STREAMVLM: ADAPTING OFFLINE VISION-LANGUAGE MODELS FOR STREAMING UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent vision-language models (VLMs) have demonstrated strong capabilities in video understanding. However, most existing VLMs are designed for offline inference, where the entire video sequence is jointly processed during reasoning. Directly adapting such offline VLMs to streaming tasks is challenging, since streaming scenarios require the model to incrementally process incoming frames without access to future visual content. In this work, we propose StreamVLM, a training-free framework for adapting offline VLMs to streaming video understanding. StreamVLM introduces Event-based Streaming Encoding, which dynamically constructs KV-cache memories according to motion-aware temporal boundaries to preserve coherent local context during streaming inference. In addition, we propose a KNN-based KV-Cache Retrieval mechanism that models neighborhood relationships among historical memory entries to enable context-aware retrieval over correlated KV representations. Extensive experiments on both offline and streaming VideoQA benchmarks demonstrate that StreamVLM consistently improves video reasoning performance over existing streaming VLM baselines.

1 INTRODUCTION

Recent advancements in vision-language models (VLMs) Zhang et al. (2024b); Bai et al. (2023); Chen et al. (2024b); Marafioti et al. (2025) have substantially improved video understanding across a broad range of tasks, including video captioning Zhao et al. (2024), retrieval Xu et al. (2023), and summarization Hua et al. (2025). Despite these advances, most existing VLMs are developed under an offline setting, where the entire video sequence is available during both training and inference. However, many real-world applications, such as robotics Shao et al. (2025), surveillance Borodin et al. (2025), and live broadcasting Wang et al. (2026), operate in a streaming environment, where future frames are inaccessible at inference time. Given the widespread adoption and strong performance of existing offline VLMs, a natural question arises: can these offline VLMs be effectively adapted to streaming video understanding?

Adapting offline VLMs to streaming settings is highly desirable, but introduces several fundamental challenges. In streaming scenarios, the model can only observe video frames sequentially and must reason without access to future context. Consequently, the model must: (1) maintain an effective memory of past observations over long temporal horizons, and (2) retrieve the most relevant historical information for accurate reasoning and question answering when a query arrives. Recently, ReKV Di et al. (2025) proposed a unified framework for adapting offline VLMs to streaming video understanding, as illustrated in Figure 2. Specifically, past video streams are incrementally processed by a VLM and stored in a KV cache bank as historical memory. When a question arrives, the framework computes the similarity between the query and cached KV representations, followed by a Top-K retrieval step to select the most relevant historical information. The retrieved KV cache is then combined with the current video frames to support downstream reasoning and answer generation. Despite its effectiveness, ReKV still suffers from several limitations. First, the framework partitions streaming videos into fixed uniform chunks during memory construction. Such chunk-based segmentation may split semantically coherent events across different memory units, leading to fragmented KV representations and weakened temporal consistency. As a result, strongly correlated frames may be encoded independently, causing the stored memory to deviate from the original

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

full-context video representation. Second, the retrieval process mainly relies on direct query–KV similarity matching, where each KV entry is treated independently during retrieval. However, historical KV caches in streaming videos are often highly correlated, as temporally adjacent or semantically related events naturally form structured dependencies in the memory space. Ignoring these relationships may lead to fragmented or incomplete retrieval results, where individually relevant KV entries are selected while their associated contextual information is omitted.

To address these challenges, we propose StreamVLM, a training-free framework for adapting offline VLMs to streaming video understanding. Unlike prior approaches that partition streaming videos into fixed uniform chunks, StreamVLM first employs event segmentation to identify temporal boundaries associated with motion transitions. Based on these event boundaries, the video stream is incrementally encoded into event-level memory units, allowing each KV cache representation to preserve more coherent temporal and semantic information. To further improve retrieval quality, we introduce a KNN-based KV-Cache Retrieval mechanism that explicitly models relationships among historical memory entries. Instead of performing retrieval solely based on independent query–KV similarity matching, our method constructs a KNN graph over the KV-cache bank and performs dependency-aware retrieval by jointly considering query relevance and inter-memory connectivity. This enables the retriever to recover more coherent contextual memory rather than isolated KV fragments for downstream reasoning. Our main contributions are summarized as follows:

- We propose StreamVLM, a training-free framework for adapting offline VLMs to streaming video understanding through structured memory construction and retrieval.
- We introduce Event-based Streaming Encoding and KNN-based KV-Cache Retrieval to improve temporal coherence during streaming memory construction and contextual coherence during historical memory retrieval.
- Extensive experiments on both offline and streaming VideoQA benchmarks demonstrate that StreamVLM consistently improves long-video reasoning performance over existing streaming VLM baselines.

