along the timeline; we denote by q_t the query received at step t, with $q_t = \varnothing$ when no query is present. At each step, the model emits a_t as either (i) an explicit utterance `</response> answer` when accumulated evidence suffices to address a pending query or narrate a salient event, or (ii) a silence token `</silence>` otherwise. This formulation promotes temporally grounded responses, suppresses premature predictions, and mitigates hallucinated outputs. Each step thus constitutes an interaction unit $u_t = (\tau_t, v_t, q_t, a_t)$, and the dialogue history is $\mathcal{H}_{t-1} = \big(u_1, u_2, \ldots, u_{t-1}\big)$. Conditioned on this history, the response at step t is generated as $a_t = f_\theta(\mathcal{H}_{t-1}, \tau_t, v_t, q_t)$. In contrast to offline video understanding that consumes the entire video a priori, streaming video understanding has no access to future segments $\{v_{t+1}, \ldots, v_T\}$ and must respond causally based on previously observed frames. Through this multi-turn online interaction, the model determines whether to speak at each step t.

3 Streaming-Native VLM in the wild

As analyzed in Section 2, streaming video inference is characterized by multi-turn interaction, where the model decides at each step whether to respond or remain silent. To align with this inference-time schedule, we train a Streaming-Native VLM for proactive interaction by interleaving vision and text tokens during training, and curate a 248K dataset covering diverse proactive perception scenarios.

Data Statistics We integrate multiple open-source video datasets as sources, including EpicKitchens [29], YouCook2 [30], Ego4D [31], EgoBlind [32], EgoExo4D [33], EgoExoLearn [34], DiDeMo [35], Charades [36], OmniStar [15], QVHighlights [37], ActivityNet [38], Shot2Story [39], Assembly101 [40], HoloAssist [41], WTaG [42], MovieChat [43], QueryD [44], and Live-WhisperX [9]. To convert the raw dataset into the standard multi-turn online interaction format (see Section 2), we perform per-second frame-text alignment, in which each one-second segment is paired with either an explicit response or a silence

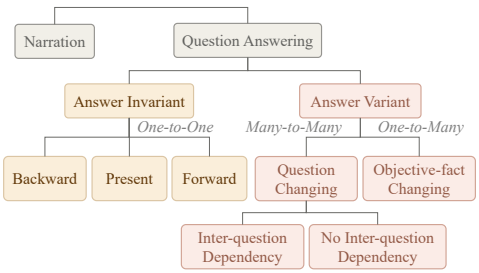


Figure 2: Overview of our task taxonomy.

token and tagged with a timestamp (e.g., <0s-1s>).
In addition, we conduct source-level inspection and discard sources with uncertain annotations. This yields 248K samples spanning diverse real-world scenarios such as *daily living*, *work and productivity*, *knowledge and learning*, and *outdoor and recreation*.

Task Taxonomy Figure 2 provides an overview of our task taxonomy. The first, ***narration***, requires the model to continuously monitor the incoming video and produce coherent descriptions of salient events. We build this task from dense temporal annotations and automatic speech recognition (ASR) transcripts. Rather than captioning every frame, the model emits a description only when such events occur. The second, ***question answering***, requires the model to respond selectively conditioned on the observed video frames. We categorize QA by whether the answer varies across observations: *(1) Answer-invariant*, a one-to-one question–answer mapping; and *(2) Answer-variant*, further partitioned into *(2.1) Objective-fact changing*, a one-to-many mapping, and *(2.2) Question changing*, a many-to-many mapping. *Question changing* is further divided into *(2.2.1) Inter-question dependency* and *(2.2.2) No inter-question dependency*. Orthogonally, the evidence frames may be situated *backward*, *present*, or *forward* relative to the query.

Training Objective In streaming scenarios, silence runs are substantially more frequent than response segments, so the token distribution over assistant positions is heavily skewed toward </silence>. Under standard supervised fine-tuning, this imbalance biases the gradient toward reinforcing continued silence rather than the transition from silence to response, diluting the response signal. To address this, we assign distinct weights to tokens based on their role. Let $\mathcal{A}$ denote the set of supervised assistant-token positions, and let $\mathcal{C} = \{c_i\}_{i=1}^{M} \subseteq \mathcal{A}$ denote the ordered subsequence of control-token positions, where $y_{c_i} \in \{\texttt{</silence>}, \texttt{</response>}\}$. A *silence run* is a maximal consecutive subsequence of </silence> tokens within $\mathcal{C}$. For $j \in \mathcal{A}$, with the convention $y_{c_0} = \varnothing$, we assign:

$$w_j = \begin{cases} w_{\text{silence}}^{\text{first}}, & y_{c_i} = \texttt{</silence>},\ y_{c_{i-1}} \neq \texttt{</silence>}, \\ w_{\text{silence}}^{\text{repeated}}, & y_{c_i} = \texttt{</silence>},\ y_{c_{i-1}} = \texttt{</silence>}, \\ w_{\text{response}}, & y_{c_i} = \texttt{</response>}, \\ 1, & \text{otherwise.} \end{cases} \tag{1}$$

We set $w_{\text{silence}}^{\text{first}} = 1$, $w_{\text{silence}}^{\text{repeated}} < 1$ to down-weight silence continuations, and $w_{\text{response}} > 1$ to up-weight response onsets. The training objective normalizes by the number of supervised positions:

$$\mathcal{L}(\theta) = -\frac{1}{|\mathcal{A}|} \sum_{j \in \mathcal{A}} w_j \ \log p_\theta(y_j \mid y_{<j}) \,. \tag{2}$$

4 Streaming Harness

While the Streaming-Native VLM trained in Section 3 acquires the ability to emit </silence> and </response> appropriately, deploying it on open-ended streams uncovers another class of challenges that lie beyond model capability and instead stem from the inference pipeline itself. Real-world streams last for hours rather than the minutes typical of training clips, demanding a memory mechanism whose footprint does not grow without bound as stream length increases. Moreover, to keep input frames and output text synchronized, each step must also be processed within a sub-second budget to preserve the responsiveness on which proactive interaction depends.

A straightforward remedy is to maintain the KV cache via a sliding window [45, 46]; an orthogonal line of work instead prunes redundant tokens to curtail cache growth [47]. While both strategies effectively alleviate memory overhead, neither supports KV cache reuse, precluding direct integration with the prefix-caching optimizations exploited by modern inference engines [48].

To address these challenges and enable proactive interaction, we introduce StreamingHarness, a plug-and-play inference framework tailored to real-world streaming video applications. StreamingHarness supports unbounded streaming video inference with hours-long memory retention under stringent real-time latency constraints. The overall architecture is illustrated in Figure 3. Specifically, StreamingHarness comprises three components: *three-tier memory management*, *prefix-aware KV cache*, and *event-driven response triggering*.

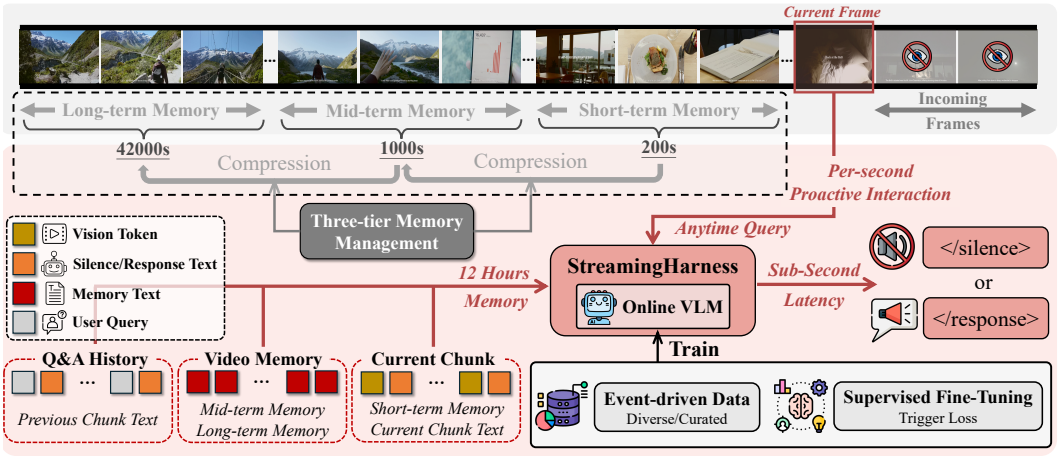


Figure 3: **Overview of StreamingHarness.** A three-tier memory management scheme compresses the streaming video into short-, mid-, and long-term memory (200 s / 1,000 s / 42,000 s), enabling up to **12 hours** of context. At each step, the Online VLM decides whether to emit `</silence>` or `</response>` with sub-second latency, enabled by our `vLLM`-friendly design.

