StreamMind: Dynamic Streaming Cognition for Online Video Understanding

Zhuojie Wu[1,2] **Xin Shen**[3] **Chenxi Miao**[2] **Weikang Li**[2†]
Liwei Qian[2] **Xin Pei**[2] **Xin Yu**[3✉]
[1]The University of Queensland [2]Baidu Inc. [3]Adelaide University
zhuojie.wu@uq.edu.au

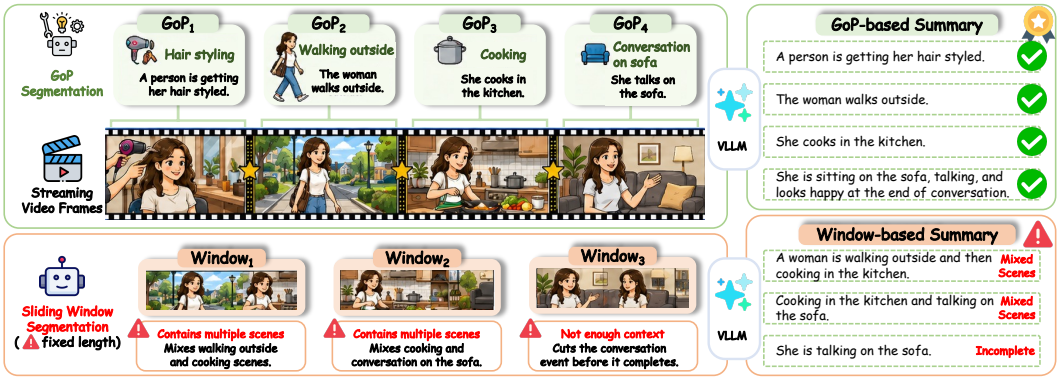

Figure 1: Motivation for StreamMind. Fixed-length windows (*bottom*) impose uniform temporal partitions on evolving video streams. Such partitions may merge distinct events or split an ongoing event across windows, fragmenting the accumulated context. In contrast, StreamMind (*top*) processes streams online over GoP-level segments, which serve as coherent units for reasoning and memory.

Abstract

Online video large language models (VLLMs) have shown remarkable capabilities in real-time streaming video understanding and proactive human-AI interaction. However, existing methods process video streams in rigid temporal units, overlooking the non-uniform temporal evolution of streaming content. This mismatch forces the model to reason over incomplete or mixed events, resulting in flickering responses and fragmented context. We observe that compressed videos are naturally organized into Groups of Pictures (GoPs), which form coherent temporal units that adapt to content. Building on this observation, we propose **StreamMind**, a dynamic streaming cognition framework that organizes reasoning and memory around GoP-level segments. Specifically, StreamMind introduces two key components. First, a **Dynamic Thinking (DynThink)** module accumulates evidence within each GoP and triggers reasoning only at its boundary, grounding decisions in complete evidence rather than transient cues. Second, a **Brain-Eye Synergy Memory (BESM)** module compresses each GoP into a compact memory slot for long-horizon understanding. It retains key-frame visual tokens as high-fidelity anchors (*Eye*) and replaces other frames with DynThink's thinking tokens (*Brain*). This design preserves visual detail and reasoning continuity while reducing redundancy. Extensive experiments on streaming and offline video benchmarks demonstrate that StreamMind achieves state-of-the-art online video understanding performance, reaching 63.4% on OVO-Bench and 80.2% on StreamingBench, while maintaining strong generalization to offline long-form video reasoning.

† Project lead.
✉ Corresponding author.

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

1 Introduction

The rapid growth of live streaming and real-time interactive AI assistants has created a strong demand for *online video understanding*. Here, a model must continuously perceive an incoming video stream and proactively interact with users in real time. This task differs fundamentally from offline video understanding. Offline methods assume access to the full video before inference [2, 7, 24, 26, 30, 49, 62, 67]. In contrast, online inference is strictly causal. Frames arrive one by one, and the model cannot see future frames. Moreover, user queries may be issued at any time. The model must therefore keep up with the ongoing stream, preserve relevant past context under limited memory, and remain ready to respond under strict latency constraints. These characteristics make online video understanding a distinct problem that requires online perception, long-horizon memory, and temporally grounded interaction [36, 46, 56].

Existing efforts have made substantial progress in online video understanding. Recent methods extend offline video large language models (VLLMs) to the online setting through causal modeling, memory compression, proactive response, and streaming reasoning [4, 15, 19, 46, 56–58, 64]. However, they process the stream in *rigid temporal units*, such as fixed frame chunks or uniform update intervals. This design simplifies online processing but overlooks the non-uniform temporal evolution of real-world streams. As illustrated in Fig. 1, such fixed units may break coherent events into arbitrary fragments or merge distinct events into the same internal state. This temporal misalignment causes unstable reasoning and flickering responses, and also fragments the accumulated context.

To address this misalignment, online video understanding should be built upon coherent temporal units instead of rigid ones. We observe that compressed video streams are naturally organized into Groups of Pictures (GoPs) [50]. Each GoP forms a coherent unit of temporally correlated frames, and its boundaries can adapt to changes in visual content. Motivated by this observation, we propose **StreamMind**, a dynamic streaming cognition framework for online video understanding. StreamMind uses GoPs as the basic unit for both proactive response and memory formation. Accordingly, StreamMind introduces two key components, a **Dynamic Thinking** module and a **Brain-Eye Synergy Memory** module.

First, we propose a **Dyn**amic **Think**ing (**DynThink**) module to mitigate reasoning instability and response flickering. Specifically, DynThink adopts GoPs as dynamic temporal units. A GoP consists of a key frame and its temporally dependent frames. It serves as a coherent temporal unit for visual evidence accumulation. Unlike rigid temporal units, GoPs have dynamic and content-adaptive boundaries driven by the encoder's scene-cut policy. DynThink accumulates visual evidence within each GoP and triggers reasoning only at its boundary. Thus, the response decision is based on complete GoP-level evidence rather than arbitrary fragments. This improves the stability of online reasoning and the consistency of proactive responses.

Second, we propose a **B**rain-**E**ye **S**ynergy **M**emory (**BESM**) module to maintain coherent long-horizon context over streaming videos. In particular, BESM builds one memory slot for each GoP. In this way, memory and reasoning share the same temporal units. Since frames within a GoP are temporally correlated, storing all their visual tokens is redundant and memory-inefficient. BESM therefore retains the key-frame visual tokens as high-fidelity anchors (*Eye*). It replaces the remaining dependent frames with compact thinking tokens (*Brain*) generated by DynThink's reasoning. Each memory slot thus preserves both fine-grained visual evidence and semantic context. By coupling GoP-level reasoning with GoP-aligned memory, BESM reduces redundant storage while preserving temporal continuity for long-horizon streaming understanding.

In summary, our main contributions are as follows:

- We propose **StreamMind**, a dynamic streaming cognition framework for online video understanding. It organizes online reasoning and memory formation around GoPs, which serve as coherent temporal units.
- We introduce a **Dyn**amic **Think**ing (**DynThink**) module. It treats each GoP as a continuous visual segment and reasons only at its boundary, stabilizing reasoning and responses.
- We propose a **B**rain-**E**ye **S**ynergy **M**emory (**BESM**) module. It builds GoP-level memory slots with key-frame visual anchors and compact thinking tokens, reducing redundant storage while preserving temporal continuity.

- Extensive experiments show that StreamMind achieves state-of-the-art online video understanding performance, reaching 63.4% on OVO-Bench and 80.2% on StreamingBench. It also maintains strong generalization to offline long-form video reasoning.