2 Related Work

2.1 Vision-Language Models for Video Understanding

Vision-language models (VLMs) typically consist of a vision encoder Maniparambil et al. (2024), a text encoder Renear (2004), and a large language model (LLM) Zhao et al. (2023); Nam et al. (2024). The vision encoder extracts visual features, while the text encoder processes textual inputs. These features are then projected into a shared embedding space, often via a multi-layer perceptron (MLP) Popescu et al. (2009), and subsequently fed into the LLM for multimodal reasoning and generation. Recently, with the rapid advancement of LLMs, VLMs have demonstrated strong performance across a wide range of video understanding tasks, including video question answering Zhong et al. (2022), video captioning Abdar et al. (2024), and summarization Meena et al. (2023). In typical video understanding settings, VLMs take as input a sequence of sampled video frames along with a textual prompt, and generate task-specific outputs. For example, Video-LLaVA Lin et al. (2024) uses video encoders and projection layers to map video features into the language feature space before feeding them, together with tokenized textual queries, into the LLM for response generation. Similarly, LLaVA-OV Li et al. (2024a) treats video frames as multiple image inputs, while Qwen2.5-VL Bai et al. (2023) adopts dynamic FPS sampling to handle videos at different sampling rates. Despite these advances, this formulation assumes that the entire video is accessible during inference, enabling global reasoning over the full temporal context. However, such an assumption inherently limits their applicability to relatively short video clips and falls short in real-world, real-time interactive scenarios, such as autonomous driving, embodied AI, and augmented reality, where video streams continuously and user queries arrive on the fly.

2.2 Streaming Video Understanding

Streaming Video Understanding refers to the setting where video frames arrive sequentially over time, and the model must perform understanding without access to future frames. VideoLLM-Online Chen et al. (2024a) proposes the LIVE framework for streaming dialogue. However, it pri-

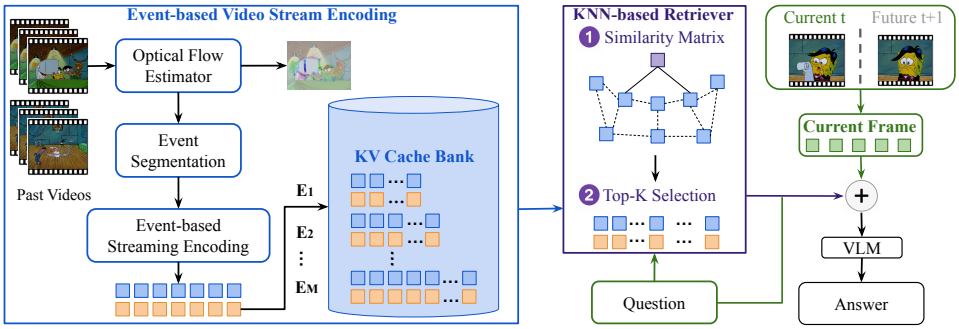


Figure 1: Overview of the proposed StreamVLM framework for adapting offline VLMs to streaming video understanding. StreamVLM consists of two key components: Event-based Video Stream Encoding and KNN-based KV-Cache Retrieval. First, optical flow estimation is used to detect motion-aware event boundaries, and the streaming video is incrementally partitioned into temporally coherent event segments. Each event segment is encoded into KV-cache representations and stored in a KV cache bank as historical memory. During inference, the KNN-based retriever constructs similarity relationships among KV-cache entries and performs dependency-aware Top-K retrieval conditioned on the query. The retrieved historical memory is then combined with the current video frames for downstream reasoning and answer generation.