Three-tier Memory Management To accommodate hours-long streams within a bounded memory budget, we organize the context into a three-tier hierarchy. As shown in Figure 3, the stream is partitioned into: (i) a *short-term memory* retaining the most recent T_s seconds as raw vision tokens; (ii) a *mid-term memory* holding up to M textual summaries of past short-term chunks, covering $T_m = MT_s$ seconds at moderate compression; and (iii) a *long-term memory* storing up to L aggressively compressed blocks, each consolidated from M consecutive mid-term summaries and thus spanning $T_l = LMT_s$ seconds.

Concretely, whenever the short-term buffer fills, the enclosed chunk is offloaded through a *mid-term memory agent*, which condenses T_s seconds of frames into a compact textual summary that faithfully captures per-frame semantics, salient details, and the temporal evolution of object and scene states, trading vision tokens for text tokens at substantially lower cost. Once M such summaries accumulate, a *long-term memory agent* consolidates them into a single block by merging overlapping content, removing redundancy, and organizing causally related events into a coherent timeline. The long-term tier is capped at L blocks with FIFO eviction, and both agents run asynchronously ahead of each chunk boundary, fully hiding memory consolidation behind mainline inference. The total temporal coverage is therefore bounded by $T_s + T_m + T_l = T_s\,(1 + M + LM)$, which, under our default configuration (T_s=200 s, M=5, L=42), reaches roughly 12 hours, sufficient for the vast majority of real-world streams.

Prefix-aware KV Cache Although the three-tier hierarchy already shortens the context, recomputing the entire KV cache at every step's prefill stage would still violate the sub-second latency budget. Modern inference engines [48, 49] accelerate the prefill stage via *prefix caching*, which caches the KV states of prior dialogue turns and reuses them whenever a subsequent request shares the same prefix. StreamingHarness is therefore organized around this principle.

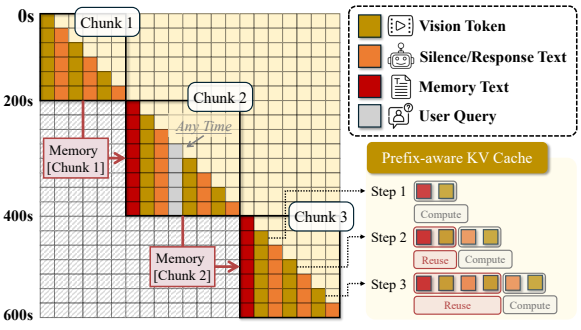


Figure 4: At each chunk, StreamingHarness reuses the KV cache.

As illustrated in Figure 4, StreamingHarness directs the engine to perform a one-time prefill of the Memory Text at the start of each chunk, converting it into the corresponding KV cache. At every subsequent step, the new frame and the previous response are appended,

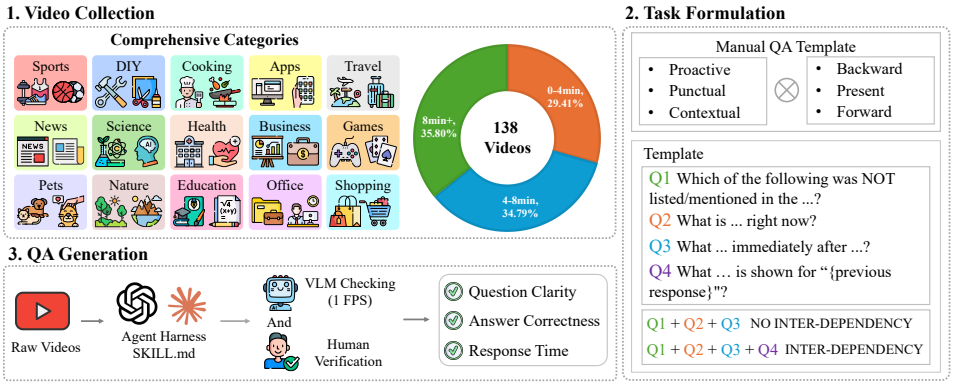


Figure 5: **Generation pipeline of Streaming-Eval.** We curate high-quality and diverse public videos, and automatically generate multiple-choice questions through an agent harness guided by a well-designed system prompt. GPT 5.4 and Claude Opus 4.6 are adopted as the VLMs for annotation.

and only their KV entries need to be computed, while all previously cached KV states, including the Memory Text and earlier turns within the chunk, are reused without recomputation. This design substantially reduces the per-step time-to-first-token (TTFT), enabling StreamingHarness to deliver near-instantaneous responses.

Event-driven Response Triggering We propose a training-free adaptation scheme that enables a base model without native proactive-interaction capability to generalize to proactive-interaction objectives. At each step, the model is conditioned on the current frame, the dialogue context of the current chunk, the past video history, and the past question-answer history. Given the active query, the model performs a judgment over this evidence: if no new and informative content is yet available, for instance when the key event has not occurred or the scene is still in transition, it remains silent and defers to the next frame; otherwise, it produces a concise response strictly grounded in the visible evidence, without fabricating unseen entities or facts. The corresponding system prompt is provided in Appendix C.1. This procedure yields a streaming inference loop of four stages: query triggering, frame-by-frame judgment, timely response, and monitoring resumption.

Together, these designs endow StreamingHarness with the long-horizon memory, real-time inference, and proactive interaction required for production-grade streaming video understanding.

5 Streaming-Eval

To facilitate community progress on real-time streaming capabilities, we introduce Streaming-Eval, a benchmark specifically designed for in-the-wild streaming video scenarios, which jointly assesses both answer correctness and response timeliness.

Benchmark Construction Figure 5 illustrates the three-stage construction pipeline of Streaming-Eval: video collection, task formulation, and QA generation. ***(i) Video collection.*** We aggregate 138 videos from PhoStream [50], Video-MMMU [28], Ego4D [31], THUMOS [51], HiREST [52], MLVU [53], RTV-Bench [54], and OVO-Bench [23], covering 15 everyday categories with a balanced split across short (0–4 min), medium (4–8 min), and long (8 min+) clips. ***(ii) Task formulation.*** To systematically evaluate streaming video comprehension, we adopt the same task taxonomy as that used for the training data, summarized in Figure 2. Each task type is constructed from a carefully designed template that pairs one mode of streaming interaction (*proactive*, *punctual*, or *contextual*) with one mode of streaming understanding (*backward*, *present*, or *forward*), thereby yielding comprehensive coverage of streaming interaction patterns. ***(iii) QA generation.*** Both questions and answers are generated by an agent harness guided by a structured SKILL, with GPT 5.4 and Claude Opus 4.6 serving as the annotation models. Each QA pair is then filtered by a VLM checker and verified by human annotators along three dimensions: question clarity, answer correctness, and response time.

Table 1: **Narration accuracy (win rate vs. baselines).** † denotes offline models adapted to our streaming harness framework. The colored bars (e.g.,) indicate the win / tie / loss rates (%) against the baseline, with the displayed number denoting the win rate. Due to API rate limits, T_s is set to 10 for closed-source models.

Method	Narration (Win Rate %)				Avg.
	vs. Claude Opus 4.6 Thinking	vs. GPT 5.4 (High)	vs. Gemini 3.1 Pro (High)	vs. Doubao Seed 2.0 Pro	
Claude Opus 4.6 Thinking †	14.5	10.5	10.0	26.0	15.3
GPT 5.4 (High) †	52.5	12.5	18.0	44.0	31.8
Gemini 3.1 Pro (High) †	**70.0**	62.0	35.5	69.0	59.1
Doubao Seed 2.0 Pro †	19.5	11.5	7.0	10.0	12.0
Qwen3-VL-8B †	4.0	4.0	2.5	4.0	3.6
Qwen3-VL-32B †	3.0	3.0	0.5	2.0	2.1
Qwen3.5-9B (Thinking) †	0.5	0.5	0.0	1.0	0.5
StreamingHarness-8B †	65.0	**65.0**	**44.0**	**71.5**	**61.4**

Evaluation Metric For narration, we sample 100 instances from the LiveSports3K-CC benchmark [9]. For each sample, we convert each candidate's narration into a timestamped transcript of the form `[`t`s] utterance`, where t denotes the time at which the utterance is emitted, and prompt Claude Sonnet 4.6 to compare the two candidates against the ground-truth Automatic Speech Recognition (ASR) transcription. The judge evaluates two criteria of equal weight: narration style (pacing and tone) and consistency with the reference. To mitigate position bias, each pair is judged twice with the candidate order swapped, and we report the win rate over the resulting $2N$ votes. The full judge prompt is provided in Appendix D.1.

