StreamRAG: Enhancing Real-Time Video Understanding with Retrieval Augmentation

Junlin Xie[1,2] Quanlong Zheng[3,†] Ruifei Zhang[1,2] Kuo Wang[4] Yanhao Zhang[3,†]
Jinguo Luo[5] Haonan Lu[3] Xiang Wan[2] Guanbin Li[4,6,7,*]

[1]The Chinese University of Hong Kong, Shenzhen [2]Shenzhen Research Institute of Big Data
[3]OPPO AI Center, OPPO Inc., China [4]Sun Yat-sen University
[5]Harbin Institute of Technology, Shenzhen [6]Shenzhen Loop Area Institute
[7]Guangdong Key Laboratory of Big Data Analysis and Processing
junlinxie@link.cuhk.edu.cn, liguanbin@mail.sysu.edu.cn

Abstract

Retrieval-Augmented Generation (RAG) has shown considerable promise in offline video comprehension; however, its application to streaming video remains relatively unexplored. Streaming video introduces unique challenges, such as continuous data influx, temporal sensitivity, and stringent latency requirements. Key obstacles in deploying RAG for streaming video include: (1) the necessity for adaptive semantic segmentation to enable real-time boundary detection; (2) the challenge of balancing latency and accuracy in knowledge extraction; and (3) the complexity of handling queries with varying degrees of temporal sensitivity. To address these issues, we present StreamRAG, an innovative framework designed for streaming video question answering. StreamRAG integrates: (1) a Stream Event Segmentation (SES) module that divides video streams into semantically coherent events; (2) a knowledge extraction accelerator that minimizes captioning latency by reusing previously processed tokens; and (3) a query-aware dynamic knowledge injection module that optimizes retrieval based on the temporal sensitivity of queries and similarity scoring. Experimental results demonstrate that StreamRAG significantly enhances the efficiency of real-time video comprehension while maintaining a balance between accuracy and responsiveness.

1. Instruction

The remarkable capabilities of Large Language Models (LLMs) [1, 16, 24] have been successfully extended to the visual domain, giving rise to powerful video-language models [4, 12–14, 26, 28, 30, 32, 38–40] that achieve impressive performance on video understanding tasks. However, their inherent limitations—fixed parametric knowledge and restricted context windows—pose significant challenges for long-video comprehension. Retrieval-Augmented Generation (RAG) [2, 6, 11] offers a principled solution by constructing external knowledge bases and integrating retrieved multimodal evidence on demand, thereby enabling efficient long-context modeling, mitigating hallucinations through evidence grounding, and supporting flexible domain adaptation without retraining. However, existing RAG approaches are predominantly designed for offline video understanding, where the complete video is available before processing. Streaming video scenarios, by contrast, introduce a fundamentally different set of challenges: (1) continuous data inflow with unbounded temporal extent, (2) stringent low-latency requirements for real-time response generation, and (3) temporal sensitivity in query resolution, as answers may depend on specific moments within an ever-growing stream.

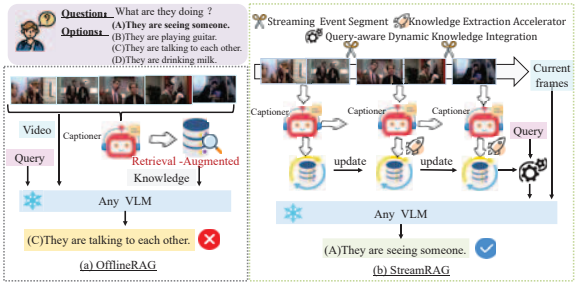

Figure 1. Architecture comparison of OfflineRAG and StreamRAG. (a) OfflineRAG constructs a static knowledge base from complete videos, lacking incremental updates and struggling with time-sensitive queries. (b) StreamRAG employs a three-module framework that supports dynamic knowledge base construction, temporal-aware retrieval, and low-latency response generation.

† Project Leader * Corresponding Author

These characteristics render offline video-RAG pipelines ill-suited for online streaming settings[5, 21, 36, 41].

To bridge these gaps and enable effective streaming RAG, we believe that an ideal RAG approach should address the following several critical bottlenecks: (1) **Determining the timing for database updates**. In streaming scenarios, an effective online database update strategy must detect meaningful boundaries in real time to trigger updates aligned with the evolving video stream, while avoiding excessive fragmentation or intolerable delays. This calls for adaptive streaming event segmentation algorithms capable of inferring contextual shifts on the fly without ground-truth annotations. (2) **Minimizing Knowledge Extraction Latency.** Most offline RAG systems [23, 31] rely on external captioning models to produce detailed textual descriptions for enriching the knowledge base, yet such pipelines inevitably introduce substantial computational overhead that exacerbates response delays in streaming settings. An efficient knowledge extraction mechanism that balances descriptive richness with processing speed is therefore essential. (3) **Dynamic Regulation of Retrieval and Fusion Granularity.** User queries exhibit varying degrees of temporal sensitivity—some target the immediate video context while others require reasoning over long-range history—a distinction that conventional similarity-based retrieval methods handle poorly. Dynamically adjusting retrieval scope and fusion granularity according to query temporality remains an open problem. These challenges collectively highlight the need for an efficient yet expressive framework that incrementally constructs a RAG knowledge database from video streams, enabling both precise understanding and real-time retrieval to optimize query responses across different temporal scales.