marily focuses on maintaining conversational consistency and does not fully address the challenges of fine-grained temporal reasoning and efficient long-term memory management in streaming video understanding. Qian et al. Qian et al. (2024a) propose VideoStreaming with two core designs: Memory-Propagated Streaming Encoding and Adaptive Memory Selection. The former incrementally encodes incoming video clips while propagating compact representations of past observations to maintain temporal context, whereas the latter dynamically selects task-relevant memory from historical representations conditioned on the input query. However, it primarily relies on compressing historical observations into condensed memory representations and retrieving query-relevant segments, which may lead to information loss and limit its ability to capture fine-grained temporal dependencies. Subsequent works, such as Flash-VStream Zhang et al. (2024a) and Dispider Qian et al. (2025), further explore streaming video understanding by introducing memory-centric and asynchronous architectures for real-time processing. Specifically, Flash-VStream designs a two-process framework with a hierarchical memory mechanism to efficiently handle extremely long video streams and reduce temporal redundancy, while Dispider adopts a proactive streaming pipeline with disentangled processing and interaction modules to support timely and context-aware responses. However, despite improving efficiency, these methods largely depend on complex memory aggregation and asynchronous processing, which may overlook fine-grained temporal dynamics and do not fully enable continuous, incremental reasoning over streaming video. To address these limitations, ReKV Di et al. (2025) introduces a retrieval-based framework that retrieves query-relevant KV caches to support streaming video question answering. By incrementally encoding incoming frames and maintaining historical context in a KV cache, it improves adaptability to streaming scenarios. However, this approach segments the video stream into uniform chunks, which may disrupt the underlying temporal and semantic continuity. Moreover, the retrieval mechanism relies primarily on query–visual KV similarity, ignoring potential cross-modal misalignment and the structured dependencies among memory entries.

3 METHODS

3.1 PRELIMINARIES

We study the problem of adapting an offline VLM to streaming video understanding without additional training. In the offline setting, the model jointly processes the entire video sequence together with the textual query during inference. In contrast, streaming scenarios require the model to process incoming video frames incrementally without access to future visual content. To adapt offline VLMs to streaming settings, existing approaches reformulate the inference process into a three-stage

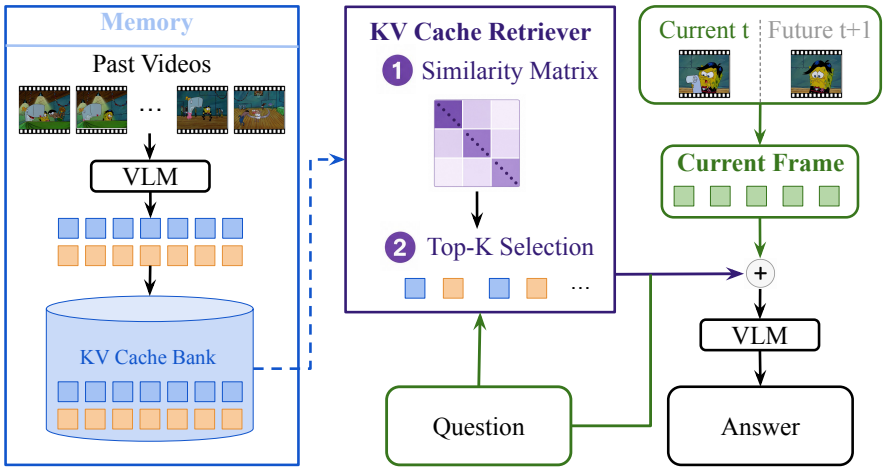


Figure 2: Overview of the streaming video understanding framework adapted from offline VLMs. Past video streams are incrementally processed by a VLM and stored as historical KV representations in a KV cache bank. When a question arrives, a KV cache retriever computes the similarity matrix between the query and cached memory representations, followed by Top-K selection to retrieve relevant historical information. The retrieved KV cache is then combined with the current video frames for downstream reasoning and answer generation.

pipeline consisting of streaming video encoding and KV-cache memory construction, KV-cache retrieval, and downstream answer generation, as illustrated in Figure 2.

Specifically, the incoming video stream is sequentially partitioned into causal segments $\{S_i\}_{i=1}^{N}$, where each segment is independently encoded into KV-cache representations:

$$(K_i, V_i) = \mathcal{E}(S_i), \tag{1}$$

where $\mathcal{E}(\cdot)$ denotes the frozen VLM encoder. The encoded KV representations are incrementally stored in a historical memory bank:

$$\mathcal{M} = \{(K_i, V_i)\}_{i=1}^{N}. \tag{2}$$

When a question q arrives at streaming time step t, the retriever selects relevant historical memory entries according to query–KV similarity:

$$\mathcal{I} = \mathrm{TopK}\big(\mathrm{sim}(q, K_i)\big), \tag{3}$$

where $\mathcal{I}$ denotes the retrieved memory indices. The retrieved historical KV cache is then combined with the current observed video frames for downstream reasoning and answer generation:

$$y = \mathcal{E}\big(\{(K_i, V_i)\}_{i\in\mathcal{I}}, f_t, q\big), \tag{4}$$

where f_t denotes the current video frame.