For question answering, we use a rule-based scheme. Each question is annotated with a ground-truth answer window W. A non-silence prediction is a true positive (TP) only if it is emitted within W and exactly matches the ground-truth answer. Each question and each prediction can be matched at most once. Every unmatched non-silence prediction is a false positive (FP), including an incorrect prediction within W, a duplicate correct prediction, and any prediction outside W. A question with no matched correct prediction within W is a false negative (FN). Thus, an incorrect in-window answer contributes both one FP and one FN, whereas remaining silent contributes one FN and no FP. To jointly assess correctness and timeliness in the streaming setting while treating answer correctness as our primary concern, we adopt a streaming weighted F1 (SW-F1) that assigns higher weights to TP and FN:

$$\text{SW-F1} = \frac{\text{TP}}{\text{TP} + w_{\text{FP}}\,\text{FP} + \text{FN}}, \tag{3}$$

with $w_{\text{FP}} = 0.1$. Compared to accuracy-style metrics that count only TP, the FP term penalizes "always-respond" behaviors that emit answers at every step to inflate TP. Compared to vanilla F1, the lower FP weight prioritizes answer correctness, preventing over-penalization.

6 Experiment

6.1 Experimental Setup

Training. We continue training from Qwen3-VL-8B-Instruct [1] on Streaming-Train-248K with a learning rate of 1×10^{-5}. During fine-tuning, the vision encoder is kept frozen, while the connector and the LLM are fully updated. Each training video is sampled at 1 FPS, with the maximum number of pixels per frame capped at 131,072 and the maximum context length set to 48K tokens. For the token weights in Eq. 1, we set $w^{\text{first}}_{\text{silence}} = 1$, $w^{\text{repeated}}_{\text{silence}} = 0.8$, and $w_{\text{response}} = 1.5$. The full training run consumes approximately 4,096 H200 GPU-hours.

Table 2: **Evaluation results of different models on Streaming-Eval (QA).** Scores reported in the table are SW-F1. † denotes offline models adapted to our streaming harness framework. Due to API rate limits, T_s is set to 10 for closed-source models.

Method	Answer Invariant QA			Answer Variant QA			Avg.
	Backward	Present	Forward	OFC	OC w/ IQD	OC w/o IQD	
Claude Opus 4.6 Thinking †	12.5	6.1	10.3	8.6	11.1	8.4	9.5
GPT 5.4 (High) †	26.0	17.4	15.1	9.9	20.2	21.0	18.3
Gemini 3.1 Pro (High) †	17.0	8.2	16.9	11.9	14.7	14.2	13.8
Doubao Seed 2.0 Pro †	14.6	9.7	17.9	13.2	13.1	11.3	13.3
Qwen3-VL-8B †	16.2	5.7	13.7	7.0	10.2	12.0	10.8
Qwen3-VL-32B †	17.0	8.0	15.2	6.8	10.4	13.8	11.9
Qwen3.5-VL-9B (Thinking) †	0.0	0.0	0.1	0.0	0.2	0.3	0.1
StreamingHarness-8B †	**54.2**	**21.0**	**48.9**	**32.5**	**61.9**	**56.3**	**45.8**
($T_s = 10, M = 4, L = 8$)	20.6	22.3	50.1	31.2	56.7	54.7	39.3

Baselines. We evaluate two categories of models as baselines in this study. (1) Top closed-source models, including Claude Opus 4.6 Thinking, GPT 5.4 (High), Gemini 3.1 Pro (High), and Doubao Seed 2.0 Pro. (2) Open-source models, including Qwen3-VL-8B, Qwen3-VL-32B, and Qwen3.5-9B.

Inference. All baselines process videos at 1 FPS, with the maximum number of pixels per frame capped at $262{,}144$. For our three-tier memory management (Sec. 4), we adopt the default configuration $T_s = 200$ s, $M = 5$, $L = 42$ for open-source models. For closed-source models, we reduce the configuration to $T_s = 10$ s, $M = 4$, $L = 8$ due to API rate limits. The mid-term memory agent is capped at $262{,}144$ input pixels per frame and $4{,}000$ output tokens, while the long-term memory agent is capped at $2{,}000$ output tokens.

6.2 Accuracy Evaluation

Narration We evaluate narration accuracy by reporting pairwise win rates against four closed-source VLMs, as shown in Table 1. We find that both closed-source and open-source baselines over-narrate in a dense captioning style, failing to produce natural commentary that speaks only at key moments. Smaller open-source models perform particularly poorly, as they frequently hallucinate, misreport critical outcomes, repeat templated phrases, and miss key events. In contrast, our 8B streaming-native VLM achieves a 61.4% average win rate, reaching 71.5% against Doubao Seed 2.0 Pro and 65.0% against both Claude Opus 4.6 Thinking and GPT 5.4 (High). The reason is that large-scale training on in-the-wild tasks, paired with a tailored loss function (Section 3), gives the model proactive interaction capabilities and lets it produce output that reads like live broadcast commentary (see Figure 8). We further observe that Claude Opus 4.6 Thinking† and Gemini 3.1 Pro† (High) achieve higher win rates than loss rates against Claude Opus 4.6 Thinking and Gemini 3.1 Pro (High), respectively. The improvement stems from our three-tier memory management (Section 4), which retains context such as player names and scores to anchor subsequent commentary and mitigate hallucination (see Figure 9).

Question Answering As shown in Table 2, we conduct a systematic comparison between StreamingHarness-8B and the baselines on QA tasks. The results demonstrate that StreamingHarness-8B achieves significant improvements over all current VLMs across all streaming QA tasks. Specifically, our harness (Section 4) retrieves past evidence from the memory text and preserves Q&A history across turns, yielding large gains on Backward and Inter-question Dependency (see Figure 11). Meanwhile, our 8B streaming-native VLM (Section 3) enables stronger proactive interaction by deferring responses until future evidence becomes available (see figure 10), substantially improving Forward and Objective-fact Changing.

With the memory configuration matched to closed-source baselines ($T_s = 10$, $M = 4$, $L = 8$), StreamingHarness-8B achieves an average SW-F1 of 39.3, compared with 18.3 for the strongest closed-source baseline. Relative to its default configuration, Backward QA drops from 54.2 to 20.6, showing the importance of a longer memory horizon for historical recall.

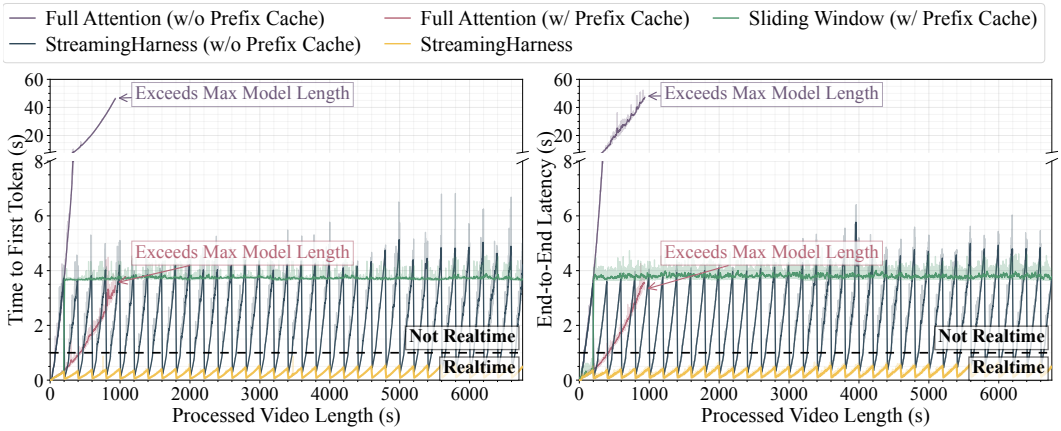


Figure 6: Time to First Token (left) and End-to-End Latency (right) vs. processed video length (up to 2 h). StreamingHarness keeps latency low and stable, below the real-time threshold.

6.3 Efficiency Evaluation

As shown in Figure 6, we evaluate TTFT and End-to-End latency for five inference strategies on a 2-hour sports broadcast video [45]: full attention (w/o prefix cache), full attention (w/ prefix cache), sliding window (w/ prefix cache), StreamingHarness (w/o prefix cache), and our full StreamingHarness. All measurements are conducted at 1 FPS on a single NVIDIA H200. The real-time threshold is set to 1 s, matching the inter-frame interval and ensuring that each response is generated before the next input arrives.