To this end, we propose **StreamRAG**, a three-module framework that brings RAG to streaming video comprehension: First, to tackle the "when-to-update" dilemma, we introduce the **Stream Event Segmentation** module, which continuously monitors incoming video streams and intelligently partitions them into semantically coherent events while preserving their inherent causal relationships. Secondly, we propose a **Knowledge Extraction Accelerator** that reuses tokens from previous event captions to expedite caption generation. This approach accelerates the knowledge extraction process, thereby effectively mitigating the issue of high latency. Finally, to better adapt to queries with varying temporal sensitivities, we introduce a **Query-aware Dynamic Knowledge Injection** module. This module combines the LLM's assessment of query "instantaneity" with a joint scoring mechanism based on similarity to RAG knowledge, ultimately enabling optimized knowledge selection with focus on different temporal segments of the video stream. Designed as a plug-and-play solution, our framework seamlessly integrates with existing mllms, enhancing their streaming capabilities without architectural modifications. Extensive evaluations demonstrate state-of-the-art performance on two major benchmarks, achieving up to 11% accuracy gains. These results establish a new paradigm for real-time video understanding systems by simultaneously improving both accuracy and processing speed.

2. Related Work

Streaming Video Understanding. Early efforts extend offline Video-LLMs to the streaming setting through frame-level sampling strategies [9, 19, 20, 29, 34]. Subsequent work improves efficiency by compressing redundant visual tokens [35] or introducing memory buffers with decayed compression for lightweight offline-to-online adaptation [25]. On the interaction side, ViSpeak [7] formulates visual instruction feedback across seven fundamental tasks, while Dispider [22] disentangles continuous perception from responsive interaction via parallel processing. Despite these advances, existing methods remain bounded by their parametric knowledge and fixed context windows. StreamRAG takes a complementary perspective: instead of refining the model itself, we dynamically construct and retrieve an external knowledge base aligned with the evolving video stream, enabling temporally grounded comprehension without sacrificing real-time responsiveness.

RAG for Video Understanding. Retrieval-Augmented Generation (RAG) [8, 37], originally developed for text-based tasks, has been increasingly adopted for long-form video comprehension [10, 23]. Rather than processing entire videos exhaustively, RAG-based approaches selectively retrieve informative segments, effectively alleviating temporal redundancy and enabling efficient long-range reasoning. Existing methods differ primarily in retrieval strategy. DrVideo [18] converts long videos into textual documents and employs an agent-based iterative loop to locate question-relevant keyframes. Goldfish [3] retrieves the top-k clips most relevant to a given query via caption-level semantic matching, scaling to arbitrarily long videos. While effective, both rely solely on textual representations and may miss fine-grained visual cues. VideoRAG [17] mitigates this by incorporating multimodal auxiliary signals (*e.g.*, OCR, scene graphs) during indexing, improving retrieval precision at the cost of heavier computation. AdaRAG [33] further introduces a lightweight intent classifier that adaptively selects retrieval schemes according to query complexity, balancing accuracy and efficiency. However, the above methods all operate on static knowledge bases built from complete videos. The closest work to ours is StreamChat [31], which maintains a hierarchical memory that is progressively updated during streaming. Yet its

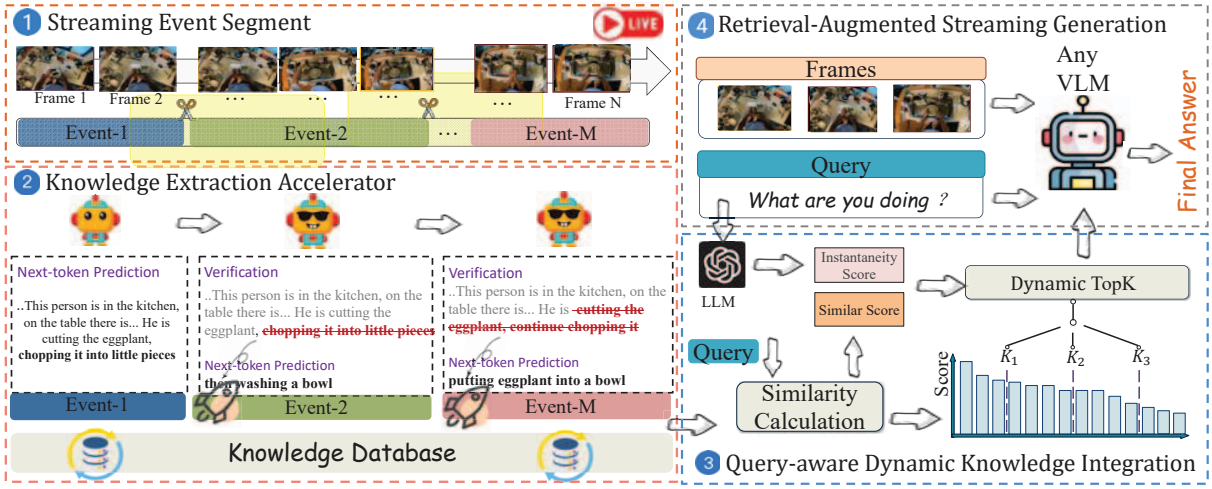

Figure 2. Overview of the StreamRAG framework. StreamRAG incrementally constructs a streaming knowledge base via three modules: (1) *Streaming Event Segmentation* partitions the video stream into semantically coherent events; (2) *Knowledge Extraction Accelerator* performs low-latency knowledge extraction from each event; (3) *Query-aware Dynamic Knowledge Integration* retrieves and fuses temporally relevant knowledge for real-time video qa.

fixed-rate update strategy cannot capture temporal dynamics, its heavyweight captioning pipeline creates a processing bottleneck, and its purely similarity-based retrieval ignores the temporal sensitivity of queries. Our framework, StreamRAG, addresses these limitations through its three modules, to our knowledge, the first RAG framework explicitly designed for streaming video QA.

3. Methodology

3.1. Overview

Problem Definition. Streaming Video QA requires a model to answer a query q in real time based on an incrementally observed video stream $V_{[0,t]} = \{v_1, \ldots, v_t\}$, where v_t denotes the visual features of the t-th frame. The model $\mathcal{M}$ generates an answer $a = \mathcal{M}(V_{[0,t]}, q)$ under strict causality—no future frames are accessible. Unlike offline settings where the complete video $V_{[0,T]}$ is available, a streaming model must dynamically incorporate new observations while retaining previously acquired knowledge.