Limitations of Streaming Reformulation. Although such a reformulation enables offline VLMs to operate under streaming video settings, it inevitably introduces discrepancies compared with the original offline inference. First, streaming methods independently encode local video segments instead of jointly attending to the entire video sequence. Consequently, attention can only be performed within each local segment, limiting the model's ability to capture long-range temporal dependencies across the full video. Moreover, existing approaches typically partition videos into fixed uniform segments, which may separate temporally correlated frames into different memory units and result in fragmented KV representations. Second, existing retrieval methods typically perform memory retrieval solely based on independent query-KV similarity matching. However, relevant information in streaming videos is often distributed across multiple correlated memory entries rather than a single isolated KV representation. As a result, direct Top-K retrieval may only recover partial evidence of a relevant event while missing related contextual memory entries that are important for

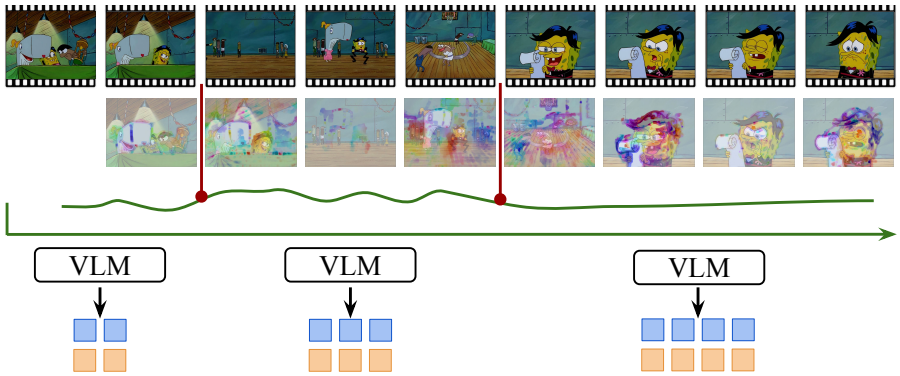


Figure 3: Illustration of the proposed Event-based Video Stream Encoding. Optical flow is first estimated between adjacent video frames to capture temporal motion variations. Event boundaries are dynamically detected according to motion transitions, partitioning the streaming video into temporally coherent event segments instead of fixed uniform windows. Each event segment is then incrementally processed by the VLM and encoded into KV-cache representations.

complete reasoning. To address these limitations, we propose a training-free framework, termed StreamVLM, as illustrated in Figure 1. StreamVLM consists of two key components: Event-based Streaming Encoding (Section 3.2) and KNN-based KV-Cache Retrieval (Section 3.3), which improve temporal coherence during streaming memory construction and contextual coherence during historical memory retrieval, respectively.

3.2 Event-based Streaming Encoding

To reduce the fragmentation introduced by fixed-window streaming encoding, we propose Event-based Streaming Encoding. Instead of constructing KV-cache entries from uniformly divided temporal windows, our method adapts the memory boundaries according to motion changes in the video stream. The goal is to keep temporally correlated frames within the same memory unit, so that each cached KV representation contains a more coherent local context.

As illustrated in Figure 3, frames within the same local event tend to exhibit relatively consistent motion patterns, whereas transitions between different events are often associated with noticeable motion changes. This observation motivates using motion transitions to adaptively determine streaming memory boundaries, thereby reducing artificial fragmentation of temporally correlated visual content across different KV-cache entries. To capture such temporal transitions, we estimate the motion change between consecutive frames using optical flow:

$$m_t = \|\text{flow}(f_{t-1}, f_t)\|, \tag{5}$$

where m_t denotes the motion magnitude at time t. We identify a boundary when the motion change exceeds a threshold τ:

$$b_t = \begin{cases} 1, & m_t > \tau, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The detected boundaries partition the streaming video into a sequence of motion-consistent segments:

$$E = \{E_1, E_2, \ldots, E_M\}. \tag{7}$$

Each segment is then encoded into KV-cache representations:

$$(K_i, V_i) = \mathcal{E}(E_i), \tag{8}$$

and stored in the historical memory bank for later retrieval.

3.3 KNN-based KV-Cache Retrieval

To reduce the retrieval discrepancy introduced by independent query–KV matching, we propose KNN-based KV-Cache Retrieval. Existing methods score each KV entry independently according

to its similarity to the query, which may retrieve only isolated memory fragments. However, relevant evidence in streaming videos often appears as a group of correlated memory entries. Therefore, we first model the neighborhood structure within the KV-cache bank and then perform retrieval by considering both query relevance and inter-memory dependency.