2 Related Work

2.1 Streaming Video Understanding

Recent video large language models (VLLMs) have demonstrated remarkable success in offline video understanding [2, 24, 26, 38, 47, 53]. However, these methods assume access to the full video before inference. This assumption does not hold in online video understanding, where frames arrive sequentially, future observations are unavailable, and responses may be required at any moment. To extend VLLMs to the online setting, recent works have explored causal modeling, streaming-oriented training, and incremental inference. StreamingVLM [56] and Streamo [54] improve online adaptation through chunk-overlap training strategies and streaming instruction tuning, respectively. VideoLLM-online [4], Flash-VStream [63], and Memento [21] further develop incremental architectures that process incoming frames on the fly and support real-time interaction.

A critical challenge in online video understanding is long-horizon context management under bounded memory and latency budgets. Existing methods typically maintain compressed memory or key-value (KV) caches through fixed-size memory pools, heuristic truncation, or token selection strategies. Dispider [36] adopts scene-based clips, but detects boundaries via extra frame-similarity computation and decouples decision from generation. StreamMem [57] builds a query-agnostic memory bank by merging historical context using proxy query attention. QueryStream [65] adopts a query-aware pruning strategy to discard task-irrelevant tokens dynamically.

However, most existing methods process the stream in *rigid temporal units* (*e.g.*, fixed frame chunks or uniform update intervals), which do not match the non-uniform temporal evolution of video streams. This misalignment causes unstable reasoning, flickering responses, and fragmented context. To address this, StreamMind adopts Groups of Pictures (GoPs) as coherent temporal units with content-adaptive boundaries for both online reasoning and memory formation. DynThink grounds reasoning in complete GoP-level evidence, and BESM maintains coherent long-horizon context.

2.2 Multimodal Chain-of-Thought Reasoning

Multimodal chain-of-thought (CoT) equips large vision-language models with the ability to disentangle complex visual cues through intermediate logical steps. Traditional paradigms predominantly operate in an offline manner. They typically either convert fully observed visual inputs into textual abstractions prior to inference [14, 18, 48], or interleave visual and textual tokens across a complete, static context window to promote cross-modal interaction [13, 16, 39]. While some recent efforts reduce computational overhead via adaptive reasoning [33] or compact CoT representations [55, 68], these methods still assume that the complete input is available before reasoning. Consequently, they incur unacceptable latency and cannot adapt to continuous, unbounded video streams.

To bridge this gap, a nascent line of research has shifted toward streaming cognition and proactive interaction [44]. Recent frameworks amortize reasoning latency over the input stream by interleaving perception with incremental reasoning steps, enabling models to think while watching [15, 32, 43, 64]. However, these efforts mainly optimize reasoning efficiency over rigid temporal units, rather than addressing when reasoning should be triggered and how reasoning traces should be preserved under the non-uniform temporal evolution of video streams. StreamMind is complementary to these directions. Instead of reasoning over fixed frame chunks or full offline inputs, StreamMind performs online reasoning over dynamic, content-adaptive GoPs through the DynThink module, which triggers reasoning at GoP boundaries after evidence accumulation within each coherent segment. It further couples visual evidence and reasoning traces of each GoP in the BESM module, enabling efficient long-horizon memory while preserving temporal continuity for online video understanding.

3 Preliminaries

3.1 Video Codec

Modern video compression standards (*e.g.*, MPEG [23], H.264 [50], and HEVC [40]) reduce storage and transmission costs by exploiting temporal redundancy in videos. Instead of encoding every frame independently, these codecs represent a video $\mathcal{V}$ as a sequence of T frames $\{F^{(1)}, \ldots, F^{(T)}\}$ organized into Groups of Pictures (GoPs). Frames in a GoP typically use one of three coding types: **I-frames** (intra-coded), **P-frames** (predictive), and **B-frames** (bidirectional predictive).

Group of Pictures. A typical GoP starts with an I-frame, followed by a sequence of inter-coded frames such as P-frames and optionally B-frames (*e.g.*, $I\ B\ P\ P\ B\ P\ \ldots$). The I-frame is encoded independently and serves as the key frame of the GoP. The subsequent inter-coded frames are predicted from previously decoded reference frames. Thus, they are temporally dependent on the I-frame, either directly or indirectly. The encoder starts a new GoP by inserting an I-frame, *e.g.*, when its scene-cut policy detects a substantial content change or when the maximum GoP length is reached. As a result, GoP boundaries are dynamic and content-adaptive rather than fixed.

In practical streaming systems, B-frames are often disabled to reduce latency, as bidirectional prediction requires frame reordering and additional buffering during encoding and decoding [22, 35]. In this case, each GoP consists of an I-frame followed by P-frames that are predicted only from preceding frames, which is naturally compatible with causal online processing.

3.2 Streaming Video Understanding

Most existing video large language models (VLLMs) [5, 9, 47] operate in an offline setting, where the entire video is available before inference. Given a query Q, the model generates the answer A after observing the complete video $\mathcal{V}$. This paradigm is impractical for real-time streaming scenarios, where frames arrive sequentially and future frames are unavailable.

In contrast, online video understanding is strictly causal. At each time step t, the model can only access the partial observations $\mathcal{V}_{\leq t} = \{F^{(1)}, \ldots, F^{(t)}\}$ ($t \leq T$), and a user query Q may be issued at any time. The model must therefore determine whether the current observations $\mathcal{V}_{\leq t}$ contain sufficient evidence to respond to Q. If the evidence is insufficient, the model defers its response and continues accumulating observations from the incoming stream. Once sufficient evidence becomes available, the model proactively generates a response in real time.

4 StreamMind

We now present **StreamMind**, a dynamic streaming cognition framework that builds online video understanding upon dynamic, content-adaptive temporal units. Specifically, StreamMind instantiates these units as GoPs, whose boundaries adapt to changes in visual content. We first provide an overview in §4.1. Then we introduce two key components of StreamMind: (1) a **Dynamic Thinking (DynThink)** module (§4.2), which mitigates reasoning instability and response flickering by accumulating evidence within each GoP and triggering reasoning only at GoP boundaries; and (2) a **Brain-Eye Synergy Memory (BESM)** module (§4.3), which compresses each processed GoP into a compact memory slot by coupling key-frame visual anchors (*Eye*) with reasoning traces (*Brain*).

4.1 Overview

As illustrated in Fig. 2, StreamMind processes the incoming video stream with GoPs as dynamic temporal units instead of rigid ones. Formally, the stream $\mathcal{V}$ is partitioned into consecutive GoPs $\{G_1, G_2, \ldots\}$, where $G_k = \{v_{k,1}, \ldots, v_{k,N_k}\}$ contains N_k frames and starts with its I-frame $v_{k,1}$. Once G_k is fully received (*i.e.*, at its boundary), StreamMind processes it with a sequential two-stage pipeline, in which a reasoning-and-response stage handled by DynThink is followed by a memory-writing stage handled by BESM. Let E_k denote the dense visual tokens extracted from G_k, $\mathcal{M}_{k-1}$ the memory accumulated over previous GoPs, and $\mathcal{P}_{k-1}$ the available textual context (including system prompts, prior dialogue history, and user queries). The overall flow is formulated as follows:

$$(\mathcal{R}_k, d_k, A_k) = \mathrm{DynThink}(\mathcal{P}_{k-1}, \mathcal{M}_{k-1}, E_k), \qquad \mathcal{M}_k = \mathrm{BESM}(\mathcal{M}_{k-1}, E_k, \mathcal{R}_k). \tag{1}$$