Full attention saturates the model's maximum context length within a few hundred seconds, regardless of prefix caching. Sliding window prevents context overflow and effectively alleviates memory overhead, but its latency stabilizes at approximately 4 s, substantially exceeding the real-time threshold; since consecutive steps share no common prompt prefix, `vLLM`'s [48] prefix caching mechanism provides no benefit under this scheme. In contrast, our full StreamingHarness is natively compatible with `vLLM`'s prefix caching, as detailed in Section 4. Compared with StreamingHarness (w/o prefix cache), enabling prefix caching reduces latency from approximately 4 s to below 1 s. As a result, both TTFT and End-to-End latency of StreamingHarness remain consistently below the real-time threshold throughout the entire 2-hour video.

6.4 Ablation Study

To evaluate the effectiveness of the token weighting in Eq. (1), we vary $w_{\text{silence}}^{\text{repeated}} \in \{0.2, 0.5, 0.8\}$ while keeping $w_{\text{silence}}^{\text{first}} = 1$ and w_{response} fixed, and evaluate on the QA and Narration tasks. As shown in Figure 7 (a), the overall average score increases as $w_{\text{silence}}^{\text{repeated}}$ grows, suggesting that a larger $w_{\text{silence}}^{\text{repeated}}$ reduces mistimed responses. Empirically, however, $w_{\text{silence}}^{\text{repeated}}$ cannot be pushed too high (e.g., to 1), as this causes the model to overcommit to silence on tasks that require frequent responses (e.g., narration). We therefore strike a balance and adopt $w_{\text{silence}}^{\text{repeated}} = 0.8$ as the default,

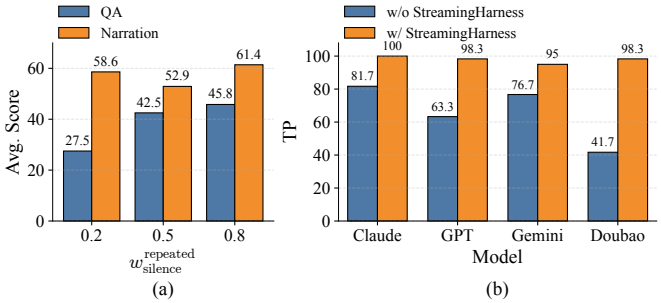


Figure 7: **Ablation studies.** (a) Avg. SW-F1 on QA and Avg. win rate on Narration under varying repeated-silence weights $w_{\text{silence}}^{\text{repeated}} \in \{0.2, 0.5, 0.8\}$ in Eq. (1). (b) Effect of StreamingHarness on the Backward task.

which enables the model to learn
when to respond and prevents an overly aggressive silence bias.

To assess the effectiveness of our plug-and-play system, StreamingHarness, we compare models equipped with it (*w/ StreamingHarness*) against their vanilla counterparts (*w/o StreamingHarness*), where the vanilla setting retains only the most recent T_s seconds of visual context, without mid-term or long-term memory. Figure 7 (b) reports the accuracy (TP) of four strong closed-source models on the *Backward* task, which requires recalling distant evidence and thus relies heavily on long-horizon memory. The results show that equipping models with StreamingHarness yields substantial and consistent improvements across all four models, demonstrating its effectiveness as a general-purpose enhancement for long-horizon streaming understanding (see Figure 11).

7 Conclusion

In this paper, we introduce a unified stack for streaming video understanding *in the wild*, consisting of three components: **Streaming-Train-248K**, a large-scale dataset with per-second frame–text alignment that adapts off-the-shelf VLMs into streaming-native models; **StreamingHarness**, a plug-and-play framework that equips any VLM with *proactive interaction*, *12-hour memory*, and *sub-second latency*; and **Streaming-Eval**, a benchmark that jointly assesses answer correctness and response timeliness under realistic streaming scenarios. Extensive experiments demonstrate that our stack outperforms strong closed-source VLMs while maintaining stable sub-second latency across hours of continuous inference. Together, these contributions demonstrate the potential of deploying VLMs in real-world streaming applications, and advance the community's transition from offline video understanding toward deployable streaming intelligence.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 62472419, 62472420)

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL `https://arxiv.org/abs/2511.21631`.

[2] Google DeepMind. Gemini 3 pro model card. Available at Google DeepMind Model Cards, 2025.

[3] Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, S. H. Cai, Yuan Cao, Y. Charles, H. S. Che, Cheng Chen, Guanduo Chen, Huarong Chen, Jia Chen, Jiahao Chen, Jianlong Chen, Jun Chen, Kefan Chen, Liang Chen, Ruijue Chen, Xinhao Chen, Yanru Chen, Yanxu Chen, Yicun Chen, Yimin Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Ziwei Chen, Dazhi Cheng, Minghan Chu, Jialei Cui, Jiaqi Deng, Muxi Diao, Hao Ding, Mengfan Dong, Mengnan Dong, Yuxin Dong, Yuhao Dong, Angang Du, Chenzhuang Du, Dikang Du, Lingxiao Du, Yulun Du, Yu Fan, Shengjun Fang, Qiulin Feng, Yichen Feng, Garimugai Fu, Kelin Fu, Hongcheng Gao, Tong Gao, Yuyao Ge, Shangyi Geng, Chengyang Gong, Xiaochen Gong, Zhuoma Gongque, Qizheng Gu, Xinran Gu, Yicheng Gu, Longyu Guan, Yuanying Guo, Xiaoru Hao, Weiran He, Wenyang He, Yunjia He, Chao Hong, Hao Hu, Jiaxi Hu, Yangyang Hu, Zhenxing Hu, Ke Huang, Ruiyuan Huang, Weixiao Huang, Zhiqi Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yu Jing, Guokun Lai, Aidi Li, C. Li, Cheng Li, Fang Li, Guanghe Li, Guanyu Li, Haitao Li, Haoyang Li, Jia Li, Jingwei Li, Junxiong Li, Lincan Li, Mo Li, Weihong Li,

Wentao Li, Xinhang Li, Xinhao Li, Yang Li, Yanhao Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zheming Li, Weilong Liao, Jiawei Lin, Xiaohan Lin, Zhishan Lin, Zichao Lin, Cheng Liu, Chenyu Liu, Hongzhang Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Tianyu Liu, Weizhou Liu, Xiangyan Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yuanxin Liu, Yue Liu, Zhengying Liu, Zhongnuo Liu, Enzhe Lu, Haoyu Lu, Zhiyuan Lu, Junyu Luo, Tongxu Luo, Yashuo Luo, Long Ma, Yingwei Ma, Shaoguang Mao, Yuan Mei, Xin Men, Fanqing Meng, Zhiyong Meng, Yibo Miao, Minqing Ni, Kun Ouyang, Siyuan Pan, Bo Pang, Yuchao Qian, Ruoyu Qin, Zeyu Qin, Jiezhong Qiu, Bowen Qu, Zeyu Shang, Youbo Shao, Tianxiao Shen, Zhennan Shen, Juanfeng Shi, Lidong Shi, Shengyuan Shi, Feifan Song, Pengwei Song, Tianhui Song, Xiaoxi Song, Hongjin Su, Jianlin Su, Zhaochen Su, Lin Sui, Jinsong Sun, Junyao Sun, Tongyu Sun, Flood Sung, Yunpeng Tai, Chuning Tang, Heyi Tang, Xiaojuan Tang, Zhengyang Tang, Jiawen Tao, Shiyuan Teng, Chaoran Tian, Pengfei Tian, Ao Wang, Bowen Wang, Chensi Wang, Chuang Wang, Congcong Wang, Dingkun Wang, Dinglu Wang, Dongliang Wang, Feng Wang, Hailong Wang, Haiming Wang, Hengzhi Wang, Huaqing Wang, Hui Wang, Jiahao Wang, Jinhong Wang, Jiuzheng Wang, Kaixin Wang, Linian Wang, Qibin Wang, Shengjie Wang, Shuyi Wang, Si Wang, Wei Wang, Xiaochen Wang, Xinyuan Wang, Yao Wang, Yejie Wang, Yipu Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhexu Wang, Zihan Wang, Zizhe Wang, Chu Wei, Ming Wei, Chuan Wen, Zichen Wen, Chengjie Wu, Haoning Wu, Junyan Wu, Rucong Wu, Wenhao Wu, Yuefeng Wu, Yuhao Wu, Yuxin Wu, Zijian Wu, Chenjun Xiao, Jin Xie, Xiaotong Xie, Yuchong Xie, Yifei Xin, Bowei Xing, Boyu Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinbo Xu, Xinran Xu, Yangchuan Xu, Yichang Xu, Yuemeng Xu, Zelai Xu, Ziyao Xu, Junjie Yan, Yuzi Yan, Guangyao Yang, Hao Yang, Junwei Yang, Kai Yang, Ningyuan Yang, Ruihan Yang, Xiaofei Yang, Xinlong Yang, Ying Yang, Yi Yang, Yi Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Dan Ye, Wenjie Ye, Zhuorui Ye, Bohong Yin, Chengzhen Yu, Longhui Yu, Tao Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Xiaokun Yuan, Yang Yue, Weihao Zeng, Dunyuan Zha, Haobing Zhan, Dehao Zhang, Hao Zhang, Jin Zhang, Puqi Zhang, Qiao Zhang, Rui Zhang, Xiaobin Zhang, Y. Zhang, Yadong Zhang, Yangkun Zhang, Yichi Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yushun Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Chenguang Zhao, Feifan Zhao, Jinxiang Zhao, Shuai Zhao, Xiangyu Zhao, Yikai Zhao, Zijia Zhao, Huabin Zheng, Ruihan Zheng, Shaojie Zheng, Tengyang Zheng, Junfeng Zhong, Longguang Zhong, Weiming Zhong, M. Zhou, Runjie Zhou, Xinyu Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu, Yuxuan Zhu, Zhen Zhu, Jingze Zhuang, Weiyu Zhuang, Ying Zou, and Xinxing Zu. Kimi k2.5: Visual agentic intelligence, 2026. URL `https://arxiv.org/abs/2602.02276`.