KNN Graph Construction. Given the KV-cache memory bank

$$\mathcal{M} = \{(K_i, V_i)\}_{i=1}^{M}, \tag{9}$$

we construct a KNN graph over the key representations. For each memory entry K_i, its neighborhood is defined as

$$\mathcal{N}(i) = \mathrm{TopK}_{j \neq i}\big(\mathrm{sim}(K_i, K_j)\big), \tag{10}$$

where $\mathcal{N}(i)$ contains the nearest KV-cache entries to K_i in the memory representation space. We further define the edge affinity between neighboring entries as

$$A_{ij} = \frac{\exp(\mathrm{sim}(K_i, K_j))}{\sum_{l \in \mathcal{N}(i)} \exp(\mathrm{sim}(K_i, K_l))}, \quad j \in \mathcal{N}(i), \tag{11}$$

which measures the dependency strength between K_i and its neighbors.

Dependency-aware Retrieval. Given a question q, we first compute the query relevance of each memory entry:

$$r_i = \mathrm{sim}(q, K_i). \tag{12}$$

Instead of directly selecting entries according to r_i, we refine the retrieval score by propagating query relevance over the KNN graph:

$$s_i = r_i + \lambda \sum_{j \in \mathcal{N}(i)} A_{ij} r_j, \tag{13}$$

where λ controls the contribution of neighborhood information. The final retrieval set is obtained by selecting the top-ranked entries according to the dependency-aware scores:

$$\mathcal{I} = \mathrm{TopK}(s_i). \tag{14}$$

Finally, the retrieved historical KV cache is combined with the current video frames and the question for answer generation.

4 Experiments

4.1 Baseline and Benchmarks

We adopt ReKV Di et al. (2025) as the primary baseline and follow its evaluation protocol to assess our method on a diverse set of long-video and streaming VideoQA benchmarks under both multiple-choice (MC) and open-ended (OE) settings. We implement our method on top of LLaVA-OV-0.5B and LLaVA-OV-7B Li et al. (2024a), which provide strong video understanding capability while maintaining relatively model design.

Following ReKV Di et al. (2025), we first evaluate our method in the offline setting on four widely used long-form VideoQA benchmarks: $\text{MLVU}_{\text{dev-mc}}$ Zhou et al. (2025), $\text{QAEgo4D}_{\text{test-mc}}$ Di & Xie (2024), EgoSchema Mangalam et al. (2023), and ActivityNet-QA Yu et al. (2019). These benchmarks cover complementary aspects of long-video understanding. Specifically, $\text{MLVU}_{\text{dev-mc}}$ evaluates fine-grained temporal reasoning across single-detail, multi-detail, and holistic questions. $\text{QAEgo4D}_{\text{test-mc}}$ focuses on egocentric video understanding with temporally grounded reasoning over annotated relevant segments. EgoSchema evaluates long-horizon reasoning over extended video contexts, while ActivityNet-QA assesses open-ended spatiotemporal reasoning through free-form answer generation. To further evaluate streaming video understanding capability, we additionally conduct experiments on the streaming benchmarks RVS-Ego, RVS-Movie Zhang et al. (2024a). RVS-Ego and RVS-Movie contain timestamped open-ended questions over long streaming videos, requiring models to incrementally process incoming frames and perform temporally grounded reasoning without access to future video content.

Table 1: Comparison on offline video understanding benchmarks.

Method	MLVU	QAEgo4D	EgoSchema	ActivityNet
GPT-4V Yang et al. (2023)	49.2	–	–	57.0
GPT-4o Hurst et al. (2024)	64.6	–	–	–
Gemini-1.5-Flash Team et al. (2024)	–	–	65.7	55.3
Gemini-1.5-Pro Team et al. (2024)	–	–	72.2	57.5
Video-ChatGPT-7B Maaz et al. (2024)	31.3	–	–	–
LLaMA-VID-7B Li et al. (2024b)	33.2	–	–	47.4
MiniGPT4-Video-7B Ataallah et al. (2024)	44.5	–	–	44.3
Video-LLaVA-7B Lin et al. (2024)	47.3	–	–	–
LongVA-7B Zhang et al. (2024c)	56.3	–	–	50.0
VideoStreaming Qian et al. (2024b)	–	–	44.1	–
Flash-VStream-7B Zhang et al. (2024a)	50.2	38.2	38.1	51.9
LLaVA-OV-0.5B Li et al. (2024a)	53.2	42.6	29.6	50.5
+ReKV Di et al. (2025)	54.1	50.0	31.0	52.4
+Ours	**54.4**	**50.6**	**31.4**	**52.9**
LLaVA-OV-7B Li et al. (2024a)	64.7	52.8	59.8	56.6
+ReKV Di et al. (2025)	68.5	55.6	60.2	59.8
+Ours	**69.2**	**56.6**	**60.9**	**60.2**