In this work, we introduce RAG into the streaming video setting, reformulating answer generation as $a = \mathcal{M}(V_{[0,t]}, q, \mathcal{C}_q)$, where $\mathcal{C}_q$ denotes the retrieved knowledge relevant to query q. As illustrated in Fig. 2, StreamRAG incrementally constructs and maintains a knowledge base through three modules. **Streaming Event Segmentation** (Sec. 3.2) processes incoming frames, detecting semantic transitions to partition the stream into coherent events. **Knowledge Extraction Accelerator** (Sec. 3.3) accelerates caption-based knowledge extraction, minimizing the latency of knowledge base updates to meet real-time demands. **Query-aware Dynamic Knowledge Integration** (Sec. 3.4) retrieves temporally and semantically relevant entries from the knowledge base and fuses them into the generation process, enabling accurate, context-aware responses for online video QA.

3.2. Streaming Event Segment

The proposed Streaming Event Segment processes video streams in an incremental frame-by-frame manner $\mathcal{W}_t = \{v_{t-w}, \ldots, v_t\}$ of fixed window size w to enable real-time boundary detection. For each incoming frame v_t at time t, the system performs the following computation when the buffer contains sufficient frames ($t \geq w$): *Semantic similarity* c_t^{ViT} using the [CLS] token of a Vision Transformer (ViT) between adjacent frames. Then it proceeds by analyzing depth scores derived from the similarity representation within the sliding window $\mathcal{W}_t$. For the i-th frame within the window, following [27], the depth score d_i is computed as:

$$d_i = \frac{(c_{l_i}^{\text{ViT}} + c_{r_i}^{\text{ViT}} - 2c_i^{\text{ViT}})}{2}, \tag{1}$$

where $c_{l_i}^{\text{ViT}}$ and $c_{r_i}^{\text{ViT}}$ represent the peak similarity scores to the left and right of c_i^{ViT}, respectively, within the current window. A statistically significant boundary is detected when d_i exceeds the threshold $\mu + \tau \cdot \sigma$, where μ and σ denote the mean and standard deviation of the depth scores in $\mathcal{W}_t$, while τ controls the segmentation granularity.

Upon detecting a boundary at time t_b, the system extracts

the frame sequence $\{v_{t_{prev}+1}, ..., v_{t_b}\}$ as a semantically coherent event segment e_k, (Event-k in Fig. 2), where t_{prev} denotes the previous boundary timestamp. The analysis window $\mathcal{W}_t$ then advances to t_b. This window shifting operation preserves temporal continuity while ensuring:

$$\mathcal{W}_t \leftarrow \{v_{t_b+1}, \ldots, v_{t_b+w}\} \tag{2}$$

The incremental processing enables streaming event segmentation. This approach identifies event boundaries in video streams without disrupting the internal causality of the video, thereby facilitating better knowledge extraction in subsequent stages and contributing to overall performance improvement.

3.3. Knowledge Extraction Accelerator

Conventional approaches [5, 10, 31] to accelerating streaming RAG knowledge base construction typically rely on keyframe selection and filtering or token compression to reduce redundancy. Instead, we leverage the inherent semantic continuity in streaming videos to minimize redundancy by reusing and refining output semantic tokens, while simultaneously improving system robustness. Given the previous event caption $C_{k-1} = (w_1, \ldots, w_{L_c}) \in \mathcal{V}^{L_c}$ and the new event e_k with its associated video frames $\mathbf{V}_k = \{v_t\}_{t=t_{prev}+1}^{t_b}$, where $\mathcal{V}$ is the vocabulary, L_c is the caption length, our method processes these inputs through the following stages:

Input Representation The text processor encodes the previous event caption $C_{k-1} \in \mathcal{V}^{L_c}$ into a sequence of token embeddings:

$$E_c = \text{TextProcessor}(C_{k-1}) \in \mathbb{R}^{L_c \times d} \tag{3}$$

where d is the embedding dimension. Simultaneously, the new event's video frames $\mathbf{V}_k$ and the textual query $Q =$ "Describe this video" are processed through a multimodal encoder:

$$E_v = \text{MultimodalEncoder}(\mathbf{V}_k, Q) \in \mathbb{R}^{L_v \times d} \tag{4}$$

yielding a joint video-text representation, where L_v is the video feature sequence length. These components are concatenated to form the combined input:

$$X = [E_v; E_c] \in \mathbb{R}^{(L_v+L_c)\times d} \tag{5}$$

Joint Processing The captioner performs a single forward pass on the concatenated input $X = [E_v; E_c] \in \mathbb{R}^{(L_v+L_c)\times d}$ (akin to prefilling), generating the entire output probability matrix **in parallel**:

$$P = \text{Model}(X) \in \mathbb{R}^{(L_v+L_c)\times|\mathcal{V}|} \tag{6}$$

where $|\mathcal{V}|$ denotes the size of vocabulary. The matrix P captures both the influence of visual context on text generation and the preservation of relevant caption content.

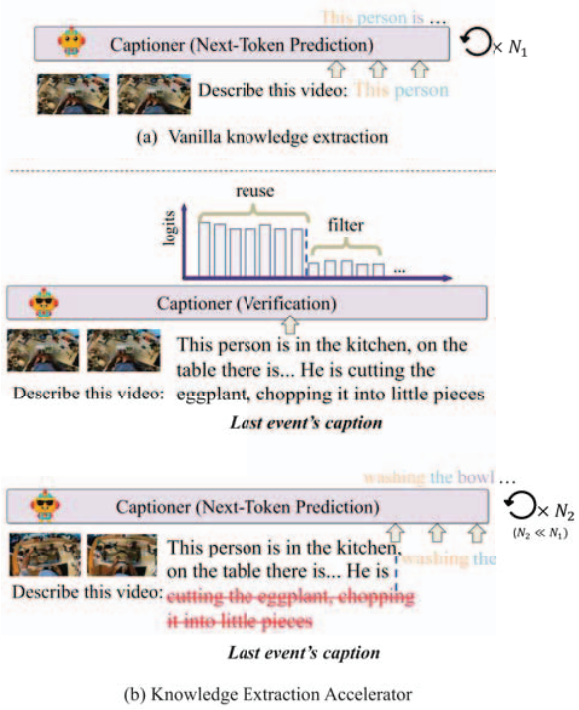

Figure 3. Comparisons of Vanilla Knowledge Extraction (a) and our proposed Knowledge Extraction Accelerator (b).

