
Recurrent Attention-based Token Selection for Efficient Streaming Video-LLMs

Vaggelis Dorovatas^{1,2}

Soroush Seifi¹

Gunshi Gupta³

Rahaf Aljundi¹

¹Toyota Motor Europe

²Archimedes RU, Athena RC

³University of Oxford

Abstract

Video Large Language Models (Video-LLMs) excel at understanding videos in-context, provided they have full access to the video when answering queries. However, these models face challenges in streaming scenarios where hour-long videos must be processed online, and questions need timely responses. In this work, we propose a training-free approach compatible with standard Video-LLMs, leveraging three key concepts: 1) LLM-informed selection of visual tokens to identify those that the LLM has attended to and contributed to its understanding of each short clip. Our attention-based selection allows us to discard up to $\sim 95\%$ of unimportant visual tokens with minimal performance loss; 2) Recurrent processing of past selected tokens to generate temporally coherent understanding of each processed clip; 3) Caption-based question answering for lightweight and accurate responses. Our method achieves state-of-the-art performance on streaming video benchmarks, striking a balance between efficiency and effectiveness.

1 Introduction

Efficient and effective video question answering and understanding are crucial for deploying Large Language Models (LLMs) as intelligent assistants in domains requiring continuous visual input comprehension. This capability is essential for applications such as autonomous driving, surveillance, healthcare, and entertainment, where dynamic visual information must be understood in real-time. Current video understanding research with VLMs [31, 17, 2] typically processes entire videos in-context, presenting densely sampled frames alongside text queries to generate responses, with common training approaches focusing on text generation tasks like video captioning or question answering. While effective for short clips, this brute-force approach faces critical limitations with longer videos: computational costs escalate as visual tokens multiply, and exceeding context length limits forces sparse sampling that risks information loss. These challenges become particularly pronounced in streaming scenarios where continuous visual input renders full-video processing unsustainable. This necessitates efficient compression techniques that selectively process frames, condense visual information, and maintain easily retrievable memory structures for effective query response.

Recent efforts have focused on compressing the information from short video clips, moving away from the brute-force approach of processing the entire video in a single pass, and thus extending model capabilities to handle long video understanding. These approaches include methods that either compress only the visual information (before the LLM) encoded by a vision encoder [38, 12, 10], or store only textual descriptions of short clips [1], and retrieve only those relevant to the input

*First two authors provide contracted services for Toyota.

*Correspondence: vdorovatas@hotmail.gr

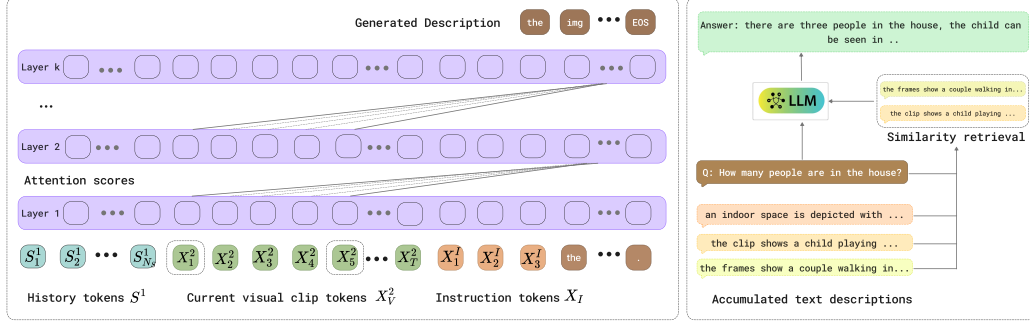


Figure 1: **Pipeline of rLiVS:** (Left): Long videos are broken down into short visual clips which are processed in a streaming fashion. A short clip is pre-pended with selected visual tokens from past clips in the stream to generate a textual description. History is accumulated by selecting a subset of tokens from the short clip based on attention scores. (Right): Given a query on the past video, the accumulated textual descriptions over a long video are compared to the query embedding and the most similar subset is input to the LLM (within the videoLLM) for question-answering.