4.2 Implementation Details

Following the evaluation protocol of GPT-4o on MLVU Zhou et al. (2024), we uniformly sample video frames at 0.5 FPS for all benchmarks. During inference, incoming video frames are incrementally processed and stored in the KV-cache memory bank. The local attention window is set to 15K tokens for streaming video encoding. For Event-based Streaming Encoding, we adopt Dual TV-L1 optical flow Wedel et al. (2009) to perform online event segmentation for dynamically determining temporal boundaries of streaming memory units, where the maximum motion probe count is set to 64. For KNN-based KV-cache retrieval, the neighborhood graph is constructed based on cosine similarity between KV representations, and the number of nearest neighbors is set to 8. All experiments are conducted on NVIDIA L40S GPUs (48GB memory) using FP16 precision.

4.3 Offline Video Understanding

To validate the effectiveness of our method, we first evaluate it in the offline setting following the protocol of ReKV. Specifically, we conduct experiments on four widely used long-form VideoQA benchmarks and compare against state-of-the-art methods. As shown in Table 1, our method consistently improves the performance of both LLaVA-OV-0.5B and LLaVA-OV-7B across all benchmarks without additional training. Compared with ReKV, our approach achieves further gains, highlighting its effectiveness in improving encoding and retrieval quality. Notably, it also outperforms existing streaming VideoQA approaches, including memory-based methods such as VideoStreaming Qian et al. (2024b) and Flash-VStream Zhang et al. (2024a).

4.4 Streaming Video Understanding

We further evaluate our method in the streaming setting on the RVS-Ego and RVS-Movie Zhang et al. (2024a) benchmarks. During streaming inference, each query is issued immediately after its annotated end timestamp, and the model must generate answers based solely on the preceding video context. As shown in Table 2, our method consistently outperforms the ReKV baseline. Moreover, it enables LLaVA-OV-7B to surpass strong streaming baselines such as Flash-VStream Zhang et al. (2024a), demonstrating its effectiveness for streaming video understanding.

Table 2: StreamingVQA results on RVS-Ego and RVS-Movie.

Method	RVS-Ego		RVS-Movie	
	Acc.	Score	Acc.	Score
Flash-VStream-7B Zhang et al. (2024a)	57.3	4.0	53.1	3.3
LLaVA-OV-0.5B Li et al. (2024a)				
+ReKV Di et al. (2025)	54.7	3.9	44.6	3.4
+Ours	56.0	3.9	46.1	3.4
LLaVA-OV-7B Li et al. (2024a)				
+ReKV Di et al. (2025)	63.7	4.0	54.4	3.6
+Ours	64.7	4.0	56.8	3.6

Table 3: Ablation study on $\text{MLVU}_{\text{dev-mc}}$ for event-based encoding with LLaVA-OV-7B.

Method	Single Detail			Multi Detail		Holistic		Avg.
	Needle	Ego	PlotQA	Order	Count	Topic	Anomaly	
Uniform Encoding	74.65	66.29	76.07	46.72	35.92	90.11	74.00	68.46
Event Based Encoding	75.72	67.16	76.79	46.46	37.10	89.79	72.25	68.82

4.5 Ablation Studies

In this section, we conduct ablation studies to evaluate the effectiveness of Event-based Streaming Encoding and KNN-based KV-Cache Retrieval. We further provide sensitivity analyses on the maximum motion probe count and the number of nearest neighbors.

Effectiveness of Event-based Encoding. We conduct ablation experiments by incorporating the proposed Event-based Stream Encoding into the baseline method ReKV to validate its effectiveness. The results on $\text{MLVU}_{\text{dev-mc}}$, shown in Table 3, demonstrate that event-based encoding consistently improves VideoQA performance across different reasoning categories. Compared with uniform encoding, the proposed method achieves higher overall accuracy, suggesting that adaptive motion-aware memory partitioning reduces fragmented KV representations and provides more coherent temporal context for downstream reasoning.

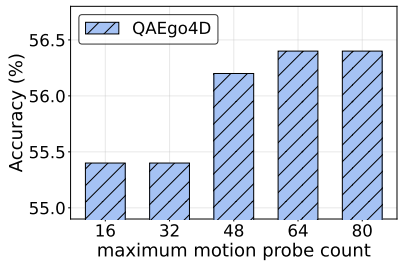
Effectiveness of KNN-based Retrieval. We further conduct ablation experiments to evaluate the effectiveness of the proposed KNN-based KV-cache Retrieval. As shown in Table 4, the proposed retrieval strategy achieves the best overall performance, outperforming both Clip-based retrieval and the original internal retrieval used in ReKV. The consistent improvements across different reasoning categories demonstrate that modeling neighborhood dependencies among KV-cache entries provides more effective historical context for downstream reasoning.