Generation Confidence Analysis Given the probability matrix $P \in \mathbb{R}^{(L_v+L_c)\times|\mathcal{V}|}$ from the joint processing step, we evaluate the model's alignment with the original caption $C_{k-1} = (w_1, \ldots, w_{L_c})$ by examining next-token prediction scores. For each position $z \in [1, L_c - 1]$ in the original caption, we extract the model's predicted probability for the ground-truth next token w_{z+1}:

$$p_z = P[L_v + z - 1, w_{z+1}] \tag{7}$$

where $L_v + z - 1$ indexes the position in P corresponding to the z-th caption token. The logarithmic transformation of these probabilities provides stable confidence measures:

$$\ell_z = \log p_z \tag{8}$$

To identify significant confidence drops in the generation process, we establish a detection mechanism based on relative and absolute thresholds. Given the sequence of logarithmic probabilities $\ell_z{}_{z=1}^{L_c-1}$, we define the moving average over a window of size $\delta \in \mathbb{Z}^+$ as:

$$\mu_{[z-\delta:z-1]} = \frac{1}{\delta} \sum_{j=z-\delta}^{z-1} \ell_j \tag{9}$$

The divergence point o is determined as the first position $z \in [\delta+1, L_c-1]$ where the confidence drop simultaneously

exceeds both a relative threshold $\alpha \in (0,1)$ and an absolute threshold $\beta > 0$, formally expressed as:

$$o = \min\{z \in [\delta+1, L_c-1] \mid (\mu_{[z-\delta:z-1]} - \ell_z) > \max(\alpha \cdot |\mu_{[z-\delta:z-1]}|,\ \beta)\} \quad (10)$$

A high value of p_z indicates that the original caption's $(z+1)$-th token remains probable in the new visual context, while a significant drop in p_z suggests the model's preferred continuation diverges from the original caption beyond position z, signaling the need for content regeneration.

Adaptive Generation Strategy Building upon the identified divergence point k from confidence analysis, this stage first verifies the preservable portion of the original caption $C_{k-1} = (w_1, \ldots, w_{L_c})$ before regenerating the subsequent content conditioned on the new visual features. The generation process employs an adaptive two-phase approach that maintains verified context before producing novel continuations, as in Fig. 3. The context preservation phase constructs the input representation by combining the visual encoding $E_v \in \mathbb{R}^{L_v \times d}$ with embeddings of the verified caption prefix $C_{k-1}^{1:o} = (w_1, \ldots, w_o)$:

$$X_{\text{preserve}} = [E_v; \text{TextProcessor}(C_{k-1}^{1:o})] \in \mathbb{R}^{(L_v+o)\times d} \quad (11)$$

Subsequent tokens are generated through conditional autoregressive sampling from the model's probability distribution P:

$$G \sim \text{Model}(X_{\text{preserve}}), \quad |G| \leq 128 \quad (12)$$

where $G = (g_{o+1}, \ldots, g_{o+m})$ represents the newly generated token sequence of length m. The final output seamlessly integrates preserved and generated content:

$$C_k = (w_1, \ldots, w_o, g_{o+1}, \ldots, g_{o+m}) \in \mathcal{V}^{o+m} \quad (13)$$

By validating and reusing prior semantic tokens rather than recomputing from raw inputs, we achieve significant efficiency gains for real-time applications while preserving output quality. This represents a fundamental shift from traditional input-reduction strategies to intelligent output-semantic preservation in streaming video understanding. In streaming applications such as autonomous driving and third-person video from AI glasses, where scenes naturally evolve in a continuous manner, our approach is particularly suited to leverage this inherent coherence. Then, it is incorporated into the knowledge base $\mathcal{KB}^+ = \{C_1, \ldots, C_{k-1}\}$, which maintains the complete sequence of previously generated event descriptions. The update operation is formally expressed as:

$$\mathcal{KB}^+ \leftarrow \mathcal{KB}^+ \cup C_k \quad (14)$$

This incremental update mechanism ensures $\mathcal{KB}^+$ always contains the most recent and consistent event representations while maintaining temporal coherence across the entire sequence $\mathcal{KB}^+ = \{C_i\}_{i=1}^k$ for subsequent processing modules.

3.4. Dynamic Knowledge Integration

Query Instantaneity Classification via LLM The system first evaluates whether a query requires real-time information. We employ an LLM-based temporal classifier that processes the raw query text to produce a normalized urgency score $S_t(q) \in [0,1]$. This score reflects the likelihood that the query concerns time-sensitive content, where values approaching 1 indicate urgent topics (e.g., "What are they doing?") while scores near 0 suggest perennial questions (e.g., "What were the climbers doing before opening the backpack?").

Historical Relevance Assessment The system retrieves the top-3 most relevant historical captions $C_{(1)}, C_{(2)}, C_{(3)}$ from knowledge base $\mathcal{KB}$ based on their semantic similarity to the user query q, computed using the pretrained text encoder $\psi(\cdot)$. The mean relevance score $S_r(q) = \frac{1}{3}\sum_{j=1}^{3} \text{sim}(q, C_{(j)})$ serves as the quantitative alignment measure between the query and retrieved video-derived knowledge, where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity in the shared embedding space.