[4] ByteDance Seed. Seed2.0 Model Card: Towards Intelligence Frontier for Real-World Complexity. Available at ByteDance Seed Model Cards, 2026.

[5] Chaoyou Fu, Haozhi Yuan, Yuhao Dong, Yi-Fan Zhang, Yunhang Shen, Xiaoxing Hu, Xueying Li, Jinsen Su, Chengwu Long, Xiaoyao Xie, Yongkang Xie, Xiawu Zheng, Xue Yang, Haoyu Cao, Yunsheng Wu, Ziwei Liu, Xing Sun, Caifeng Shan, and Ran He. Video-mme-v2: Towards the next stage in benchmarks for comprehensive video understanding, 2026. URL `https://arxiv.org/abs/2604.05015`.

[6] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. Lvbench: An extreme long video understanding benchmark, 2025. URL `https://arxiv.org/abs/2406.08035`.

[7] Moloud Abdar, Meenakshi Kollati, Swaraja Kuraparthi, Farhad Pourpanah, Daniel McDuff, Mohammad Ghavamzadeh, Shuicheng Yan, Abduallah Mohamed, Abbas Khosravi, Erik Cambria, and Fatih Porikli. A review of deep learning for video captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2024. doi: 10.1109/TPAMI.2024.3522295.

[8] Mengxue Qu, Xiaodong Chen, Wu Liu, Alicia Li, and Yao Zhao. ChatVTG: Video Temporal Grounding via Chat with Video Dialogue Large Language Models . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1847–1856, Los Alamitos, CA, USA, June 2024. IEEE Computer Society. doi: 10.1109/CVPRW63382.2024.00191. URL `https://doi.ieeecomputersociety.org/10.1109/CVPRW63382.2024.00191`.

[9] Joya Chen, Ziyun Zeng, Yiqi Lin, Wei Li, Zejun Ma, and Mike Zheng Shou. Live: Learning video llm with streaming speech transcription at scale. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29083–29095, 2025. doi: 10.1109/CVPR52734.2025.02708.

[10] Tongtong Yuan, Xuange Zhang, Bo Liu, Kun Liu, Jian Jin, and Zhenzhen Jiao. Surveillance video-and-language understanding: From small to large multimodal models. *IEEE Trans. Cir. and Sys. for Video Technol.*, 35(1):300–314, January 2025. ISSN 1051-8215. doi: 10.1109/TCSVT.2024.3462433. URL https://doi.org/10.1109/TCSVT.2024.3462433.

[11] Yihang Li, Xuelong Wei, Jingzhou Luo, Yingjing Xiao, Yibo Bai, Guangyuan Zhou, Teng Zou, Chenguang Gui, Jiajun Wen, He Zhang, Kangliang Chen, Xing Pan, Shuaiyan Liu, Daming Wang, Tao An, Jiayi Li, Shibo Jin, Wanwan Zhang, Tianyu Wang, Boren Wei, Zhixuan Huang, Fangsheng Liu, Ruodai Li, Hui Zhang, Anson Li, Yicheng Gong, Peng Cao, Jiaming Liang, and Liang Lin. Egolive: A large-scale egocentric dataset from real-world human tasks, 2026. URL https://arxiv.org/abs/2604.23570.

[12] Yueqian Wang, Xiaojun Meng, Yuxuan Wang, Jianxin Liang, Jiansheng Wei, Huishuai Zhang, and Dongyan Zhao. VideoLLM knows when to speak: Enhancing time-sensitive video comprehension with video-text duet interaction format. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6338–6359, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.336. URL https://aclanthology.org/2025.findings-emnlp.336/.

[13] Haibo Wang, Bo Feng, Zhengfeng Lai, Mingze Xu, Shiyu Li, Weifeng Ge, Afshin Dehghan, Meng Cao, and Ping Huang. Streambridge: Turning your offline video large language model into a proactive streaming assistant. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=DTvviEnW2A.

[14] Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, Yi Wang, and Limin Wang. Streamforest: Efficient online video understanding with persistent event memory. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=9loSPaBwGO.

[15] Zhenyu Yang, Kairui Zhang, Yuhang Hu, Bing Wang, Shengsheng Qian, Bin Wen, Fan Yang, Tingting Gao, Weiming Dong, and Changsheng Xu. Livestar: Live streaming assistant for real-world online video understanding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=4n7IifN7yr.

[16] Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large language model for streaming video. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18407–18418, 2024. doi: 10.1109/CVPR52733.2024.01742.

[17] Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Dispider: Enabling Video LLMs with Active Real-Time Interaction via Disentangled Perception, Decision, and Reaction . In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24045–24055, Los Alamitos, CA, USA, June 2025. IEEE Computer Society. doi: 10.1109/CVPR52734.2025.02239. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52734.2025.02239.

[18] Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-vstream: Efficient real-time understanding for long video streams, 2025. URL https://arxiv.org/abs/2506.23825.

[19] Haowei Zhang, Shudong Yang, Jinlan Fu, See-Kiong Ng, and Xipeng Qiu. Hermes: Kv cache as hierarchical memory for efficient streaming video understanding, 2026. URL https://arxiv.org/abs/2601.14724.

[20] Xin Ding, Hao Wu, Yifan Yang, Shiqi Jiang, Qianxi Zhang, Donglin Bai, Zhibo Chen, and Ting Cao. Streammind: Unlocking full frame rate streaming video dialogue through event-gated cognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13448–13459, October 2025.

[21] Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. Streamingvlm: Real-time understanding for infinite video streams, 2025. URL https://arxiv.org/abs/2510.09608.

[22] Junming Lin, Zheng Fang, Chi Chen, Haoxuan Cheng, Zihao Wan, Fuwen Luo, Ziyue Wang, Peng Li, Yang Liu, and Maosong Sun. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12147–12151, 2026.

[23] Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. Ovo-bench: How far is your video-llms from real-world online video understanding? In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18902–18913, 2025.

[24] Yuxuan Wang, Yueqian Wang, Bo Chen, Tong Wu, Dongyan Zhao, and Zilong Zheng. Omni-immi: A comprehensive multi-modal interaction benchmark in streaming video contexts, 2025. URL https://arxiv.org/abs/2503.22952.

[25] Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. Online video understanding: Ovbench and videochat-online. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3328–3338, 2025. doi: 10.1109/CVPR52734.2025.00316.

[26] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding, 2024. URL https://arxiv.org/abs/2407.15754.

[27] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis, 2025. URL https://arxiv.org/abs/2405.21075.

[28] Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. Video-mmmu: Evaluating knowledge acquisition from multi-discipline professional videos, 2025. URL https://arxiv.org/abs/2501.13826.

[29] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines, 2020. URL https://arxiv.org/abs/2005.00343.

[30] Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos, 2017. URL https://arxiv.org/abs/1703.09788.

[31] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu,

Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18973–18990, 2022. doi: 10.1109/CVPR52688.2022.01842.

[32] Junbin Xiao, Nanxin Huang, Hao Qiu, Zhulin Tao, Xun Yang, Richang Hong, Meng Wang, and Angela Yao. Egoblind: Towards egocentric visual assistance for the blind, 2025. URL https://arxiv.org/abs/2503.08221.