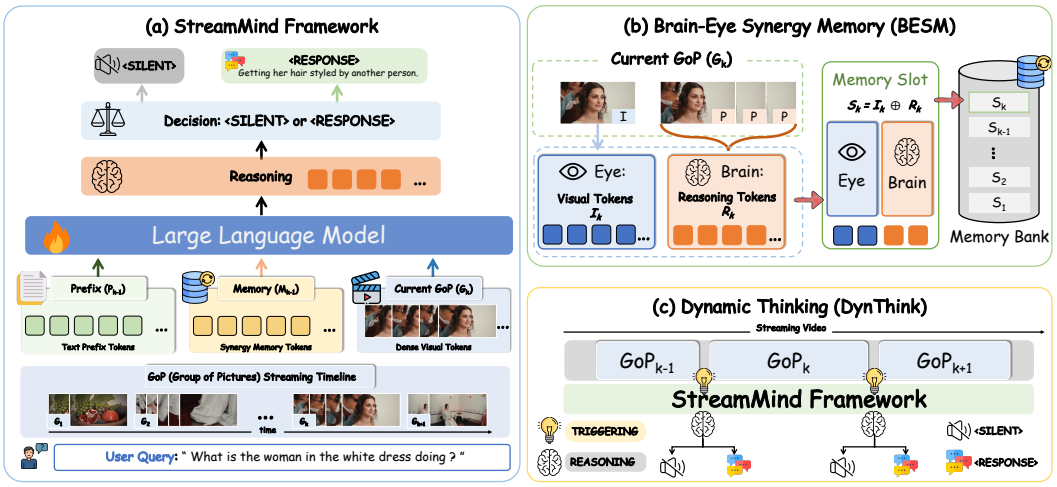

Figure 2: Overview of StreamMind. (a) StreamMind organizes online reasoning and memory around GoP-level segments. (b) BESM stores each processed GoP as a compact Brain-Eye memory slot by combining key-frame visual anchors with reasoning traces. (c) DynThink accumulates evidence within each GoP and decides whether to respond at GoP boundaries.

In the first stage, DynThink reasons over the textual context $\mathcal{P}_{k-1}$, the historical memory $\mathcal{M}_{k-1}$, and the visual tokens E_k of the current GoP to generate a GoP-level reasoning trace $\mathcal{R}_k$ and a decision token d_k. If d_k indicates a response, the model generates a user-facing response A_k and updates the textual context as $\mathcal{P}_k = \mathcal{P}_{k-1} \oplus A_k$, where $\oplus$ denotes sequence concatenation. Otherwise, the model remains silent and keeps $\mathcal{P}_k = \mathcal{P}_{k-1}$. In the second stage, BESM compresses E_k and $\mathcal{R}_k$ into a compact memory slot and appends it to $\mathcal{M}_{k-1}$, yielding the updated memory $\mathcal{M}_k$ for the next GoP.

4.2 Dynamic Thinking

Motivation. In proactive streaming interactions, the model must determine when to trigger reasoning and emit responses. Most existing methods make this decision at a preset pace, regardless of how the content evolves. Thus, a rigid temporal unit may either cut through an event, leaving its evidence incomplete, or straddle an event boundary, mixing evidence across events. In both cases, the model reasons from partial or mixed observations, so its conclusion may change as later frames arrive. This causes unstable reasoning and flickering responses.

Key Insight. Human cognition does not base complex reasoning on isolated moments but on temporally coherent segments. GoPs naturally provide such segments, since their boundaries follow changes in visual content and thus usually align with event transitions. DynThink therefore accumulates evidence within each GoP, then reasons and decides only at its boundary. If an event continues beyond the current GoP, DynThink can remain silent and carry its reasoning forward through the historical memory.

Dynamic Thinking. To implement GoP-level proactive interaction without external decision modules, we integrate the response decision into the autoregressive generation of the LLM. At the boundary of G_k, the model has access to the textual context $\mathcal{P}_{k-1}$, the historical memory $\mathcal{M}_{k-1}$, and the dense visual tokens E_k. Based on this context, the LLM first generates the reasoning trace $\mathcal{R}_k$.

$$\mathcal{R}_k = \texttt{LLM}_\theta(\mathcal{P}_{k-1} \oplus \mathcal{M}_{k-1} \oplus E_k). \tag{2}$$

After reasoning, the model predicts a discrete decision token $d_k \in \{\texttt{<RESPONSE>}, \texttt{<SILENT>}\}$ to determine whether a response is required.

$$d_k = \texttt{LLM}_\theta(\mathcal{P}_{k-1} \oplus \mathcal{M}_{k-1} \oplus E_k \oplus \mathcal{R}_k). \tag{3}$$

During inference, if $d_k = \texttt{<RESPONSE>}$, the model transitions from the "watching and thinking" state to the "responding" state and decodes the user-facing response A_k.

$$A_k = \texttt{LLM}_\theta(\mathcal{P}_{k-1} \oplus \mathcal{M}_{k-1} \oplus E_k \oplus \mathcal{R}_k \oplus \texttt{<RESPONSE>}). \tag{4}$$

If $d_k = \texttt{<SILENT>}$, the generation terminates for the current GoP and the model remains silent. In both cases, $\mathcal{R}_k$ is preserved in the memory by BESM (§4.3) for subsequent GoPs.

4.3 Brain-Eye Synergy Memory

Motivation. In streaming video understanding, bounded memory capacity limits long-horizon context. Existing work typically relies on heuristic frame pruning or coarse textual summarization [15, 21, 32, 52, 59, 64], but both have limitations. Aggressive pruning fragments temporal continuity, while coarse summarization strips away fine-grained visual details. The core challenge is to construct memory efficiently without sacrificing temporal continuity or visual fidelity.

Key Insight. Within a GoP, frames share most of their visual content, which is why P-frames can be predicted efficiently from preceding frames. Storing the visual tokens of all frames is therefore largely redundant. Meanwhile, the reasoning trace generated by DynThink already describes how the content evolves within the GoP. This trace can thus serve as a compact semantic substitute for the redundant frames. By anchoring this trace (*Brain*) to the visual tokens of the key frame (*Eye*), we construct a lightweight memory slot for each GoP that preserves long-horizon context over the unbounded stream.

Brain-Eye Synergy Memory. Based on this insight, we propose an asymmetric memory retention strategy at the GoP level. Specifically, let $E_k = \{e_{k,1}, \ldots, e_{k,N_k}\}$ denote the dense visual tokens of G_k, where $e_{k,i} = \Phi_v(v_{k,i})$ represents the tokens extracted from the i-th frame by the vision encoder Φ_v. For each processed GoP G_k, BESM writes one compact memory slot that combines two complementary components, (i) high-fidelity visual anchors, referred to as *Eye* tokens, and (ii) compact reasoning-derived summaries, referred to as *Brain* tokens. First, we retain the visual tokens $e_{k,1}$ of the key frame as the *Eye*, denoted as $\mathcal{I}_k$. Second, we discard the visual tokens of the remaining frames $\{e_{k,2}, \ldots, e_{k,N_k}\}$. Their temporal dynamics are instead represented by the reasoning trace $\mathcal{R}_k$ from DynThink, which serves as the *Brain*. The compact memory slot $\mathcal{S}_k$ is then formed, and the memory is updated as follows:

$$\mathcal{S}_k = \mathcal{I}_k \oplus \mathcal{R}_k, \tag{5}$$

$$\mathcal{M}_k = \mathcal{M}_{k-1} \oplus \mathcal{S}_k. \tag{6}$$

When the memory reaches its capacity, the oldest slot is evicted in a first-in-first-out manner. This design preserves both *what the model observes* and *how it interprets those observations*. The *Eye* tokens retain high-fidelity visual grounding for subsequent reasoning, while the *Brain* tokens summarize the temporal evolution of the GoP at a much lower cost than storing all its visual tokens. Moreover, BESM and DynThink share the same GoP-level temporal units. Memory formation is thus naturally aligned with the reasoning process that produces $\mathcal{R}_k$, which improves the consistency between online reasoning and long-horizon context accumulation.