query. Another line of work involves the LLM in video processing by generating compressed representations in the form of summarization tokens, like VideoStreaming [24], or storing KV-cache during inference, like ReKV [9], effectively combining the two modalities. While all these approaches mark important progress, they face amplified challenges in streaming settings, where past frames cannot be revisited and inputs can span hours. Training-based approaches [24, 38] face extrapolation issues with arbitrarily lengthy videos, and scaling their training can be computationally prohibitive or suffer from data scarcity and annotation issues for long videos. Training-free approaches like ReKV, while avoiding retraining, rely on storing large KV-caches, which becomes memory-intensive and introduces potential redundancy in streaming, ultimately affecting overall performance. Conversely, clip-wise caption-only methods (e.g., Goldfish [1]) offer efficiency but lack continuity, making it difficult to track entities and reason across clips, thus hindering holistic understanding.

In this work, we propose an efficient, training-free, and video LLM-agnostic solution for streaming video question answering. Our approach processes long videos (offline or online) by splitting them into short clips and focuses on three key aspects, integrating visual perception with language understanding for holistic video comprehension: 1) Compressing video information to avoid redundancy and inefficiencies in memory usage by selecting a small set of key visual tokens per short clip, 2) Accessing past selected visual tokens when processing new short clips, enabling visual information sharing across short clips and enhancing overall video comprehension, and 3) Performing text-based retrieval and answering, harnessing LLMs’ proven strength in reasoning over extended contexts (such as multi-clip captions).

Specifically, *during the processing* of each short clip (i.e., generating a caption), we draw inspiration from cognitive neuroscience, particularly the interplay between attention and memory [6]. Given memory’s limited capacity [23, 7], attention is key to selective encoding [6]. Accordingly, instead of randomly sampling visual tokens, we retain those that received the highest attention from the LLM, treating them as a compressed memory trace of the clip. Furthermore, building on the idea that past experiences shape current attention [6], we reuse previously selected tokens when processing subsequent clips. These past tokens are passed forward with new inputs, allowing memory to guide attention and support context-aware understanding. Inspired by the brain’s recurrent visual processing [15], we implement a simple FIFO memory: newly selected tokens are prepended, and the oldest are discarded once the context limit is reached.

Finally, *when answering* a question about past events in a video stream, we retrieve the top K captions based on the similarity of the question to the previously stored captions of short clips. This approach is motivated by the efficiency of storing captions alone and empirical evidence suggesting that long video reasoning is more effective with representative text. This approach enables us to answer questions based on the stored captions, rather than reprocessing the previous visual input, which is a common practice in current works [38, 10]. Figure 1 illustrates our method.

Our approach can be applied to any video-LLM pretrained on short clips, enabling long video understanding without additional training and maintaining a low memory footprint. We evaluate our method on three offline and streaming benchmarks, achieving state-of-the-art performance with significantly lower memory requirements. The simplicity and strong performance of our design set a robust baseline for online long-video question answering.

Our contributions are as follows:

- We propose Recurrent LLM-informed Visual Selection (rLiVS), a simple, training-free approach for long video understanding and question answering.
- Our approach is agnostic to the Video-LLM architecture and does not require any external modules.
- We achieve state-of-the-art performance with significantly lower memory requirements.

In the following, we discuss the closely related work in Section 2 and present our method in Section 3. We evaluate our method and ablate the design choices in Section 4 and conclude in Section 5.

2 Related Work

Video-Language Models: Recent video-language models have achieved strong performance on short-form video tasks by aligning visual features with text using pretrained vision-language transformers [30, 16, 19]. These models typically sample a small number of frames from each video and encode them into visual tokens using a vision backbone (e.g., ViT), which are then fed into a language model either directly or via cross-modal fusion. However, the number of visual tokens grows linearly with the number of frames and spatial patches—e.g., sampling 32 frames with a ViT-base backbone (16×16 patch size) yields over 1,000 tokens—making it prohibitively expensive to process long videos end-to-end [34, 29]. As a result, these models are limited to short clips (typically under 30 seconds), lack persistent memory, and cannot capture dependencies across longer temporal horizons without substantial computational and memory overhead.