Weighting Strategy Our framework dynamically balances temporal freshness $S_t(q)$ and semantic irrelevance $(1 - S_r(q))$ through a gating mechanism:

$$S_c(q) = \gamma \cdot S_t(q) + (1-\gamma) \cdot (1 - S_r(q)) \quad (15)$$

where $\gamma \in (0,1)$ is a predefined balancing parameter.

Retrieval Strategy Selection The composite score $S_c(q)$ jointly reflects query urgency and historical insufficiency, enabling robust retrieval decisions. Only when $S_c(q) > \theta_1$—indicating both high temporal sensitivity and low historical relevance—does the system rely solely on the most recent event caption (K_1 in Fig. 2). Otherwise, the top-k semantically closest historical captions are retrieved, where k is inversely modulated by $S_c(q)$: a moderate score ($\theta_2 < S_c \leq \theta_1$) retrieves k_1 entries (K_2 in Fig. 2), while a low score ($S_c \leq \theta_2$) retrieves $k_2 > k_1$ entries to incorporate broader historical context (K_3 in Fig. 2). This design ensures that relevant historical knowledge is preserved even for time-sensitive queries when the knowledge base contains useful context. The selected captions are assembled into $\mathcal{C}_q$ as the augmented context for answer generation.

Table 1. Performance on OVOBench. Attribute definitions are in the supplementary.

Model	Setting	Real-Time Visual Perception						Avg.	Backward Tracing			Avg.	Forward Active Responding			Avg.
		OCR	ACR	ATR	STU	FPD	OJR		EPM	ASI	HLD		REC	SSR	CRR	
Human																
Human Agents		93.96	92.57	94.83	92.70	91.09	94.02	93.20	92.59	93.02	91.37	92.33	95.48	89.67	93.56	92.90
Blind LLMs																
GPT-4-turbo	-	28.86	24.77	25.67	33.76	27.72	26.63	27.90	42.76	48.65	70.05	53.82	-	-	52.92	-
Proprietary Multimodal Models-Offline																
Gemini 1.5 Pro	1fps	85.91	66.97	79.31	58.43	63.37	61.96	69.32	58.59	76.35	52.64	62.54	35.53	74.24	61.67	57.15
GPT-4o	64	69.8	64.22	71.55	51.12	70.3	59.78	64.46	57.91	75.68	48.66	60.75	27.58	73.21	59.4	53.40
Open-source Multimodal Models-Offline																
LLaVA-Video-7B	64	69.13	58.72	68.83	49.44	74.26	59.78	63.52	56.23	57.43	7.53	40.40	34.10	69.95	60.42	54.82
LongVU-7B	1fps	55.70	49.54	59.48	48.31	68.32	63.04	57.40	43.10	66.22	9.14	39.49	12.18	69.48	60.83	47.50
InternVL-V2.5-8B	64	68.46	58.72	68.97	44.94	67.33	55.98	60.73	43.10	61.49	27.41	44.00	26.50	59.14	54.14	46.60
Qwen2-VL-72B	64	65.77	60.55	69.83	51.69	69.31	54.35	61.92	52.53	60.81	57.53	56.95	38.83	64.07	45.00	49.30
Qwen2-VL-7B	1fps	69.13	53.21	63.79	50.56	66.34	60.87	60.65	44.44	66.89	34.41	48.58	31.09	65.98	24.58	40.55
Qwen2-VL-7B+ours	1fps	84.56	69.72	68.97	56.74	71.29	68.48	69.96	49.49	46.24	63.51	53.08	31.25	67.73	27.94	42.30
Open-source Multimodal Models-Online																
Flash-VStream-7B	1fps	24.16	29.36	28.45	33.71	25.74	28.80	28.37	39.06	37.16	5.91	27.38	8.02	67.25	60.00	45.09
VideoLLM-online-8B	2fps	8.05	23.85	12.07	14.04	45.54	21.20	20.79	22.22	18.80	12.18	17.73	-	-	-	-
Dispider	1fps	57.72	49.54	62.07	44.94	61.39	51.63	54.55	48.48	55.41	4.30	36.06	18.05	37.36	48.75	34.72
ViSpeak	1fps	75.17	58.72	71.55	51.12	74.26	66.85	66.28	59.93	48.65	63.98	57.52	33.81	68.52	60.42	54.25
ViSpeak + StreamChatRAG	1fps	67.11	55.96	68.97	48.31	80.20	66.30	64.48	56.23	57.43	64.52	59.39	21.49	67.09	60.00	49.53
ViSpeak + VideoRAG	1fps	72.48	44.04	62.07	43.26	64.26	57.07	57.21	58.25	35.81	60.22	51.43	31.09	68.52	60.42	53.82
ViSpeak + Ours	1fps	81.21	63.30	69.83	52.25	79.21	72.28	69.68	65.66	63.98	52.70	60.78	34.38	68.36	61.67	54.80

Table 2. Component ablation study. SES: Streaming Event Segmentation; KEA: Knowledge Extraction Accelerator; DKI: Dynamic Knowledge Integration.

KEA	SES	DKI	R-Avg	B-Avg
			55.98	46.46
✓			65.33	49.75
✓	✓		68.54	51.28
✓	✓	✓	69.96	53.08

Table 3. Performance comparison across different token reuse ratios, including its accuracy on historical questions (B-Avg), real-time questions (R-Avg), and the corresponding latency.

Token	R-Avg	B-Avg	Latency	Accelerate
0%	68.85	51.80	15.24s	–
∼18%	69.96	53.08	11.12s	27%
∼33%	69.23	50.47	8.43s	45%

4. Experiments

In this section, we evaluate the performance of Stream-RAG, aiming to answer the following questions: (1) Can Stream-RAG effectively improve the factuality of mllms in streaming video question answering ? (2) How effective is each proposed component on performance? (3) Does reusing knowledge accelerate inference while maintaining performance? (4) Does dynamic knowledge integration based on query instantaneity dynamics better select RAG knowledge sources to optimize performance?