5 Experiments

5.1 Experimental Setup

Datasets and Metrics. We evaluate **StreamMind** on both *online* streaming and *offline* long-form video understanding. For online streaming understanding, we use OVO-Bench [34] and Streaming-Bench [28]. OVO-Bench contains 12 subtasks across three temporal perception modes (real-time, backward, and forward). For StreamingBench, we evaluate on its Real-Time Visual Understanding tasks. For offline long-form video understanding, we evaluate StreamMind on VideoMME [12], LongVideoBench [51], and VideoHolmes [8]. We utilize accuracy as the principal evaluation metric to measure task performance.

Baselines. We compare **StreamMind** with a comprehensive set of state-of-the-art baselines in three categories. (1) **Open-source Online VLLMs.** This is our primary comparison group. It includes reasoning-enhanced streaming models (*e.g.*, ThinkStream [32] and VST [15]), other recent streaming models (*e.g.*, TimeChat-Online [59], StreamForest [61], and Streamo [54]), and earlier online VLLMs (*e.g.*, Dispider [36], Flash-VStream [63], and VideoLLM-online [4]). (2) **Open-source Offline VLLMs.** To assess general video understanding and long-context generalization, we include general-purpose models (*e.g.*, Qwen2.5-VL [2], LLaVA-OneVision [24], and LLaVA-Video [67]), long-video models (*e.g.*, LongVA [66] and LongVU [37]), and reasoning-centric models (*e.g.*, Video-R1 [11]). (3) **Proprietary Multimodal Models.** We further include GPT-4o [20], Gemini 1.5 Pro [42], and Claude 3.5 Sonnet [1] as performance references.

Implementation Details. We adopt Qwen2.5-VL [2] as the base model for both the 3B and 7B variants. The training data is based on LLaVA-Video [67] and Tarsier2-Recap-585K [60]. Following

Table 1: Evaluation results on OVO-Bench [34] comprising three categories: i) *Real-Time Visual Perception* (OCR: Optical Character Recognition, ACR: Action Recognition, ATR: Attribute Recognition, STU: Spatial Understanding, FPD: Future Prediction, OJR: Object Recognition), ii) *Backward Tracing* (EPM: Episodic Memory, ASI: Action Sequence Identification, HLD: Hallucination Detection), and iii) *Forward Active Responding* (REC: Repetition Event Count, SSR: Sequential Steps Recognition, CRR: Clues Reveal Responding).

Model	Venue	Real-Time							Backward				Forward				Overall
		OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.	REC	SSR	CRR	Avg.	Avg.
Human Agents	–	94.0	92.6	94.8	92.7	91.1	94.0	93.2	92.6	93.0	91.4	92.3	95.5	89.7	93.6	92.9	92.8
Proprietary Multimodal Models																	
Gemini 1.5 Pro [42]	–	85.9	67.0	79.3	58.4	63.4	62.0	69.3	58.6	76.3	52.6	62.5	35.5	74.2	61.7	57.1	63.0
GPT-4o [20]	–	69.8	64.2	71.5	51.1	70.3	59.8	64.5	57.9	75.7	48.7	60.8	27.6	73.2	59.4	53.4	59.5
Open-source Offline VLLMs																	
Qwen2-VL-72B [47]	–	65.8	60.5	69.8	51.7	69.3	54.4	61.9	52.5	60.8	57.5	57.0	38.8	64.1	45.0	49.3	56.1
LLaVA-Video-7B [67]	*TMLR'25*	69.1	58.7	68.8	49.4	74.3	59.8	63.4	56.2	57.4	7.5	40.4	34.1	70.0	60.4	54.8	52.9
LLaVA-OneVision-7B [24]	*TMLR'25*	66.4	57.8	73.3	53.4	71.3	62.0	64.0	54.2	55.4	21.5	43.7	25.6	67.1	58.8	50.5	52.7
Qwen2-VL-7B [47]	–	60.4	50.5	56.0	47.2	66.3	55.4	56.0	47.8	56.1	35.5	46.5	31.7	65.8	48.8	48.7	50.4
InternVL-V2-8B [7]	*CVPR'24*	67.1	60.5	63.8	46.1	68.3	56.5	60.4	48.1	57.4	24.7	43.4	26.5	59.1	54.1	46.6	50.1
LongVU-7B [37]	*ICML'25*	53.7	53.2	62.9	47.8	68.3	59.8	57.6	40.7	59.5	4.8	35.0	12.2	69.5	60.8	47.5	46.7
Open-source Online VLLMs																	
Flash-VStream-7B [63]	*ICCV'25*	24.2	29.4	28.4	33.7	25.7	28.8	28.4	39.1	37.2	5.9	27.4	8.0	67.2	60.0	45.1	33.6
VideoLLM-online-8B [4]	*CVPR'24*	8.1	23.9	12.1	14.0	45.5	21.2	20.8	22.2	18.8	12.2	17.7	–	–	–	–	–
Dispider-7B [36]	*CVPR'25*	57.7	49.5	62.1	44.9	61.4	51.6	54.5	48.5	55.4	4.3	36.1	18.1	37.4	48.8	34.7	41.8
TimeChat-Online-7B [59]	*MM'25*	75.2	46.8	70.7	47.8	69.3	61.4	61.9	55.9	59.5	9.7	41.7	31.6	38.5	40.0	36.7	46.8
Streamo-3B [54]	*CVPR'26*	78.5	52.3	67.2	44.4	55.5	71.2	61.5	51.2	57.4	16.7	41.8	27.9	50.7	82.5	53.7	52.3
Streamo-7B [54]	*CVPR'26*	79.2	57.8	75.0	49.4	64.4	70.1	66.0	54.6	52.0	31.7	46.1	30.0	51.0	83.3	**54.8**	55.6
ThinkStream-3B [32]	*ECCV'26*	85.2	64.2	69.8	49.4	69.3	64.1	67.0	53.9	59.5	43.6	52.3	17.3	42.6	47.9	35.9	51.7
VST-7B [15]	*ECCV'26*	80.5	55.1	72.4	55.1	76.2	64.1	67.2	56.9	64.9	48.4	56.7	33.0	66.9	62.1	54.0	59.3
StreamMind-3B (Ours)	–	81.9	62.4	68.1	46.1	70.3	60.9	65.0	48.8	56.8	83.9	63.2	16.1	59.5	58.3	44.6	57.6
StreamMind-7B (Ours)	–	87.2	60.6	70.7	51.1	68.3	67.4	**67.6**	49.2	63.5	99.5	**70.7**	17.6	65.8	72.1	51.8	**63.4**

ThinkStream [32], we convert the data into a unified streaming dialogue format and adopt the same supervised fine-tuning objective. To simulate streaming scenarios, all videos are transcoded with a low-delay configuration, and the codec metadata is extracted using FFmpeg. During fine-tuning, we freeze the vision encoder and update only the connector and the LLM. We train for one epoch with a batch size of 64 and a learning rate of 1×10^{-5}. All training experiments are conducted on a single node with 8 NVIDIA H800 GPUs.

5.2 Main Results on Streaming Video Benchmarks

Results on OVO-Bench. We evaluate StreamMind on OVO-Bench, which rigorously tests causal streaming capabilities across Real-Time Visual Perception, Backward Tracing, and Forward Active Responding. As shown in Table 1, StreamMind establishes a new state of the art, with StreamMind-7B achieving an overall accuracy of 63.4%. Notably, it outperforms strong online baselines such as ThinkStream-3B (51.7%) and VST-7B (59.3%). Furthermore, StreamMind clearly surpasses previous streaming models such as TimeChat-Online-7B (46.8%) and Streamo-7B (55.6%). In the Backward Tracing tasks (EPM, ASI, and HLD), StreamMind-7B achieves an average of 70.7%. This is primarily attributed to our Brain-Eye Synergy Memory. It preserves high-fidelity visual anchors and encapsulates temporal dynamics into compact reasoning traces, allowing the model to trace back to historical evidence. In Forward Active Responding, our model achieves a competitive 51.8%. This suggests that our Dynamic Thinking helps mitigate response flickering by accumulating visual evidence within each GoP.