Offline long-video understanding: To overcome the limitations of short-clip models, several offline approaches have been proposed for long video understanding by introducing mechanisms to compress or summarize information across time. Some methods focus on compressing only the visual modality, either by keyframe selection, feature pooling, or storing tokenized visual memory [32, 34, 28, 10]. Others rely solely on textual abstraction, converting short clips into natural language summaries that are later retrieved and composed to answer queries [1]. More recently, hybrid methods like VideoStreaming [24] train auxiliary models to generate trainable summarization tokens that condense the content of each clip into a compact representation, which is stored and retrieved during inference. While effective, these methods often require significant offline training to align the summarization with downstream tasks, are tied to specific pretraining pipelines, and typically process each clip independently—limiting their ability to build hierarchical or persistent memory across long videos.

Streaming and Memory-Augmented Video Understanding: In contrast to offline approaches, streaming video understanding aims to process incoming video frames incrementally, enabling real-time or low-latency inference over long temporal horizons. Retrieval-augmented methods maintain external memory stores of past embeddings and retrieve relevant context on-the-fly based on the current input, avoiding full-sequence reprocessing [38]. Similarly, MeMViT [35] incorporate visual memory modules or cross-modal attention over cached tokens to improve temporal coherence, though they often rely on fixed-length context windows or soft attention mechanisms, which scale poorly or suffer from memory drift over time. In the same direction of visual memory, Video-XL [27] introduces a custom-trained Video-LLM architecture that learns to compress and retain visual information over time through supervised training. However, its performance is tightly coupled to the specific model architecture and the distribution of video lengths and compression strategies seen during fine-tuning. Recently, ReKV [9] explored streaming video question answering by storing full decoder key-value caches and retrieving relevant context either through internal attention-based mechanisms or external CLIP-based similarity, demonstrating that retrieval over cached embeddings can be both effective and efficient. However, many of these methods either treat memory as a flat buffer or require storing large volumes of raw activations, limiting scalability and longer-term abstraction. Our work builds on these insights by introducing a memory consolidation mechanism that persistently retains and hierarchically abstracts multimodal context across a stream, enabling scalable and task-adaptive long-term understanding.

3 Method

We aim to develop an efficient method for streaming scenarios. Our approach uses a short-clip video-LLM as the backbone, which is capable of multimodal processing of short video clips and generating textual responses. Specifically, for the general captioning task, the model, given a sequence of visual tokens (representing the input frames, V) extracted from a visual encoder (eg. CLIP [25]) and a potential instruction ("Describe what is happening in the video."), generates a textual description C , representing the model's understanding of the short clip. Rather than storing all the information (as in [9]) we aim to compress the visual information by leveraging the relation with the textual tokens of each short clip, producing a compressed visual-guided-by-text representation, S . We further target a model-agnostic design differently from [11] and [24]). The key aspect here is that we have the input to the LLM, V , and its corresponding caption, C , which can be interpreted as the model's understanding expressed in text.

3.1 Attention based Visual Token Selection

Several works have explored token selection based on attention weights [37, 13, 26]. These methods typically involve dropping tokens from early layers to enhance the efficiency of processing long contexts in later layers. Evidence from these studies suggests that a model's attention can serve as a strong indicator of the relevance of visual tokens.

Our goal is to globally select tokens from a short clip after it has been processed by the VideoLLM, providing context for subsequent clips. These tokens will guide the LLM's attention in the next short clip to focus on content that is coherent with past events, thereby reducing attention to background tokens, noise, and redundant visual tokens.

Short Clip Processing. We employ a pre-trained video-LLM, which consists of a visual encoder VE, an LLM and a projector module P. Given a T -frame video clip V and an instruction X_I , outputs a textual caption of the video, C :

$$X_V = P(VE(V)), \quad C = LLM(X_V, X_I) \quad (1)$$

where $X_V \in \mathbb{R}^{TN_V \times D}$ and N_V denotes the number of spatial tokens per frame, and D is the dimension of the LLM tokens. The target is to select a sparse set of tokens $S \in \mathbb{R}^{T \times N_S \times D}$ where $N_S \ll N_V$.

Attention Coefficients Calculation. Let us denote X_C the embedding of the generated caption. By the last generated token, attention coefficient tensor is estimated between the generated text tokens and the input tokens. We define the attention matrix $A^{l,h} \in \mathbb{R}^{(TN_V+N_I+N_C) \times (TN_V+N_I+N_C)}$ calculated on $X^l = [X_V^l, X_I^l, X_C^l]$ in a given layer l and a given attention head h where N_I is the number of text tokens in the instruction prompt and N_C is the number of tokens in the number of generated caption tokens.