Table 4. Performance comparison of knowledge base update strategies: event-triggered segmentation (SES) *vs*. fixed-interval methods.

Extraction Method	Frames	R-Avg	B-Avg
Fixed-Interval	16	65.33	49.75
Fixed-Interval	32	68.98	50.04
SES	∼ 32	69.96	53.08

Table 5. Performance comparison of our adaptive knowledge integration (+Ours) against two fixed retrieval baselines (+LatestKB and +FullKB), demonstrating the effectiveness of dynamic retrieval scope selection.

Method	R-Avg	B-Avg
+LatestKB	69.59	51.25
+FullKB	68.54	51.28
+Ours	69.96	53.08

4.1. Experimental Setups

Evaluation Datasets. Following [7, 22], we evaluate on two streaming video QA benchmarks: OVO-Bench [20] (644 videos, 0.5–30 min, 7 domains) and Streaming-

Table 6. Performance Comparison On StreamingBench. Attribute definitions are in the supplementary.

Model	Frames	Real-Time Visual Understanding										Avg.	Contextual Understanding		Avg.
		OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT		ACU	MCU	
Human															
Human Agents	-	89.47	92.00	93.60	91.47	95.65	92.52	88.00	88.75	89.74	91.30	91.46	88.80	90.40	92.1
Proprietary Multimodal Models-Offline															
Gemini 1.5 Pro	32	79.02	80.47	83.54	79.67	80.00	84.74	77.78	64.23	71.95	48.70	75.69	51.41	40.73	74.5
GPT-4o	32	77.11	80.47	83.91	76.47	70.19	83.80	66.67	62.19	69.12	49.22	73.28	41.20	38.40	57.6
Open-source Multimodal Models-Offline															
Qwen2-VL-72B	32	75.20	82.81	73.19	77.45	68.32	71.03	72.22	61.19	61.47	46.11	69.04	31.20	26.00	28.6
LLaVA-NeXT-Video-7B	32	78.20	70.31	73.82	76.80	63.35	69.78	57.41	56.10	64.31	38.86	66.96	29.20	30.40	29.8
LLaVA-OneVision-7B	128	80.38	74.22	76.03	80.72	72.67	71.65	67.59	65.45	65.72	45.08	71.12	32.70	34.20	33.50
LongVU-7B	32	70.03	63.28	61.20	70.92	62.73	59.50	61.11	53.66	54.67	34.72	59.96	32.80	29.60	31.20
VILA-1.5	8B	53.68	49.22	70.98	56.86	53.42	53.89	54.63	48.78	50.14	17.62	52.32	26.80	34.00	43.20
Video-CCAM	14B	56.40	57.81	65.30	62.75	64.60	51.40	42.59	47.97	49.58	31.61	53.96	27.60	24.40	42.53
Video-LLaMA2	7B	55.86	55.47	57.41	58.17	52.80	43.61	39.81	42.68	45.61	35.23	49.52	24.80	26.80	40.40
Qwen2-VL-7B	1fps	75.75	79.69	76.58	79.08	74.53	75.08	74.07	65.85	65.16	41.97	71.15	34.27	26.40	30.34
Qwen2-VL-7B+ours	1fps	80.38	77.34	89.24	84.92	79.38	82.55	75.93	65.85	76.49	46.63	77.33	47.37	41.20	44.27
Open-source Multimodal Models-Online															
Flash-VStream-7B	1fps	25.89	43.57	24.91	23.87	27.33	13.08	18.52	25.20	23.87	48.70	23.23	28.40	26.00	35.5
VideoLLM-online-8B	2fps	39.07	40.06	34.49	31.05	45.96	32.40	31.48	34.16	42.49	27.89	35.99	28.45	24.19	32.48
ViSpeak	1fps	79.84	88.28	83.28	81.05	76.40	75.08	70.37	65.85	77.34	34.20	74.36	38.80	36.80	37.80
ViSpeak+ours	1fps	83.37	90.63	86.43	83.99	78.26	85.67	77.78	73.17	77.90	31.10	78.12	42.40	39.60	41.00

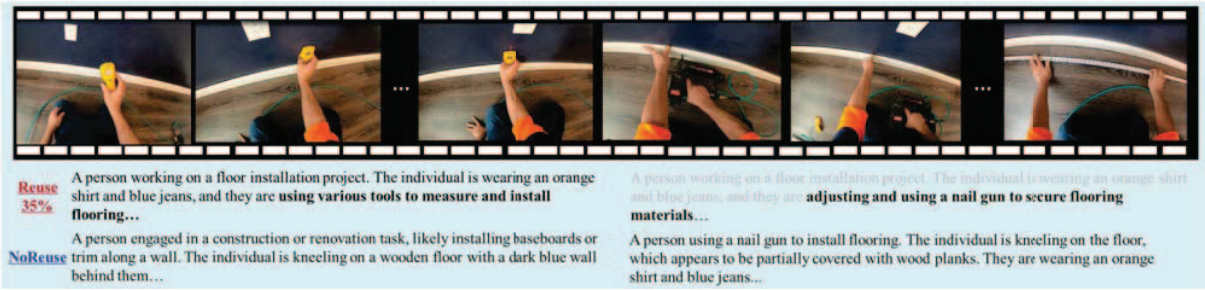

Figure 4. Comparison of token reuse in a knowledge extraction accelerator: The top example (35% reuse) maintains scene consistency by preserving unchanged elements while regenerating only dynamic content tokens, whereas the bottom example (No reuse) exhibits more fragmented and diverse output due to reduced token recycling.

Bench [15] (900 videos, 4,500 temporally distributed questions, 8 categories).