Maximum Motion Probe Count. We analyze the sensitivity of the maximum motion probe count used in Event-based Streaming Encoding. The motion probe count controls the temporal range used for estimating motion transitions during streaming video segmentation. Increasing the probe count allows the model to capture more stable temporal motion patterns, leading to more coherent memory partitioning. However, excessively large probe counts may reduce the sensitivity to local temporal changes and introduce over-smoothed event boundaries. As shown in Figure 4a, the performance improves as the number increases and achieves the best result when the probe count is set to 64.

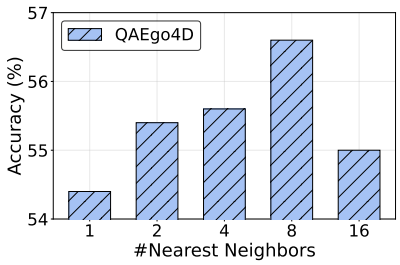
Number of Nearest Neighbors. We further analyze the sensitivity of the number of nearest neighbors in the proposed KNN-based retrieval mechanism. As shown in Figure 4b, the performance on QAEgo4D initially improves as the number of neighbors increases, indicating that incorporating neighborhood information helps recover more coherent contextual memory. The best performance is achieved when the number of nearest neighbors is set to 8. However, using too many neighbors leads to a performance drop, likely because excessive neighborhood expansion introduces irrelevant memory entries and noisy contextual information during retrieval.

Table 4: Ablation study on $\text{MLVU}_{\text{dev-mc}}$ for retrieval. The experiments are based on LLaVA-OV-7B.

Method	Single Detail			Multi Detail		Holistic		Avg.
	Needle	Ego	PlotQA	Order	Count	Topic	Anomaly	
Clip-based Retrieval	76.38	68.95	72.31	41.02	37.78	85.43	64.50	66.32
Internal Retrieval	74.65	66.29	76.07	46.72	35.92	90.11	74.00	68.46
KNN-based Retrieval	76.04	67.53	77.44	45.58	36.56	89.36	72.83	68.90



(a) Sensitivity analysis of the maximum motion probe count in the proposed Event-based Video Stream Encoding on QAEgo4D.



(b) Sensitivity analysis of the number of nearest neighbors in the proposed KNN-based KV-cache Retrieval on QAEgo4D.

Figure 4: Sensitivity analysis of key hyperparameters on QAEgo4D.

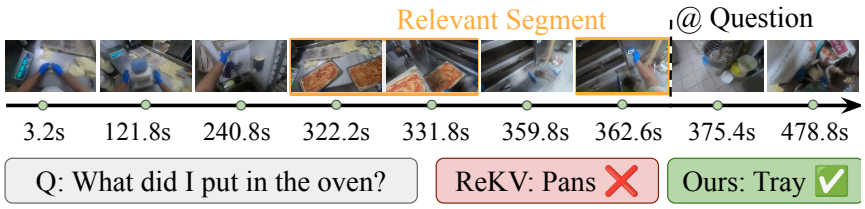


Figure 5: Qualitative comparison between ReKV and the proposed StreamVLM on streaming video question answering.

4.6 Qualitative Results

We further provide qualitative comparisons to analyze the effectiveness of the proposed StreamVLM framework for streaming video understanding. As shown in Figure 5, the question requires the model to retrieve the temporally relevant cooking event from a long streaming video. Existing retrieval methods such as ReKV fail to recover sufficient contextual information and incorrectly predict "Pans." In contrast, StreamVLM successfully retrieves the temporally coherent event segment corresponding to placing the tray into the oven, leading to the correct prediction "Tray."

5 Conclusion

In this paper, we present StreamVLM, a training-free framework for adapting offline VLMs to streaming video understanding. We reformulate streaming video understanding as a structured memory construction and retrieval problem, and identify two key limitations in existing streaming adaptations: fragmented memory construction caused by fixed uniform segmentation and incomplete contextual retrieval caused by independent query–KV matching. To address these challenges, we propose Event-based Streaming Encoding and KNN-based KV-Cache Retrieval, which improve temporal coherence during streaming memory construction and enable context-aware retrieval over historical KV-cache representations. Extensive experiments on both offline and streaming VideoQA benchmarks demonstrate that StreamVLM consistently improves video reasoning performance over existing streaming VLM baselines.