Results on StreamingBench. Table 2 details the performance on StreamingBench, focusing on Real-Time Visual Understanding tasks. StreamMind-7B achieves the best overall score of 80.2%, pushing the boundaries of real-time understanding. It outperforms competitive models such as TimeChat-Online-7B (75.3%) and VST-7B (79.5%). On fine-grained perception tasks such as Object Perception (OP) and Attribute Perception (ATP), StreamMind-7B maintains highly competitive scores of 87.5% and 86.9%, respectively. This indicates that replacing the redundant dependent frames within a GoP with compact reasoning tokens does not degrade instantaneous visual reasoning. Instead, these tokens retain the semantics required for accurate online answering.

Table 2: Performance comparison on StreamingBench [28] focusing on *Real-Time Visual Understanding* tasks. Real-Time Visual Understanding encompasses Object Perception (OP), Causal Reasoning (CR), Clips Summarization (CS), Attribute Perception (ATP), Event Understanding (EU), Text-Rich Understanding (TR), Prospective Reasoning (PR), Spatial Understanding (SU), Action Perception (ACP), and Counting (CT). The **bold** values indicate the best performance.

Model	Venue	OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	Overall
Human	–	89.5	92.0	93.6	91.5	95.7	92.5	88.0	88.8	89.7	91.3	91.5
Proprietary Multimodal Models												
Gemini 1.5 Pro [42]	–	79.0	80.5	83.5	79.7	80.0	84.7	77.8	64.2	72.0	48.7	75.7
GPT-4o [20]	–	77.1	80.5	83.9	76.5	70.2	83.8	66.7	62.2	69.1	49.2	73.3
Claude 3.5 Sonnet [1]	–	73.3	80.5	84.1	82.0	75.4	79.5	61.1	61.8	69.3	43.1	72.4
Open-source Offline VLLMs												
Video-LLaMA2-7B [9]	*arXiv'24*	55.9	55.5	57.4	58.2	52.8	43.6	39.8	42.7	45.6	35.2	49.5
VILA-1.5-8B [27]	*CVPR'24*	53.7	49.2	71.0	56.9	53.4	53.9	54.6	48.8	50.1	17.6	52.3
Video-CCAM-14B [10]	*arXiv'24*	56.4	57.8	65.3	62.8	64.6	51.4	42.6	48.0	49.6	31.6	53.9
LongVA-7B [66]	*TMLR'25*	70.0	63.3	61.2	70.9	62.7	59.5	61.1	53.7	54.7	34.7	59.9
Kangaroo-7B [31]	*IJCV'26*	71.1	84.4	70.7	73.2	67.1	61.7	56.5	55.7	62.0	38.9	64.6
LLaVA-NeXT-Video-32B [29]	–	78.2	70.3	73.8	76.8	63.4	69.8	57.4	56.1	64.3	38.9	66.9
MiniCPM-V-2.6-8B [17]	*COLM'24*	71.9	71.1	77.9	75.8	64.6	65.7	70.4	56.1	62.3	53.4	67.4
LLaVA-OneVision-7B [24]	*TMLR'25*	80.4	74.2	76.0	80.7	72.7	71.7	67.6	65.5	65.7	45.1	71.1
Qwen2.5-VL-7B [2]	–	78.3	80.5	78.9	80.5	76.7	78.5	79.6	63.4	66.2	53.2	73.7
Open-source Online VLLMs												
Flash-VStream-7B [63]	*ICCV'25*	25.9	43.6	24.9	23.9	27.3	13.1	18.5	25.2	23.9	48.7	23.2
VideoLLM-online-8B [4]	*CVPR'24*	39.1	40.1	34.5	31.1	46.0	32.4	31.5	34.2	42.5	27.9	35.9
Dispider-7B [36]	*CVPR'25*	74.9	75.5	74.1	73.1	74.4	59.9	76.1	62.9	62.2	45.8	67.6
TimeChat-Online-7B [59]	*MM'25*	80.2	82.0	79.5	83.3	76.1	78.5	78.7	64.6	69.6	58.0	75.3
StreamForest-7B [61]	*NeurIPS'25*	83.1	82.8	82.7	84.3	77.5	78.2	76.9	69.1	75.6	54.4	77.3
ThinkStream-3B [32]	*ECCV'26*	83.7	70.3	76.0	82.6	76.7	78.1	76.8	70.7	74.4	45.2	75.0
VST-7B [15]	*ECCV'26*	85.4	82.0	86.4	89.1	74.2	87.2	82.4	73.1	73.9	47.3	79.5
StreamMind-3B (Ours)	–	82.9	79.7	79.8	82.4	74.8	76.6	80.6	68.3	73.3	49.5	75.6
StreamMind-7B (Ours)	–	87.5	82.0	85.2	86.9	78.0	85.4	83.3	74.4	76.4	50.5	**80.2**

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

Model	Venue	VideoMME w/o sub.		LongVideoBench	VideoHolmes
		Long	Overall		
Proprietary Multimodal Models					
Gemini 1.5 Pro [42]	–	67.4	75.0	64.0	45.7
GPT-4o [20]	–	65.3	71.9	66.7	42.0
Open-source Offline VLLMs					
LongVA-7B [66]	*TMLR'25*	47.6	54.3	56.3	–
Video-R1-7B [11]	*NeurIPS'25*	–	61.4	–	36.5
LongVILA-R1-7B [6]	*NeurIPS'25*	55.2	65.1	58.0	–
REVISOR-7B [25]	*CVPR'26*	56.2	65.7	57.5	–
Open-source Online VLLMs					
Dispider-7B [36]	*CVPR'25*	–	57.2	–	–
StreamForest-7B [61]	*NeurIPS'25*	–	61.4	–	–
TimeChat-Online-7B [59]	*MM'25*	48.4	62.4	55.4	–
ThinkStream-3B [32]	*ECCV'26*	–	61.9	56.4	–
VST-7B [15]	*ECCV'26*	55.3	64.9	58.0	41.9
StreamMind-3B (Ours)	–	54.6	63.1	57.2	39.4
StreamMind-7B (Ours)	–	**57.1**	**66.4**	**59.3**	**43.4**

5.3 Results on Offline Video Benchmarks

To verify that our dynamic streaming architecture does not compromise offline reasoning capabilities, we evaluate StreamMind on three offline benchmarks, namely VideoMME (without subtitles), LongVideoBench, and VideoHolmes. As shown in Table 3, StreamMind-7B achieves 66.4% on VideoMME (57.1% on long videos) and 59.3% on LongVideoBench, although it is optimized for strict causal streaming. Our streaming-centric design thus transfers well to offline long-form video understanding. On LongVideoBench, StreamMind-7B surpasses offline models such as LongVA-7B (56.3%) and reasoning-enhanced models such as LongVILA-R1-7B (58.0%). Compared with other

Table 4: Ablation study on OVO-Bench. **DynThink**: Dynamic Thinking module. **BESM**: Brain-Eye Synergy Memory module. $\mathcal{I}$ and $\mathcal{R}$ denote the high-fidelity visual anchors (*Eye*) and compact reasoning traces (*Brain*), respectively.