Baseline Methods. We evaluate StreamRAG's ability to enhance open-source mllms through comprehensive experiments comparing two key baselines: (1) Qwen2-VL[4] series as the representative open-source model, and (2) ViSpeak[7], the current state-of-the-art in streaming video understanding. To specifically demonstrate our method's advantages in streaming video QA, we include comparative analysis with two RAG variants: VideoRAG[17] for non-streaming scenarios and StreamChat[31] for streaming contexts.

Experimental Settings. For **SES**, the window size is $w = 8$ and the segmentation threshold $\tau = 0.7$. For **DKI**, the balancing factor $\gamma = 0.5$, with retrieval depths $k = 1$ (recent-only), $k = 3$, and $k = 5$ for the three strategy tiers. For **KEA**, the lookback window $\delta = 20$, relative threshold $\alpha = 0.85$, and absolute threshold $\beta = 1$.

4.2. Main Results

Improving open-source mllms with StreamRAG for Streaming VideoQA. We conduct rigorous experimental evaluations of our Stream-RAG framework on two established benchmarks for streaming video understanding: OVObench and StreamBench (Tables 1 and 6). Our methodology compares model performance before and after adopting the proposed framework, demonstrating statistically significant improvements across most of evaluation metrics. The results clearly demonstrate significant improvements in streaming video question answering after incorporating our framework. Notably, the framework substantially enhances general-purpose models, as evidenced

by Qwen2VL's 20% accuracy improvement after adopting StreamRAG, effectively bridging the performance gap with ViSpeak[7]. Furthermore, even specialized models benefit from our framework, with ViSpeak achieving an additional 5% performance gain after integration. These consistent improvements across diverse model architectures validate StreamRAG's versatility as a plug-and-play solution.

Comparison with state-of-the-art VideoRAG on OVOBench. In Table 1, we adopt the best-performing ViSpeak model as the baseline and conduct comparative experiments with two state-of-the-art RAG frameworks: VideoRAG and StreamChatRAG. Experimental results show that VideoRAG degrades ViSpeak's performance. Analysis reveals that audio data in streaming video is ineffective, and indiscriminate knowledge integration introduces noise, impairing model performance. On the other hand, while StreamChatRAG supports dynamic updates of the RAG knowledge base, its uniform sampling strategy for updates inherently disrupts the temporal causality of video stream events.

4.3. Analysis

This section presents a comprehensive module-wise performance analysis and systematic experiments to quantify StreamRAG's performance improvements and computational efficiency gains.

Ablation Studies. We conduct a comprehensive series of ablation studies to systematically evaluate the contribution of each key component in our proposed Stream-RAG framework. As demonstrated in Table 2, our analysis reveals three crucial findings: (1) The construction of the retrieval-augmented knowledge base demonstrates measurable benefits for video understanding. This observation aligns with current research findings in non-streaming video RAG systems. (2) Our proposed streaming event segmentation mechanism significantly enhances performance, as it effectively preserves complete event causality chains. (3) The adaptive knowledge injection module specifically addresses the temporal sensitivity requirements in streaming video scenarios. When answering temporal queries like "what is he doing?", as the module successfully filters out irrelevant information while retaining temporally pertinent knowledge. This advantage becomes especially pronounced in real-time processing scenarios where computational efficiency and temporal precision are paramount.

Trade-off Between Knowledge-Reuse Efficiency and Effectiveness As illustrated in Table 3, we systematically evaluate the performance trade-offs under varying knowledge reuse ratios. The baseline method (first row), which employs conventional caption extraction akin to standard RAG systems, establishes our reference performancewith a latency of 15.24s. Our optimized knowledge extraction accelerator (second row) achieves a favorable balance: by reusing about 18% of cached knowledge, it attains superior accuracy while reducing latency by 27% . However, aggressive reuse at about 33% (third row) reveals diminishing returns—though latency improves further to 8.43s (45% acceleration), the accuracy declines to 69.23% R-Avg and 50.47% B-Avg, suggesting that excessive reuse introduces noise or redundancy. These findings underscore a critical trade-off: moderate knowledge reuse enhances both efficiency and effectiveness, whereas higher reuse ratios prioritize speed at the cost of model robustness.

Update Strategy. Table 4 compares knowledge base update strategies. Fixed-interval methods (first two rows) update every 32 frames regardless of content, whereas our SES updates at detected semantic boundaries, better preserving event integrity. This yields a 2% accuracy gain, particularly on historical event analysis tasks.

Knowledge Integration. Table 5 compares our adaptive retrieval strategy (+Ours) with two fixed baselines: using only the latest caption (+LatestKB) and using all available captions (+FullKB). +LatestKB discards useful historical context, while +FullKB introduces noise and increases latency. Our method dynamically adjusts the retrieval scope per query, achieving the best balance across both benchmarks.

5. Conclusions

We introduce the first streaming RAG framework for real-time video processing, featuring event-aware segmentation, efficient knowledge extraction, and dynamic knowledge injection. This plug-and-play solution enhances MLLMs without any architectural changes, setting a new standard for dynamic knowledge retrieval in live video understanding. By continuously processing video streams, the framework maintains an always-up-to-date contextual model, enabling superior performance for real-time querying and analytics. This work establishes a new paradigm for deploying RAG in dynamic, time-sensitive visual environments, moving beyond its traditional use in static corpora.