After transferring X^l into Key $K^{l,h}$, Value $V^{l,h}$ and Query $Q^{l,h}$, the attention scores $A^{l,h}$ are computed using the scaled dot-product attention:

$$Q^{l,h} = X^l W_Q^{l,h}, \quad K^{l,h} = X^l W_K^{l,h}, \quad V^{l,h} = X^l W_V^{l,h} \quad (2)$$

where $W_Q^{l,h}, W_K^{l,h}, W_V^{l,h}$ are the learnable weight matrices for queries, keys, and values, respectively¹.

$$A^{l,h} = \text{Softmax} \left(\frac{Q^{l,h} (K^{l,h})^\top}{\sqrt{d_k}} \right) \quad (3)$$

where d_k is the dimension of attention heads.

First we extract the attention coefficients representing the cross attention between the generated caption and the input visual tokens:

$$A_V^{l,h} = A^{l,h}[TN_V + N_I : TN_V + N_I + N_C, 0 : TN_V] \quad (4)$$

¹Our method focuses on attention weights, making operations involving the values $V^{l,h}$ unnecessary and thus discarded.

We then calculate an attention score for each input visual token $\mathbf{X}_{V,j,:}$ by averaging the attention scores from all generated caption tokens N_C . Finally, the global importance score for a given visual token $\mathbf{X}_{V,j,:}$ is determined by averaging the attention scores across different attention heads H and layers L , representing the overall attention of the model.

$$a_j = \frac{1}{L} \sum_{l=1}^L \frac{1}{H} \sum_{h=1}^H \left(\frac{1}{N_C} \sum_{i=1}^{N_C} \mathbf{A}_{V,ij}^{l,h} \right) \quad (5)$$

In practice, to limit the compute cost, we consider only a subset of layers. Note that instead of averaging considering other operations such as max pooling is straight forward.

Selecting Top Tokens: Finally, the top N_S tokens of $\mathbf{X}_V$ with the highest attention coefficients are selected:

$$\mathbf{S} = \mathbf{X}_V[\pi(1), \pi(2), \dots, \pi(N_S), :] \quad (6)$$

where $\pi = \text{argsort}(\mathbf{a})$ are the indices that sort $\mathbf{a}$ in descending order².

3.2 Long Video Recurrent Processing.

For the very first short clip $\mathbf{X}_V^{(0)}$ we have selected a set of N_S tokens $\mathbf{S}^{(0)}$ receiving the highest attention from the generated caption. Now for a next shortclip $\mathbf{X}_V^{(1)}$, we provide as input to the LLM the previously selected tokens $\mathbf{S}^{(0)}$ as input along with the full set of tokens comprising the current video clip $\mathbf{X}_V^{(1)}$. Similarly, after generating a caption $C^{(1)}$ we select a sparse set of most relevant tokens receiving highest attention scores from $\mathbf{X}_V^{(1)}: \mathbf{S}^{(1)}$.

For the following video shortclips, we create a FIFO queue of past selected tokens $[\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \dots, \mathbf{S}^{(t)}]$ which are provided to the LLM as context along with the next short clip raw tokens $\mathbf{X}_V^{(t+1)}$ up to the limit of the context window length W or compute constraint.

3.3 Efficient Video Question Answering

We have illustrated how to process each video clip tokens $\mathbf{X}_V^{(t)}$ representing T frames while having access to a long window of past selected tokens. During this process a caption $C^{(t)}$ is generated, capturing the LLM thoughts and understanding about the current clip events conditioned on past clips provided in the context window. We store these generated captions embeddings $\{\mathbf{X}_C^{(t)}\}$ and given a question q , we compute the average cosine similarity between the question tokens $\mathbf{X}_q$ and the tokens of each stored caption $\{\mathbf{X}_C^{(t)}\}$. Instead of retrieving the top K captions based solely on average cosine similarity to the query q , we use Maximal Marginal Relevance (MMR) [4], a common retrieval technique where a score is calculated by balancing relevance to the query (cosine similarity between caption and query) with diversity among the selected captions (cosine similarity between candidate caption and already selected ones). Utilizing this technique, we effectively reduce potential redundancy that can emerge in recurrent caption generation, while promoting coverage of distinct information when answering. Algorithm 1 summarizes our approach.