AI USE STATEMENT

We did not use generative AI tools for any research tasks requiring disclosure under the ICLR 2027 AI Policy for Authors. We used ChatGPT to edit the manuscript for clarity and readability and consulted a ChatGPT-generated image when selecting colors for the figures. The authors reviewed and revised all AI-assisted edits, made the final decisions about figure colors, and take responsibility for the final content of this paper.

REFERENCES

Moloud Abdar, Meenakshi Kollati, Swaraja Kuraparthi, Farhad Pourpanah, Daniel McDuff, Mohammad Ghavamzadeh, Shuicheng Yan, Abduallah Mohamed, Abbas Khosravi, Erik Cambria, et al. A review of deep learning for video captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Deyao Zhu, Jian Ding, and Mohamed Elhoseiny. Minigpt4-video: Advancing multimodal llms for video understanding with interleaved visual-textual tokens. *arXiv preprint arXiv:2404.03413*, 2024.

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.

Kirill Borodin, Kirill Kondrashov, Nikita Vasiliev, Ksenia Gladkova, Inna Larina, Mikhail Gorodnichev, and Grach Mkrtchian. Benchmarking compact vlms for clip-level surveillance anomaly detection under weak supervision. *Journal of Imaging*, 11(11):400, 2025.

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, 2024a.

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*, pp. 24185–24198, 2024b.

Shangzhe Di and Weidi Xie. Grounded question-answering in long egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12934–12943, 2024.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Tao Zhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video kv-cache retrieval. *arXiv preprint arXiv:2503.00540*, 2025.

Hang Hua, Yunlong Tang, Chenliang Xu, and Jiebo Luo. V2xum-llm: Cross-modal video summarization with temporal prompt instruction tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3599–3607, 2025.

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.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pp. 323–340. Springer, 2024b.

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*, pp. 5971–5984, 2024.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12585–12602, 2024.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

Mayug Maniparambil, Raiymbek Akshulakov, Yasser Abdelaziz Dahou Djilali, Mohamed El Amine Seddik, Sanath Narayan, Karttikeya Mangalam, and Noel E O'Connor. Do vision and language encoders represent the world similarly? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14334–14343, 2024.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, et al. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*, 2025.

Preeti Meena, Himanshu Kumar, and Sandeep Kumar Yadav. A review on video summarization techniques. *Engineering Applications of Artificial Intelligence*, 118:105667, 2023.

Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. Using an llm to help with code understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pp. 1–13, 2024.

Marius-Constantin Popescu, Valentina E Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multilayer perceptron and neural networks. *WSEAS transactions on circuits and systems*, 8(7): 579–588, 2009.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models, 2024a. URL https://arxiv.org/abs/2405.16009.

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, 2024b.

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*, pp. 24045–24055, 2025.

Allen H Renear. Text encoding. *A companion to digital humanities*, pp. 218–239, 2004.

Rui Shao, Wei Li, Lingsen Zhang, Renshan Zhang, Zhiyang Liu, Ran Chen, and Liqiang Nie. Large vlm-based vision-language-action models for robotic manipulation: A survey. *arXiv preprint arXiv:2508.13073*, 2025.

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.

Xiaodong Wang, Langling Huang, Zhirong Wu, Xu Zhao, Teng Xu, Xuhong Xia, and Peixi Peng. Livibench: An omnimodal benchmark for interactive livestream video understanding. *arXiv preprint arXiv:2601.15016*, 2026.

Andreas Wedel, Thomas Pock, Christopher Zach, Horst Bischof, and Daniel Cremers. An improved algorithm for tv-l 1 optical flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis: International Dagstuhl Seminar, Dagstuhl Castle, Germany, July 13-18, 2008. Revised Papers*, pp. 23–45. Springer, 2009.

Jiaqi Xu, Cuiling Lan, Wenxuan Xie, Xuejin Chen, and Yan Lu. Retrieval-based video language model for efficient long video question answering. 2023.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 2023.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 9127–9134, 2019.

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*, 2024a.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644, 2024b.

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. *arXiv preprint arXiv:2406.16852*, 2024c.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2):1–124, 2023.

Yue Zhao, Long Zhao, Xingyi Zhou, Jialin Wu, Chun-Te Chu, Hui Miao, Florian Schroff, Hartwig Adam, Ting Liu, Boqing Gong, et al. Distilling vision-language models on millions of videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13106–13116, 2024.

Yaoyao Zhong, Wei Ji, Junbin Xiao, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: Datasets, algorithms and challenges. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 6439–6455, 2022.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2(5):6, 2024.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13691–13701, 2025.