Acknowledgements

This work is supported in part by the National Key R&D Program of China (NO. 2024YFB3908503 and 2024YFB3908500), and in part by the National Natural Science Foundation of China (NO. 62322608).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Muhammad Arslan, Hussam Ghanem, Saba Munawar, and Christophe Cruz. A survey on rag with llms. *Procedia computer science*, 246:3781–3790, 2024. 1

[3] Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Mingchen Zhuge, Jian Ding, Deyao Zhu, Jürgen Schmidhuber, and Mohamed Elhoseiny. Goldfish: Vision-language understanding of arbitrarily long videos. In *European Conference on Computer Vision*, pages 251–267. Springer, 2024. 2

[4] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 1, 7

[5] 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. 2, 4

[6] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6491–6501, 2024. 1

[7] Shenghao Fu, Qize Yang, Yuan-Ming Li, Yi-Xing Peng, Kun-Yu Lin, Xihan Wei, Jian-Fang Hu, Xiaohua Xie, and Wei-Shi Zheng. Vispeak: Visual instruction feedback in streaming videos. *arXiv preprint arXiv:2503.12769*, 2025. 2, 6, 7, 8

[8] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2024. 2

[9] Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. Online video understanding: A comprehensive benchmark and memory-augmented method. *arXiv e-prints*, pages arXiv–2501, 2024. 2

[10] Soyeong Jeong, Kangsan Kim, Jinheon Baek, and Sung Ju Hwang. Videorag: Retrieval-augmented generation over video corpus. *arXiv preprint arXiv:2501.05874*, 2025. 2, 4

[11] Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O Arik. Long-context llms meet rag: Overcoming challenges for long inputs in rag. *arXiv preprint arXiv:2410.05983*, 2024. 1

[12] Xie Junlin, Zhihong Chen, Ruifei Zhang, and Guanbin Li. Large multimodal agents: a survey. *Visual Intelligence*, 3(1): 24, 2025. 1

[13] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024.

[14] 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. *arXiv preprint arXiv:2311.10122*, 2023. 1

[15] 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. 7

[16] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. 1

[17] Yongdong Luo, Xiawu Zheng, Xiao Yang, Guilin Li, Haojia Lin, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, and Rongrong Ji. Video-rag: Visually-aligned retrieval-augmented long video comprehension. *arXiv preprint arXiv:2411.13093*, 2024. 2, 7

[18] Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. Drvideo: Document retrieval based long video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18936–18946, 2025. 2

[19] Zhenyu Ning, Jieru Zhao, Qihao Jin, Wenchao Ding, and Minyi Guo. Inf-mllm: Efficient streaming inference of multimodal large language models on a single gpu. *arXiv preprint arXiv:2409.09086*, 2024. 2

[20] 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. 2, 6

[21] Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37:119336–119360, 2024. 2

[22] 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. 2, 6

[23] Xubin Ren, Lingrui Xu, Long Xia, Shuaiqiang Wang, Dawei Yin, and Chao Huang. Videorag: Retrieval-augmented generation with extreme long-context videos. *arXiv preprint arXiv:2502.01549*, 2025. 2

[24] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1

[25] 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. *arXiv preprint arXiv:2505.05467*, 2025. 2

[26] Kuo Wang, Quanlong Zheng, Junlin Xie, Yanhao Zhang, Jinguo Luo, Haonan Lu, Liang Lin, Fan Zhou, and Guanbin Li. Free-moref: Instantly multiplexing context perception capabilities of video-mllms within single inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22499–22508, 2025. 1

[27] Yuxuan Wang, Yiqi Song, Cihang Xie, Yang Liu, and Zilong Zheng. Videollamb: Long streaming video understanding with recurrent memory bridges. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24170–24181, 2025. 3

[28] Yushen Wei, Yang Liu, Hong Yan, Guanbin Li, and Liang Lin. Visual causal scene refinement for video question answering. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 377–386, 2023. 1

[29] Cheng-Kuang Wu, Zhi Rui Tam, Chieh-Yen Lin, Yun-Nung Vivian Chen, and Hung-yi Lee. Streambench: Towards benchmarking continuous improvement of language agents. *Advances in Neural Information Processing Systems*, 37:107039–107063, 2024. 2

[30] Junlin Xie, Ruifei Zhang, Zhihong Chen, Xiang Wan, and Guanbin Li. Whodunitbench: Evaluating large multimodal agents via murder mystery games. *Advances in Neural Information Processing Systems*, 37:86655–86687, 2024. 1

[31] 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. *arXiv preprint arXiv:2501.13468*, 2025. 2, 4, 7

[32] Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*, 2024. 1

[33] Zhucun Xue, Jiangning Zhang, Xurong Xie, Yuxuan Cai, Yong Liu, Xiangtai Li, and Dacheng Tao. Adavideorag: Omni-contextual adaptive retrieval-augmented efficient long video understanding. *CoRR*, 2025. 2

[34] Zhenyu Yang, Yuhang Hu, Zemin Du, Dizhan Xue, Shengsheng Qian, Jiahong Wu, Fan Yang, Weiming Dong, and Changsheng Xu. Svbench: A benchmark with temporal multi-turn dialogues for streaming video understanding. *arXiv preprint arXiv:2502.10810*, 2025. 2

[35] 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. *arXiv preprint arXiv:2504.17343*, 2025. 2

[36] Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-vstream: Memory-based real-time understanding for long video streams. *arXiv preprint arXiv:2406.08085*, 2024. 2

[37] Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Junnan Dong, Hao Chen, Yi Chang, and Xiao Huang. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*, 2025. 2

[38] Ruifei Zhang, Junlin Xie, Wei Zhang, Weikai Chen, Xiao Tan, Xiang Wan, and Guanbin Li. Adadrive: Self-adaptive slow-fast system for language-grounded autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5112–5121, 2025. 1

[39] Ruifei Zhang, Wei Zhang, Xiao Tan, Sibei Yang, Xiang Wan, Xiaonan Luo, and Guanbin Li. Vldrive: Vision-augmented lightweight mllms for efficient language-grounded autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5923–5933, 2025.

[40] Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. Llava-mini: Efficient image and video large multimodal models with one vision token. *arXiv preprint arXiv:2501.03895*, 2025. 1

[41] Yucheng Zhao, Chong Luo, Chuanxin Tang, Dongdong Chen, Noel Codella, and Zheng-Jun Zha. Streaming video model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14602–14612, 2023. 2