[33] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zach Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, Maria Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo, Changan Chen, Xinzhu Fu, Ryosuke Furuta, Cristina Gonzalez, Prince Gupta, Jiabo Hu, Yifei Huang, Yiming Huang, Weslie Khoo, Anush Kumar, Robert Kuo, Sach Lakhavani, Miao Liu, Mi Luo, Zhengyi Luo, Brighid Meredith, Austin Miller, Oluwatumininu Oguntola, Xiaqing Pan, Penny Peng, Shraman Pramanick, Merey Ramazanova, Fiona Ryan, Wei Shan, Kiran Somasundaram, Chenan Song, Audrey Southerland, Masatoshi Tateno, Huiyu Wang, Yuchen Wang, Takuma Yagi, Mingfei Yan, Xitong Yang, Zecheng Yu, Shengxin Cindy Zha, Chen Zhao, Ziwei Zhao, Zhifan Zhu, Jeff Zhuo, Pablo Arbelaez, Gedas Bertasius, David Crandall, Dima Damen, Jakob Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C. V. Jawahar, Richard Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shou, and Michael Wray. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives, 2024. URL https://arxiv.org/abs/2311.18259.

[34] Yifei Huang, Guo Chen, Jilan Xu, Mingfang Zhang, Lijin Yang, Baoqi Pei, Hongjie Zhang, Lu Dong, Yali Wang, Limin Wang, and Yu Qiao. Egoexolearn: A dataset for bridging asynchronous ego- and exo-centric view of procedural activities in real world, 2025. URL https://arxiv.org/abs/2403.16182.

[35] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5804–5813, 2017. doi: 10.1109/ICCV.2017.618.

[36] Gunnar A. Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Charades-ego: A large-scale dataset of paired third and first person videos, 2018. URL https://arxiv.org/abs/1804.09626.

[37] WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. Query - dependent video representation for moment retrieval and highlight detection. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23023–23033, 2023. doi: 10.1109/CVPR52729.2023.02205.

[38] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015. doi: 10.1109/CVPR.2015.7298698.

[39] Mingfei Han, Linjie Yang, Xiaojun Chang, Lina Yao, and Heng Wang. Shot2story: A new benchmark for comprehensive understanding of multi-shot videos, 2025. URL https://arxiv.org/abs/2312.10300.

[40] Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21064–21074, 2022. doi: 10.1109/CVPR52688.2022.02042.

[41] Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, Neel Joshi, and Marc Pollefeys. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world, 2023. URL https://arxiv.org/abs/2309.17024.

[42] Yuwei Bao, Keunwoo Yu, Yichi Zhang, Shane Storks, Itamar Bar-Yossef, Alex de la Iglesia, Megan Su, Xiao Zheng, and Joyce Chai. Can foundation models watch, talk and guide you step by step to make a cake? In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12325–12341, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.824. URL https://aclanthology.org/2023.findings-emnlp.824/.

[43] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. Moviechat: From dense token to sparse memory for long video understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18221–18232, 2024. doi: 10.1109/CVPR52733.2024.01725.

[44] Andreea-Maria Oncescu, João F. Henriques, Yang Liu, Andrew Zisserman, and Samuel Albanie. Queryd: A video dataset with high-quality text and audio narrations. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2265–2269, 2021. doi: 10.1109/ICASSP39728.2021.9414640.

[45] Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. StreamingVLM: Real-time understanding for infinite video streams. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=gVbPWbA97s.

[46] Yujiao Shen, Shulin Tian, Jingkang Yang, and Ziwei Liu. A simple baseline for streaming video understanding, 2026. URL https://arxiv.org/abs/2604.02317.

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

[48] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180.

[49] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. Sglang: Efficient execution of structured language model programs, 2024. URL https://arxiv.org/abs/2312.07104.

[50] Xudong Lu, Huankang Guan, Yang Bo, Jinpeng Chen, Xintong Guo, Shuhan Li, Fang Liu, Peiwen Sun, Xueying Li, Wei Zhang, Xue Yang, Rui Liu, and Hongsheng Li. Phostream: Benchmarking real-world streaming for omnimodal assistants in mobile scenarios, 2026. URL https://arxiv.org/abs/2601.22575.

[51] Haroon Idrees, Amir R. Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, February 2017. ISSN 1077-3142. doi: 10.1016/j.cviu.2016.10.018. URL http://dx.doi.org/10.1016/j.cviu.2016.10.018.

[52] Abhay Zala, Jaemin Cho, Satwik Kottur, Xilun Chen, Barlas Oğuz, Yasher Mehdad, and Mohit Bansal. Hierarchical video-moment retrieval and step-captioning, 2023. URL https://arxiv.org/abs/2303.16406.

[53] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: Benchmarking multi-task long video understanding, 2025. URL https://arxiv.org/abs/2406.04264.

[54] ShuHang Xun, Sicheng Tao, Jungang Li, Yibo Shi, Zhixin Lin, Zhanhui Zhu, Yibo Yan, Hanqian Li, LingHao Zhang, Shikang Wang, Yixin Liu, Hanbo Zhang, Ying Ma, and Xuming Hu. RTV-bench: Benchmarking MLLM continuous perception, understanding and reasoning through real-time video. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=MpaSsvHcWC.

[55] Weicai Yan, Yuhong Dai, Qi Ran, Haodong Li, Wang Lin, Hao Liao, Xing Xie, Tao Jin, and Jianxun Lian. Proact-vl: A proactive videollm for real-time ai companions, 2026. URL https://arxiv.org/abs/2603.03447.

[56] Jiaer Xia, Peixian Chen, Mengdan Zhang, Xing Sun, and Kaiyang Zhou. Streaming video instruction tuning, 2025. URL https://arxiv.org/abs/2512.21334.

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

[58] Yiyu Wang, Xuyang Liu, Xiyan Gui, Xinying Lin, Boxue Yang, Chenfei Liao, Tailai Chen, and Linfeng Zhang. Accelerating streaming video large language models via hierarchical token compression, 2026. URL https://arxiv.org/abs/2512.00891.

[59] Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, TaoZhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video KV-cache retrieval. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8g9fs6mdEG.

[60] Dingyu Yao, Bowen Shen, Zheng Lin, Wei Liu, Jian Luan, Bin Wang, and Weiping Wang. TailorKV: A hybrid framework for long-context inference via tailored KV cache optimization. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20340–20359, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1043. URL https://aclanthology.org/2025.findings-acl.1043/.

[61] Donghyuk Kim, Sejeong Yang, Wonjin Shin, and Joo-Young Kim. V-rex: Real-time streaming video llm acceleration via dynamic kv cache retrieval. In *2026 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 1–14, 2026. doi: 10.1109/HPCA68181.2026.11408603.

[62] Dingyu Yao, Chenxu Yang, Zhengyang Tong, Zheng Lin, Wei Liu, Jian Luan, and Weiping Wang. Vecinfer: Efficient llm inference with low-bit kv cache via outlier-suppressed vector quantization, 2025. URL https://arxiv.org/abs/2510.06175.

[63] Haomiao Xiong, Zongxin Yang, Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Jiawen Zhu, and Huchuan Lu. Streaming video understanding and multi-round interaction with memory-enhanced knowledge. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=JbPb6RieNC.

[64] Yiweng Xie, Bo He, Junke Wang, Xiangyu Zheng, Ziyi Ye, and Zuxuan Wu. Fluxmem: Adaptive hierarchical memory for streaming video understanding, 2026. URL https://arxiv.org/abs/2603.02096.

[65] Shiwei Wu, Joya Chen, Kevin Qinghong Lin, Qimeng Wang, Yan Gao, Qianli Xu, Tong Xu, Yao Hu, Enhong Chen, and Mike Zheng Shou. VideoLLM-mod: Efficient video-language streaming with mixture-of-depths vision computation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=NKPXHzYusG.

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

[67] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020. URL https://arxiv.org/abs/1910.03771.

[68] Dibyadip Chatterjee, Zhanzhong Pang, Fadime Sener, Yale Song, and Angela Yao. Don't pause! every prediction matters in a streaming video, 2026. URL https://arxiv.org/abs/2604.24317.

A Limitations

Our current framework operates in a vision-language model setting and does not incorporate the audio modality. Integrating an omni-modal large model with speech understanding capabilities could further enhance narration quality, particularly for scenarios where audio cues (e.g., crowd reactions, referee whistles) carry critical information.

Second, to balance memory length and inference efficiency across the majority of real-world scenarios, we conduct our experiments at 1 FPS. However, certain domains demand higher temporal resolution to capture rapid visual changes, such as subtle facial expressions, fast-paced athletic movements, or brief but decisive moments in competitive sports. Our framework is not specifically optimized for these cases. In principle, one could reduce the memory length and increase the frame rate to accommodate such scenarios, but we leave a systematic exploration of this trade-off to future work.