Our simple design and reliance on generated captions for video question answering is stemmed from the efficiency of storing the caption tokens IDs and the utility of similarity estimation between textual

Algorithm 1 Streaming Video Processing and Query Answering with rLiVS

Streaming Video Processing

```

1:  $M_l \leftarrow [], M_s \leftarrow \text{queue}(), B \leftarrow []$ 
2:  $\text{MAX\_MEM} \leftarrow 16, \text{CLIP\_SIZE} \leftarrow 16$ 
3: while frames available do
4:    $B.append(\text{get\_next\_frame}())$ 
5:   if  $\text{length}(B) == \text{CLIP\_SIZE}$  then
6:      $\text{context} \leftarrow M_s + B$ 
7:      $B.clear()$ 
8:      $S, C \leftarrow \text{Attn\_Selection}(\text{context})$ 
9:     if  $\text{length}(M_s) == \text{MAX\_MEM}$  then
10:       $M_s.pop\_left()$ 
11:     end if
12:      $M_s.append(S)$ 
13:      $M_l.append(C)$ 
14:   end if
15: end while

##### Query Answering


16:  $Q \leftarrow \text{embed}(\text{query})$ 
17:  $C' \leftarrow \text{Retrieve\_TopK}(Q, M_l)$ 
18:  $\text{context} \leftarrow C' + Q$ 
19:  $\text{answer} \leftarrow \text{LLM\_Generate\_Answer}(\text{context})$ 

```

²Basically, we select the top N_S visual tokens (based on the attention scores), and store them in their original temporal order.

tokens without the reliance on any external embedding encoder such as CLIP [25]. Although we can store and provide both visual tokens $\mathbf{X}_V^{(t)}$ and captions $\mathbf{X}_C^{(t)}$ to the LLM for question answering, our experiments show that video-LLMs perform better with only caption input.

4 Experiments

In this section, we evaluate our proposed method against state-of-the-art approaches in streaming and long video understanding benchmarks, demonstrating strong performance and notable efficiency across experiments.

Evaluation Benchmarks. We evaluate our method’s effectiveness in online scenarios using the Realtime VStream-QA benchmark [38], which includes *RVS-Movie* (emphasizing plot understanding) and *RVS-Ego* (focusing on visual comprehension)—both featuring 40-minute videos with diverse open-ended questions. To demonstrate robustness, we also report results on offline benchmarks: *MovieChat* [28] (170 videos averaging 576 seconds across various genres with 510 questions testing long-range comprehension), offline VStream-QA (*VS-Movie* and *VS-Ego*), and *CG-Bench* [5] (1,219 videos averaging 27 minutes with 12K multiple-choice questions). Additionally, we conduct an ablation on *NextQA*-valset [36] (570 shorter videos averaging 44 seconds with 5K multiple-choice questions) to validate our attention-based visual token selection approach.

Baselines. We compare our method against established video LLMs and recent long video understanding approaches. Training-required baselines include: VideoChatGPT [22] (employing temporal-spatial pooling), Chat-UniVi [14] (offering unified image/video representations through a set of dynamic visual tokens), LLaMA-VID [21] (using learning-based compression with context/content tokens per frame), VideoScan [20] (compressing frames into single learned semantic tokens preserving temporal context), and Flash-VStream-7B [38] (processing frame-by-frame with hierarchical memory including FIFO spatial memory, weighted k-means temporal consolidation, and past-current information abstraction). All operate on non-overlapping sliding windows. Training-free baselines include: MovieChat [28] (using FIFO short-term memory and similarity-based long-term consolidation), Goldfish [1] (processing independent short clips with caption generation and retrieval), and ReKV [9] (storing full KV-Cache retrieved through cross-attention to queries). We coin our method rLiVS as a short for recurrent LLM-informed Visual Selection.