DynThink	BESM	Real-Time							Backward				Forward				Overall
		OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.	REC	SSR	CRR	Avg.	Avg.
		67.7	55.1	67.2	42.1	66.3	60.8	59.9	51.5	58.7	23.6	44.6	32.5	49.6	40.8	40.9	48.4
✔		78.5	58.2	69.1	46.8	67.5	64.2	64.1	50.4	60.2	28.5	46.4	22.4	60.5	61.5	48.1	52.8
✔	$\mathcal{I}$ (*Eye*)	82.1	48.4	68.2	45.1	58.3	65.5	61.3	45.2	41.6	35.4	40.7	18.2	52.4	55.1	41.9	48.0
✔	$\mathcal{R}$ (*Brain*)	55.4	56.1	52.3	38.6	64.5	42.1	51.5	48.1	59.5	60.2	55.9	20.1	62.1	65.4	49.2	52.2
✔	$\mathcal{I} \oplus \mathcal{R}$	87.2	60.6	70.7	51.1	68.3	67.4	**67.6**	49.2	63.5	99.5	**70.7**	17.6	65.8	72.1	**51.8**	**63.4**

online VLLMs, StreamMind-7B consistently maintains a performance lead. For instance, it scores 43.4% on VideoHolmes, outperforming VST-7B (41.9%). This suggests that converting offline videos into a sequence of temporally structured GoP memory slots allows StreamMind to act as an efficient context compressor. Unlike static frame pruning, this compression summarizes the discarded frames into reasoning traces rather than simply dropping them.

5.4 Ablation Study

As summarized in Table 4, we evaluate the individual contributions of our core components on OVO-Bench [34].

Effectiveness of DynThink. Compared to the baseline, integrating DynThink improves the overall accuracy from 48.4% to 52.8%. Notably, the **Forward Active Responding** score rises markedly from 40.9% to 48.1%. This suggests that triggering reasoning at GoP boundaries helps mitigate response flickering and improve interaction stability. In this way, the model relies on accumulated visual evidence rather than transient frame-level cues.

Internal Design of BESM: *Brain* vs. *Eye*. To further investigate the synergy between *Brain* and *Eye*, we evaluate two variants of BESM. (1) **Only *Eye* ($\mathcal{I}$):** Storing only the key-frame tokens as visual anchors reduces the overall accuracy from 52.8% (DynThink only) to 48.0%, even below the baseline (48.4%). This suggests that static visual anchors alone cannot capture the temporal evolution within a GoP. Action-related tasks thus degrade severely (*e.g.*, ACR and ASI). (2) **Only *Brain* ($\mathcal{R}$):** Relying solely on reasoning traces (52.2%) also slightly underperforms the DynThink-only variant. Without high-fidelity visual anchors, the model loses grounding in fine-grained details. This leads to a sharp decline in OCR and Object Recognition (OJR). These results show that streaming memory must provide both visual fidelity and temporal semantics. Otherwise, partial memory can act as noise and hinder understanding.

Synergy in StreamMind. The full framework, which couples both components ($\mathcal{I} \oplus \mathcal{R}$), achieves the best overall accuracy of 63.4%. The synergy is most prominent in **Backward Tracing**, where the average accuracy rises from 46.4% to 70.7%. DynThink grounds the *Brain* tokens in complete GoP-level evidence. BESM then efficiently stores these traces alongside the *Eye* anchors for subsequent reasoning. This tight coupling allows StreamMind to maintain temporal continuity and reasoning stability in long-horizon online video understanding.

6 Conclusion

In this paper, we present **StreamMind**, a dynamic streaming cognition framework that shifts online video understanding from rigid temporal units to dynamic, content-adaptive ones. By aligning online reasoning and memory formation with the inherent GoP structure, StreamMind mitigates flickering responses and fragmented context. Our **DynThink** module improves the stability of proactive interaction by reasoning only at GoP boundaries. The **BESM** module preserves long-horizon context through the synergy of high-fidelity visual anchors (*Eye*) and compact reasoning traces (*Brain*). Extensive experiments on streaming and offline video benchmarks demonstrate that StreamMind achieves state-of-the-art online video understanding performance. We hope this work provides a robust foundation for future real-time multimodal assistants in complex, dynamic environments.

Acknowledgments and Disclosure of Funding

This research is funded in part by ARC-Discovery grant (DP220100800, DE230100477, FT260100691 to XY). We thank all anonymous reviewers and ACs for their constructive suggestions.

References

[1] Anthropic. Claude 3.5 sonnet. `https://www.anthropic.com/news/claude-3-5-sonnet`, 2024. Accessed: 2026-02-11.

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[3] Brandon Castellano. PySceneDetect. `https://github.com/Breakthrough/PySceneDetect`.

[4] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18407–18418, 2024.

[5] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Zhenyu Tang, Li Yuan, et al. Sharegpt4video: Improving video understanding and generation with better captions. *NeurIPS*, 2024.

[6] Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, XIAOJUAN QI, et al. Scaling rl to long videos. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[7] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024.

[8] Junhao Cheng, Yuying Ge, Teng Wang, Yixiao Ge, Jing Liao, and Ying Shan. Video-holmes: Can mllm think like holmes for complex video reasoning? *arXiv preprint arXiv:2505.21374*, 2025.

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

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

[11] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in MLLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[12] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24108–24118, 2025.

[13] Haonan Ge, Yiwei Wang, Kai-Wei Chang, Hang Wu, and Yujun Cai. Famemind: Frame-interleaved video reasoning via reinforcement learning. *arXiv e-prints*, pages arXiv–2509, 2025.

[14] Sara Ghazanfari, Francesco Croce, Nicolas Flammarion, Prashanth Krishnamurthy, Farshad Khorrami, and Siddharth Garg. Chain-of-frames: Advancing video understanding in multimodal llms via frame-aware reasoning. *arXiv preprint arXiv:2506.00318*, 2025.

[15] Yiran Guan, Liang Yin, Dingkang Liang, Jianzhong Ju, Zhenbo Luo, Jian Luan, Yuliang Liu, and Xiang Bai. Video streaming thinking: Videollms can watch and think simultaneously. In *European Conference on Computer Vision*, pages 80–99. Springer, 2026.

[16] Vaishnavi Himakunthala, Andy Ouyang, Daniel Rose, Ryan He, Alex Mei, Yujie Lu, Chinmay Sonar, Michael Saxon, and William Wang. Let's think frame by frame with VIP: A video infilling and prediction dataset for evaluating video chain-of-thought. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 204–219, Singapore, December 2023. Association for Computational Linguistics.

[17] Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Xinrong Zhang, Zhen Leng Thai, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, dahai li, Zhiyuan Liu, and Maosong Sun. MiniCPM: Unveiling the potential of small language models with scalable training strategies. In *First Conference on Language Modeling*, 2024.

[18] Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A Smith, and Jiebo Luo. Promptcap: Prompt-guided task-aware image captioning. *arXiv preprint arXiv:2211.09699*, 2022.

[19] 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3328–3338, 2025.

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

[21] Hongxiang Jiang, Zengrui Ge, guo chen, Qixiong Wang, Jile Jiao, Xuetao Feng, Yuan Wang, and Yan Wang. Memento: Toward an all-day proactive assistant for ultra-long streaming video. In *The Fourteenth International Conference on Learning Representations*, 2026.

[22] Ivan Kirillov, Denis Parkhomenko, Kirill Chernyshev, Alexander Pletnev, Yibo Shi, Kai Lin, and Dmitry Babin. Hierarchical b-frame video coding for long group of pictures. *arXiv preprint arXiv:2406.16544*, 2024.

[23] Didier Le Gall. Mpeg: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58, 1991.

[24] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025.

[25] Jiaze Li, Hao Yin, Wenhui Tan, Jingyang Chen, Boshen Xu, Yuxun Qu, Yijing Chen, Jianzhong Ju, Zhenbo Luo, and Jian Luan. Revisor: Beyond textual reflection, towards multimodal introspective reasoning in long-form video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2026.

[26] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5971–5984, 2024.

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

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

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

[30] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[31] Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, Xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. Kangaroo: A powerful video-language model supporting long-context video input: J. liu et al. *International Journal of Computer Vision*, 134(3):114, 2026.