B Related Works

Streaming Video Understanding Beyond a versatile VLM backbone, streaming video understanding also demands proactive interaction, long-horizon memory, and real-time inference. However, most existing methods address only one or two of these capabilities in isolation: ***Proactive Interaction.*** A core challenge in streaming video understanding is deciding whether and when to respond under continuous input. VideoLLM-online [16] pioneers training the model to emit EOS on redundant frames and respond on informative ones. Later works explore alternative designs, including a decision token [55], response-state tokens unified with content generation [56], perplexity-based gating [15], and asynchronous decoupling of perception, decision, and reaction [57]. ***Long-horizon Memory.*** As streaming videos are unbounded in length, retaining all visual tokens incurs prohibitive memory and compute costs, motivating memory management. Some methods optimize visual tokens or the KV cache, e.g., redundant token pruning [47, 58], KV cache offloading [59–61], and hierarchical KV cache memory [19, 62]; others [63, 18] introduce dedicated memory architectures that store compressed representations and retrieve relevant content at query time. ***Real-time Inference.*** Unlike offline video understanding, streaming video requires low-latency responses synchronized with incoming frames, necessitating dedicated inference-efficiency optimizations. Existing efforts focus on the following directions: constraining the context via a fixed-size sliding window [45, 46], pruning redundant visual tokens selectively [64, 65], and decoupling perception from response generation through a two-stage pipeline [20, 66]. However, these efficiency gains come at the cost of long-term memory, and current methods struggle to jointly achieve real-time inference and hours-long memory. Moreover, most existing methods rely on native `Transformers` [67] and are incompatible with efficient inference engines such as `vLLM` [48] and `SGLang` [49], which limits their deployment in real-world production environments.

Streaming Video Benchmarks As for evaluation, existing mainstream benchmarks [9, 23, 25, 22] primarily assess the correctness of responses. More critically, constrained by online inference frameworks, these benchmarks reduce inference to a single-turn offline-style paradigm, which requires the model to respond immediately, with questions restricted to past or present observed content [68]. Therefore, a well-designed benchmark is essential for advancing research on streaming video. To align with in-the-wild streaming video scenarios, Streaming-Eval evaluates streaming capabilities under online deployment conditions.

C More Details of StreamingHarness

C.1 Prompt Template

System Prompt

You are a real-time streaming video agent. Each user turn contains the current timestamped frame, and earlier turns in the same chunk show recent visual context.

Output exactly one of:

- `</silence>`
- `</response>` one concise answer or commentary sentence

Rules:

- Your output must begin with `</silence>` or `</response>`.
- Use `</silence>` when no user query is active or when there is no useful new response for the active query.
- When a user query is active, answer that query using the current visual evidence, recent chunk context, and provided video memory.
- Do not invent names, scores, labels, or facts that are not visible or provided in the prompt.

D Evaluation

D.1 Narration

LLM-as-Judge Prompt

You are a strict judge of live-video narration / commentary. You will see a human reference commentary (time-stamped) and two candidate commentaries A and B produced on the same video. Decide which candidate as a whole better matches the reference.

Criteria, equally important:

1. **Narration style (pacing + tone)** — real commentators stay SILENT most of the time and speak only at key moments in natural spoken language. Candidates that narrate every second or read like written descriptions are more like captioning than narration.
2. **Consistency & accuracy** — every detail stays synchronized with the reference and never contradicts it; no hallucinated facts, covers the key moments.

If a candidate is empty, the other wins unless both are empty (tie). If both candidates are roughly equally good or equally bad, output `tie`.

Return STRICT JSON on one line:
`{"winner": "A"|"B"|"tie", "reason": "<=500 words"}`

[REFERENCE COMMENTARY]
`{gt_transcript}`

[CANDIDATE A]
`{a_pred}`

[CANDIDATE B]
`{b_pred}`

E Case Study



Ground Truth

[245s] Keep the racket vertical for at least a half a foot to a foot.
[250s] And then at that point, you can let your wrist and your forearm relax as you finish.
[256s] But don't be thinking, roll the forearm and wrist in the contact area.
[263s] The contact area is an area
[267s] of a foot to a foot and a half where you want to keep the racket vertical as it goes through.

GPT 5.4 (High): (Model A)

[241s-242s] He holds the racket up at contact, showing the strings meeting the middle of the ball right at the equator.
[244s-245s] He moves the ball away from the strings, underlining how quickly contact is made before the racket continues through.
[245s-246s] He drops the ball hand and keeps the racket path steady, emphasizing that the hit is clean and immediate.
[246s-247s] He holds the racket out in front, showing a compact follow-through and a controlled path through the contact zone.
[249s-250s] He keeps the racket face firm out in front, reinforcing a smooth extension instead of any roll over the ball.
[250s-251s] He demonstrates the forehand contact out in front, keeping the racket face firm as he swings up through the ball.
[251s-252s] He holds the contact point steady, showing the racket face staying controlled and slightly closed through the forehand.
[252s-253s] The racket now rises through contact, emphasizing that low-to-high path that creates topspin and net clearance.
[253s-254s] He finishes higher across the body, underlining the upward swing that keeps the forehand heavy and safe over the net.
[254s-255s] He completes the forehand with a high finish, reinforcing the full low-to-high motion and controlled topspin release.
[255s-256s] He holds the finish by the shoulder, spotlighting balance and a clean, controlled forehand follow-through.
…
[265s-266s] He positions the racket face next to the ball, clearly showing the exact strike zone for clean forehand contact.
[266s-267s] He lines the strings up with the middle of the ball, illustrating that clean forehand contact point right at the equator.
[267s-268s] He holds the racket steady through the ball, stressing a centered strike rather than cutting over the top.
[268s-269s] Now he drops the racket head below the ball, showing the low-to-high path that lifts the forehand safely over the net.
[269s-270s] He sinks the racket even lower under the contact point, underlining that upward swing path for topspin and clearance.
[270s-271s] He freezes at the strike zone, holding the ball in front to show the exact contact point and the racket rising up through it.
[271s-272s] He holds the ball next to the strings, emphasizing clean contact right around the middle of the ball.
[272s-273s] He swings up and through now, showing the racket face staying steady as it rises after contact.

StreamingHarness-8B: (Model B)

[256s-257s] So if I bring the racket back here and I swing up from here, that's vertical contact.
[257s-258s] All right, okay?
[259s-260s] If I bring it back and swing up, it's still vertical.
[262s-263s] That racket face is tilted forward.
[266s-267s] A lot of times when players bring this racket forward, it comes at the strings, it ratchets down, you gotta help it go up.
[269s-270s] What I'm saying is, when you swing the racket up, the racket's gonna tilt forward.

LLM-Judge

Candidate B far better matches the reference on both key criteria. First, narration style: the reference is sparse, first-person coaching instruction delivered at key moments ('Keep the racket vertical,' 'don't be thinking, roll the forearm'). Candidate B mirrors this first-person coaching voice and uses natural spoken language. Candidate A, by contrast, narrates virtually every single second in third-person descriptive prose ('He holds the racket up,' 'He moves the ball away') — this is pure visual captioning, not live commentary, and violates the core pacing requirement. Second, accuracy and consistency: Candidate B directly addresses the reference's central technical points — keeping the racket vertical through contact (256s: 'that's vertical contact'), not rolling the wrist/forearm (discussed around 269–273s), and the defined contact zone (466–472s: 'one foot below the contact point'). These align meaningfully with the reference's instructions about a foot-to-foot-and-a-half vertical contact area and avoiding forearm/wrist roll. Candidate A never substantively engages with these specific cues; it only describes physical movements without conveying the instructional content. Candidate B is imperfect — it adds some details not in the reference (closed racket position, palm-down cues) and is still somewhat wordy in places — but it is unambiguously closer to the reference in both style and substance. Candidate A's wall-to-wall third-person narration makes it the clear loser.

Figure 8: Narration comparison between GPT 5.4 (High) and StreamingHarness-8B.



Ground Truth

[244s] Logan Black, that's the strikeout of Catherine Mattis.
[250s] Mattis, she had been unable to put away yet.
[253s] Which they added on to in the fifth.
[256s] This one scooped up, touched second, flipped over to first for the double play.
[263s] Wisely done by Caitlin Hunt.
[267s] A little blooper.
[271s] Drops in.
[272s] 100 pitches now for Logan Black in this one.
[279s] Number 101 is good for a strikeout of Emily Supersinski, now with a 2-2.
[286s] delivers in there for strike three.
[289s] Second straight strikeout for Logan Black in this one and that brings her total on.

Claude Opus 4.6 Thinking (w/o StreamingHarness): (Model A)

[244s-245s] Count moves to 1-2 now, Black took a ball there. She resets and looks in for the sign, still trying to finish off this at-bat.
[245s-246s] Looks like there's some movement on the basepaths — runners are on with one out and the count at 1-2, Creighton needs to capitalize here trailing 3-1.
[246s-247s] Struck her out! That's the fourth strikeout of the game, and two down now in the top of the fifth — Kylie Miller continues to deal from the circle.
[247s-248s] The batter walks away disappointed — that's two outs now in the top of the fifth as the next Seton Hall hitter steps in, Pirates still leading 3-1.
…
[283s-284s] And that's the end of the sixth — middle of the sixth now, Seton Hall still leading 3-1. Black at 101 pitches. Creighton couldn't capitalize with the bases loaded, and the Pirates head into the top of the seventh still holding that two-run cushion.
[284s-285s] Bottom of the seventh now, last chance for Creighton. Number 11 at the plate, count 2-2, nobody out. Black up to 105 pitches — the Blue Jays need two runs to tie it up. This is do-or-die time.
…
[319s-320s] Players clearing the field now as we transition to the bottom of the seventh — Creighton's last chance to chip into that two-run Seton Hall lead.
[322s-323s] Creighton Blue Jays graphic up on the screen as we head into a break with the game all tied up.

Claude Opus 4.6 Thinking (w/ StreamingHarness): : (Model B)

[246s-247s] There it is — the fourth strikeout of the game for Logan Black, as the graphic confirms it on screen, Seton Hall maintaining that 3-1 lead.
[247s-248s] Two outs now in the bottom of the fifth, the Creighton batter heads back to the dugout — the Jays running out of chances to close this gap.
[248s-249s] Number 11 makes the slow walk back as the next Creighton batter prepares to step in — two outs, two innings left to mount a comeback.
…
[273s-274s] There's a play developing at third base — the runner comes in as the fielder receives the throw, umpire positioned perfectly to make the call on this bang-bang play.
[274s-275s] Logan Black now at 100 pitches on the day — Seton Hall still leads 3-1 in the bottom of the sixth, two outs, runners on with a 1-2 count. Number 11 steps back toward the plate as the Jays need a big at-bat here to keep the rally alive.
[275s-276s] The Creighton batter digs in, bat ready, as Black prepares to deliver — 1-2 count, two outs, bottom six. This is a critical at-bat with the tying run on base for the Bluejays.
[277s-278s] Number 11 strides toward the circle as the batter sets her stance — the Birdcage crowd watching intently, willing something to happen here with two down in the sixth.
[278s-279s] Black winds up and delivers — pitch 101 on the way, 1-2 count, the batter loads up ready to swing.
[279s-280s] The batter swings — she gets a cut on it, driving through the zone as Black completes her follow-through.
…
[318s-319s] The team fully breaks now, players scattering toward the dugout as the transition to the bottom of the seventh begins. You can see Seton Hall players also getting into position — the Pirates looking to close this one out, up 3-1 with just three outs to go.
[319s-320s] Players jogging off the field now, number 23 leading the way back to the dugout. The Bluejays need to dig deep here — bottom seven, down two, backs against the wall. Let's see who steps up to the plate first for Creighton.

LLM-Judge

Both candidates are deeply flawed against the reference: they narrate almost every second in a caption-like style, whereas the reference is extremely sparse—speaking only at key moments. Neither candidate names the specific players called out in the reference (Catherine Mattis, Emily Supersinski, Caitlin Hunt), and neither accurately describes the double play at ~256s ('scooped up, touched second, flipped over to first'). However, Candidate B is marginally better for two reasons. First, Candidate A introduces a clear factual error at 246s by attributing a strikeout to 'Kylie Miller' instead of Logan Black—a direct contradiction of the reference. Second, Candidate B's pitch-count tracking (mentioning 100 pitches around 274s and 101 pitches at 283s) aligns more closely with the reference's 100-pitch mention at 272s and 101st pitch at 279s, whereas Candidate A's pitch counts (84–86 throughout the same window) are significantly off. Both candidates are far too verbose and over-describe routine action, failing the 'real commentators stay silent most of the time' criterion almost entirely, so neither is a good match for the reference's style or accuracy—but B's fewer outright factual errors give it the narrow edge.

Figure 9: Narration comparison between Claude Opus 4.6 Thinking (w/o StreamingHarness) and Claude Opus 4.6 Thinking (w/ StreamingHarness).



Query

[171s] What is the first thing the person does immediately after cutting the salmon?\nA. Starts frying the fish.\nB. Stores the fish in the fridge.\nC. Chops the broccoli.\nD. Cleans tools/hands.\nRespond with ONLY the option letter and text, for example: A. Example option.

Ground Truth

[193s-202s] D. Cleans tools/hands.

StreamingHarenss-8B

[180s] D. Cleans tools/hands. [Response Time Error]
[201s] D. Cleans tools/hands. [Correct]

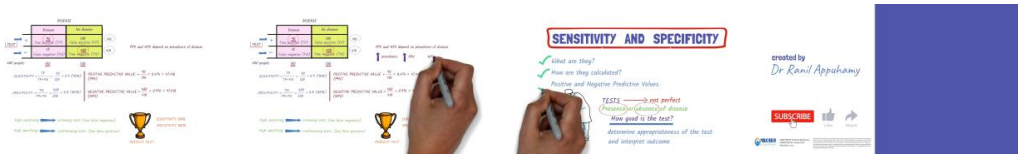
Claude Opus 4.6 Thinking (w/ StreamingHarness):

[171s] D. Cleans tools/hands. [Response Time Error]
[181s] D. Cleans tools/hands. [Response Time Error]
[182s] D. Cleans tools/hands. [Response Time Error]
…
[192s] D. Cleans tools/hands. [Response Time Error]
[193s] D. Cleans tools/hands. [Correct]
… [Correct]
[202s] D. Cleans tools/hands. [Correct]
[203s] D. Cleans tools/hands. [Response Time Error]
[204s] D. Cleans tools/hands. [Response Time Error]
… [Response Time Error]

Gemini 3.1 Pro (High) (w/ StreamingHarness):

[195s] D. Cleans tools/hands. [Correct]
[201s] D. Cleans tools/hands. [Correct]
[216s] D. Cleans tools/hands. [Response Time Error]
[217s] D. Cleans tools/hands. [Response Time Error]
... [Response Time Error]
[221s] D. Cleans tools/hands. [Response Time Error]

Figure 10: Forward Task Example.



Query

[360s] Earlier in the worked example with 600 people, what specificity was calculated for the test?\nA. 47.4%.\nB. 97.6%.\nC. 80%.\nD. 90%.\nRespond with ONLY the option letter and text, for example: A. Example option.

Ground Truth

[360s-365s] C. 80%.

StreamingHarenss-8B
[360s] C. 80%.[Correct]
[361s] C. 80%.[Correct]

Claude Opus 4.6 Thinking (w/ StreamingHarness):
[360s] C. 80%.[Correct]

Claude Opus 4.6 Thinking (w/o StreamingHarness):
[360s] B. 97.6%. [Incorrect Answer]
[367s] B. 97.6%. [Incorrect Answer]

Gemini 3.1 Pro (High) (w/ StreamingHarness):
[360s] C. 80%.[Correct]

Gemini 3.1 Pro (High) (w/o StreamingHarness):
[361s] D. 90.0%. [Incorrect Answer]

GPT 5.4 (High) (w/ StreamingHarness):
[360s] C. 80%.[Correct]

GPT 5.4 (High) (w/o StreamingHarness):
[364s] D. 90.0%. [Incorrect Answer]

Figure 11: Backward Task Example.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We elucidate our contributions in the Abstract and Intrudctions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In Appendix A, we have thoroughly discussed the limitations of our article, hoping to guide more future work.

   Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: Our study is mainly empirical exploration.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We report detailed training and inference settings in Section 6.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: This may be temporary, and we are working hard to promote the process of open source.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: In Section 6.1, we provide a comprehensive description of the experimental settings, including hyperparameters, model settings, training settings, and evaluation settings.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Given the high cost of Supervised Fine-Tuning, we do not repeat experiments under identical configurations.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We discussion computation cost in Section 6.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We guarantee that the research conducted in the paper complies with NeurIPS Code of Ethics in all aspects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The purpose of this paper is to advance streaming video understanding, without any negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of the assets used in the paper, such as code, data, and models, have been appropriately recognized, and the licenses and terms of use have been clearly mentioned and properly respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Related codes will be released with suitable license.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [No]

    Justification: LLM is used only for writing.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.