[32] Zikang Liu, Longteng Guo, Handong Li, Ru Zhen, Xingjian He, Ruyi Ji, Xiaoming Ren, Yanhao Zhang, Haonan Lu, and Jing Liu. Thinking in streaming video. In *European Conference on Computer Vision*, pages 21–38. Springer, 2026.

[33] Jinghui Lu, Haiyang Yu, Siliang Xu, Shiwei Ran, Guozhi Tang, Siqi Wang, Bin Shan, Teng Fu, Hao Feng, Jingqun Tang, et al. Prolonged reasoning is not all you need: Certainty-based adaptive routing for efficient llm/mllm reasoning. *arXiv preprint arXiv:2505.15154*, 2025.

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

[35] Reza Pourreza and Taco Cohen. Extending neural p-frame codecs for b-frame coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6680–6689, 2021.

[36] 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.

[37] Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. In *42st International Conference on Machine Learning, ICML 2025*. ML Research Press, 2025.

[38] Xin Shen, Rui Zhu, Lei Shen, Zhuojie Wu, Xinyu Wang, Kaihao Zhang, Tianqing Zhu, Shuchen Wu, Chenxi Miao, Weikang Li, et al. Fingercap: Fine-grained finger-level hand motion captioning. In *European Conference on Computer Vision*, pages 531–550. Springer, 2026.

[39] Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, et al. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *arXiv preprint arXiv:2506.23918*, 2025.

[40] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.

[41] Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29118–29128. IEEE, 2025.

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

[43] Junlong Tong, Yingqi Fan, Anhao Zhao, Yunpu Ma, and Xiaoyu Shen. Streamingthinker: Large language models can think while reading. In *The Fourteenth International Conference on Learning Representations*, 2026.

[44] Junlong Tong, Zilong Wang, YuJie Ren, Peiran Yin, Hao Wu, Wei Zhang, and Xiaoyu Shen. From static inference to dynamic interaction: A survey of streaming large language models. *arXiv preprint arXiv:2603.04592*, 2026.

[45] Boyang Wang, Guangyi Xu, Jiahui Zhang, Zhipeng Tang, and Zezhou Cheng. Omnishotcut: Holistic relational shot boundary detection with shot-query transformer. *arXiv preprint arXiv:2604.24762*, 2026.

[46] 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*, 2025.

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

[48] Yan Wang, Yawen Zeng, Jingsheng Zheng, Xiaofen Xing, Jin Xu, and Xiangmin Xu. VideoCoT: A video chain-of-thought dataset with active annotation tool. In Jing Gu, Tsu-Jui (Ray) Fu, Drew Hudson, Asli Celikyilmaz, and William Wang, editors, *Proceedings of the 3rd Workshop on Advances in Language and Vision Research (ALVR)*, pages 92–101, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[49] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, pages 396–416. Springer, 2024.

[50] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003.

[51] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

[52] Zhuojie Wu, Xingqun Qi, Zijian Wang, Wanting Zhou, Kun Yuan, Muyi Sun, and Zhenan Sun. Showface: Coordinated face inpainting with memory-disentangled refinement networks. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022.

[53] Zhuojie Wu, Shijie Wang, and Xin Yu. Metom: Metadata-guided token merging for efficient video llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10441–10450, 2026.

[54] Jiaer Xia, Peixian Chen, Mengdan Zhang, Xing Sun, and Kaiyang Zhou. Streaming video instruction tuning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2026.

[55] Kun Xiang, Zhili Liu, Zihao Jiang, Yunshuang Nie, Kaixin Cai, Yiyang Yin, Runhui Huang, Haoxiang Fan, Hanhui Li, Weiran Huang, et al. Can atomic step decomposition enhance the self-structured reasoning of multimodal large models? *arXiv preprint arXiv:2503.06252*, 2025.

[56] Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. Streamingvlm: Real-time understanding for infinite video streams. *arXiv preprint arXiv:2510.09608*, 2025.

[57] Yanlai Yang, Zhuokai Zhao, Satya Narayan Shukla, Aashu Singh, Shlok Kumar Mishra, Lizhu Zhang, and Mengye Ren. Streammem: Query-agnostic kv cache memory for streaming video understanding. *arXiv preprint arXiv:2508.15717*, 2025.

[58] 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*, 2026.

[59] 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.

[60] Liping Yuan, Jiawei Wang, Haomiao Sun, Yuchen Zhang, and Yuan Lin. Tarsier2: Advancing large vision-language models from detailed video description to comprehensive video understanding. *arXiv preprint arXiv:2501.07888*, 2025.

[61] Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, et al. Streamforest: Efficient online video understanding with persistent event memory. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[62] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 conference on empirical methods in natural language processing: system demonstrations*, pages 543–553, 2023.

[63] Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-vstream: Efficient real-time understanding for long video streams. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 21059–21069. IEEE, 2025.

[64] Jialiang Zhang, Junlong Tong, Junyan Lin, Hao Wu, Yirong Sun, Yunpu Ma, and Xiaoyu Shen. Think-as-you-see: Streaming chain-of-thought reasoning for large vision-language models. *arXiv preprint arXiv:2603.02872*, 2026.

[65] Kairui Zhang, Zhenyu Yang, Bing Wang, Shengsheng Qian, and Changsheng Xu. Querystream: Advancing streaming video understanding with query-aware pruning and proactive response. In *The Fourteenth International Conference on Learning Representations*, 2026.

[66] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *Transactions on Machine Learning Research*, 2025.

[67] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun MA, Ziwei Liu, and Chunyuan Li. LLaVA-video: Video instruction tuning with synthetic data. *Transactions on Machine Learning Research*, 2025.

[68] Anhao Zhao, Ziyang Chen, Junlong Tong, Yingqi Fan, Fanghua Ye, Shuhao Li, Yunpu Ma, Wenjie Li, and Xiaoyu Shen. On-policy supervised fine-tuning for efficient reasoning. *arXiv preprint arXiv:2602.13407*, 2026.

This appendix is organized as follows.

- Additional Implementation Details (§A).
- Additional Ablation Studies (§B).
- Runtime Analysis (§C).
- Limitations and Future Work (§D).
- Broader Impacts (§E).

A Additional Implementation Details

GoP Configuration. By default, we transcode videos with H.264 at a constant rate factor (CRF) of 23 under a low-delay configuration without B-frames. The encoder starts a new GoP when it detects a scene cut or reaches the 250-frame limit. We preserve the source frame rate during transcoding and sample frames at 2 fps for the model. On OVO-Bench, the resulting GoPs average 5.07 s.

Memory Configuration. For each GoP G_k, BESM stores a memory slot $\mathcal{S}_k = \mathcal{I}_k \oplus \mathcal{R}_k$, where $\mathcal{I}_k$ contains the key-frame visual tokens and $\mathcal{R}_k$ is the reasoning trace. We retain up to 120 GoP slots, matching the chunk budget in ThinkStream [32]. When the memory reaches its capacity, the oldest slot is evicted in a first-in-first-out manner. Slots are never merged across GoPs. We cap each reasoning trace at 20 tokens. On OVO-Bench, the traces average 15.0 tokens per GoP.

Training Supervision. To construct the training targets, we group ThinkStream's timestamped reasoning steps by GoP and concatenate them in chronological order. For each sample, we label the GoP that contains the annotated response time with `<RESPONSE>` and append the ground-truth answer. We label all other GoPs with `<SILENT>`. We train the model to predict the reasoning trace, the decision token, and any answer tokens using the standard next-token prediction objective.

B Additional Ablation Studies

Sensitivity to GoP and Encoding Settings. Table A1 evaluates StreamMind under different GoP and encoding settings, with all model and sampling settings unchanged. Limiting the maximum GoP duration to 3, 5, or 7 s changes the overall accuracy by only 0.3%–0.5%. The 3 s limit also serves as a stress test. It forces boundaries that are not triggered by content changes and may therefore split ongoing events. Enabling B-frames, changing the CRF, or switching to HEVC changes the overall accuracy by at most 1.0%. Since CRF values are codec-specific, the HEVC setting uses its default CRF of 28. It thus compares the two codecs at their default operating points rather than isolating the codec. Overall, StreamMind remains stable across the tested GoP and encoding settings.

Table A1: Sensitivity to GoP and encoding settings.

Setting	**Avg. GoP (s)**	**Overall**	**Δ**
Default (scene-cut, max. 250 frames)	5.07	63.4	–
Scene-cut, max. 3 s	2.74	62.9	-0.5
Scene-cut, max. 5 s	4.16	63.1	-0.3
Scene-cut, max. 7 s	4.82	63.0	-0.4
B-frames enabled	5.33	62.8	-0.6
CRF 18	5.02	63.5	+0.1
CRF 28	5.12	62.4	-1.0
HEVC (CRF 28)	5.18	62.7	-0.7

Ablation Study on Temporal Units. Table A2 compares GoPs with alternative temporal units under the same model and evaluation settings. The fixed-window and frame-level variants also share the same backbone, training data, 2 fps sampling rate, token budget, memory budget, and training procedure. Fixed windows reduce the overall accuracy by 1.5%–2.5%, since their uniform boundaries cannot adapt to content changes. The frame-level variant updates at every sampled frame. It fills the

memory budget more quickly and makes decisions from single frames. Both effects hurt Backward Tracing and Forward Active Responding. Adaptive frame selection keeps only sparse frames and provides less continuous evidence. Scene and shot boundaries come closer to GoPs (62.7% and 62.1% vs. 63.4%), but both require an external detector. Overall, content-adaptive boundaries outperform uniform ones. GoPs perform best among all units and come directly from codec metadata.

Table A2: Ablation study on temporal units.

Temporal unit	Boundary source	Avg. interval (s)	Real-Time	Backward	Forward	Overall	Δ
Fixed window	Uniform, 3 s	3.00	65.9	69.1	49.3	61.4	-2.0
Fixed window	Uniform, 5 s	5.00	66.4	69.6	49.8	61.9	-1.5
Fixed window	Uniform, 7 s	7.00	65.1	68.8	48.7	60.9	-2.5
Frame-level	Every sampled frame	0.50	65.9	62.7	45.2	57.9	-5.5
Scene boundary	PySceneDetect [3]	5.62	67.0	70.1	51.0	62.7	-0.7
Shot boundary	OmniShotCut [45]	3.84	66.6	69.7	50.0	62.1	-1.3
Adaptive frame	AKS [41]	–	65.8	68.2	47.1	60.4	-3.0
GoP (Ours)	Codec metadata	5.07	**67.6**	**70.7**	**51.8**	**63.4**	–

C Runtime Analysis

Table A3 compares StreamMind with a GoP variant without reasoning and a fixed 5 s variant with reasoning. All variants use the same model, hardware, and 20-token reasoning limit. We report the average model-side wall-clock latency per update. Each update operates on an evidence unit, *i.e.*, the segment over which visual evidence is accumulated (a GoP or a 5 s window). Streaming-step latency is the time taken to generate the reasoning trace and decision token once the evidence unit is fully received. Response end-to-end (E2E) latency additionally covers response generation. The time spent waiting for the unit boundary depends on when the query arrives and is thus not counted as inference overhead.

Generating the reasoning trace adds 0.40 s per update over the GoP variant without reasoning. Even so, the response E2E latency of StreamMind (1.25 s) stays well below the average GoP duration (5.07 s), so StreamMind keeps pace with the incoming stream on average. Fixed 5 s windows and GoPs have similar average lengths (5.00 vs. 5.07 s), so their latency gap mainly comes from the trace length. StreamMind produces shorter traces (15.0 vs. 18.4 tokens), likely because each GoP covers a coherent segment. A fixed window may instead cut through or straddle events and thus need more tokens to describe incomplete or mixed evidence. Shorter traces lead to lower latency (1.25 vs. 1.34 s) and higher throughput (0.97 vs. 0.89 updates/s). GoP-level reasoning also achieves higher overall accuracy (63.4% vs. 61.9%), so it is both faster and more accurate than the fixed 5 s variant.

Table A3: Runtime and reasoning cost analysis. Overhead is relative to GoP w/o reasoning.

Method	Reasoning tokens	Streaming-step latency (s)	Response E2E latency (s)	Overhead (s)	Throughput (updates/s)
GoP w/o reasoning	0	0.63	0.85	–	1.59
Fixed 5 s w/ reasoning	18.4	1.12	1.34	+0.49	0.89
StreamMind	15.0	1.03	1.25	+0.40	0.97

D Limitations and Future Work

A current limitation of StreamMind is that it focuses on visual-only streaming understanding. While GoP-level reasoning provides an effective temporal structure for online visual streams, the framework does not yet model auditory cues such as speech, sound events, or background audio, which can be important for understanding real-world streaming scenarios. Future work will extend StreamMind toward omni-modal streaming understanding by jointly modeling visual and audio signals within a unified real-time reasoning and memory framework.

E Broader Impacts

StreamMind may benefit real-time multimodal assistants in scenarios such as education, accessibility, robotics, and interactive AI, where stable and context-aware video understanding is important. By improving streaming reasoning and long-horizon context modeling, this work can support more responsive and reliable human-AI interaction in dynamic environments.

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: The abstract and introduction clearly state the paper's scope, main motivation, and contributions, including GoP-aligned Dynamic Thinking and Brain-Eye Synergy Memory for online video understanding. The claims are supported by the experimental results on streaming and offline video benchmarks in Sec. 5.

   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: The paper discusses its current visual-only scope in Appendix D and identifies omni-modal streaming understanding with audio and speech as 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: The paper does not present theoretical results, formal theorems, or proofs. The mathematical formulations are used to describe the proposed streaming reasoning and memory mechanisms.

   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: The paper describes the model architecture, training data, streaming simulation protocol, evaluation benchmarks, metrics, and implementation details in Sec. 4 and Sec. 5.1, which provide the main information needed to reproduce the reported experiments.

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: The paper does not currently provide open access to code or checkpoints. However, it uses publicly available datasets and models, and provides implementation details to support reproducibility.

   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: Sec. 5.1 specifies the evaluated benchmarks, baselines, base model, training data, streaming format construction, codec metadata extraction, trainable modules, training epoch, batch size, and learning rate.

   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: The paper reports benchmark accuracy results and ablation studies but does not include error bars or statistical significance tests. This is mainly due to the high computational cost of repeatedly training and evaluating large video language models.

   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: Sec. 5.1 reports the main training configuration and compute resources.

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: The research uses existing public datasets, models, and benchmarks for video understanding and does not involve new human-subject studies. We believe the work conforms to the NeurIPS Code of Ethics.

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: [Yes]

Justification: Appendix E discusses potential positive broader impacts of StreamMind, including applications in education, accessibility, robotics, and interactive AI systems.

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 does not introduce a new dataset or release a high-risk model or asset in the current submission. Therefore, additional release safeguards are not applicable.

    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 paper properly cites the existing models, datasets, and benchmarks used in the experiments, including Qwen2.5-VL, LLaVA-Video, Tarsier2Recap, OVO-Bench, StreamingBench, VideoMME, LongVideoBench, and VideoHolmes. These assets are used according to their intended research purposes.

    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: [N/A]

Justification: The paper does not introduce or release a new dataset, benchmark, or external asset. The contribution is a method and framework for streaming video understanding.

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 experiments or new 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 human-subject experiments, user studies, or crowdsourced data collection, so IRB approval is not applicable.

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: [N/A]

Justification: LLMs were utilized solely for editorial tasks, such as grammar correction, spelling, and formatting. They did not contribute to the development of core methodologies, data analysis, or the scientific content of the paper.

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.