Evaluation Metrics. For benchmarks with open-ended questions, we report both Accuracy and a Score based on GPT-3.5 evaluation, following prior works (we use gpt-3.5-turbo-0613, consistent with the evaluation setup of ReKV [9]). The model is given the question, ground truth, and prediction, and asked to assess whether the prediction matches the answer ("Yes/No") and to assign a compatibility score. The prompt template is provided in the Appendix. For multiple-choice benchmarks, we compute accuracy by directly comparing predictions to ground truths. Additionally, for the streaming benchmark, we report end-to-end latency, GPU usage (from query arrival to response), and KV-Cache utilization. Benchmarkings are conducted on a single A100 GPU.

Implementation Details. We implement our method on LLaVA-OneVision [17], a strong VLM for image and video tasks, enabling direct comparison with ReKV (current state-of-the-art on RVS benchmarks). We demonstrate versatility by evaluating both 7B and 0.5B variants, with 7B used unless specified otherwise. Since LLaVA-OV is trained with 32 frames (196 visual tokens each), we allocate 16 frames for current short clips and 16 for short-term recurrent memory. We select only 196 visual tokens from 3,136 available (16×196), retaining just 6.25% of total visual information per short clip. Following previous works, we process RVS-Movie and RVS-Ego at 0.5 FPS [38, 9], MovieChat at 1 FPS, and CG-Bench and offline VS-Stream at 0.5 FPS, with 10K context tokens for retrieval and generation. We average attention scores from 4 (of 28) backbone layers across experiments. To show generalization across video-LLM backbones of our streaming video pipeline, we also incorporate rLiVS to Qwen2.5-VL-7B [3] and test on RVS streaming benchmark. We provide details for this setup in Appendix A. Finally, in the Appendix we include ablations on layer selection (Appendix B), as well as on attention aggregation methods (average vs. max), and analysis of how context length affects performance and latency in Appendices C and D, respectively.

4.1 Results

We next present results across the discussed benchmarks: starting with an ablation on visual selection methods for short video understanding, followed by evaluations on offline long video QA, and concluding with streaming video understanding.

Table 1: Comparison on offline long video benchmarks. Accuracy (Acc.) and Score (Sco.) are reported where applicable. Results for ReKV are taken from [9], for VS-Stream from [38], for Moviechat from [28] and [1], while for CG-Bench from [5].

Method	VS-Ego		VS-Movie		MovieChat		CG-Bench
	Acc.	Sco.	Acc.	Sco.	Acc.	Sco.	Acc.
Video-ChatGPT [22]	51.7	3.7	54.4	3.4	47.6	2.5	-
MovieChat [28]	52.2	3.4	39.1	2.3	62.3	3.2	-
Chat-UniVi [14]	50.9	3.8	54.0	3.4	-	-	<u>25.9</u>
LLaMA-VID [21]	54.8	3.9	51.4	3.4	53.2	3.8	-
Goldfish [1]	-	-	-	-	67.6	4.2	-
Flash-VStream-7B [38]	<u>59.0</u>	3.9	<u>56.1</u>	3.4	-	-	-
rLiVS (Ours)	61.0	3.9	59.3	3.6	78.0	4.0	33.1

4.1.1 Evaluation of Visual Token Selection Strategies on Short Video Datasets

Visual token selection is central to our method, as it brings past information into current short-clip processing via recurrency, enhancing captioning and understanding of the overall video stream. We argue that naive methods such as uniform sampling or mean pooling risk discarding important information, while traditional clustering approaches like K-Means can be slow and sub-optimal in high-dimensional spaces. Instead, by leveraging the LLM’s attention—already computed during caption generation—we gain a powerful, low-overhead signal that enables aggressive compression with minimal performance loss. To empirically validate this, we conduct experiments on the NextQA-valset, a short video benchmark, to assess the token’s selection quality independently from the factors of long video understanding. We compare the full model’s performance while retaining only 6% or 12% of visual tokens—selected with different methods:

1) via uniform sampling, 2) via mean pooling, 3) via K-Means and 4) based on our attention selection mechanism to the generated caption, as described in Section 3.1. Table 2 confirms our intuition; while uniform sampling results in 2-3% performance loss, our attention-based selection outperforms uniform sampling even at twice the compression rate—incurring only a 1.5% performance drop with 6% of visual tokens, and nearly matching full-model performance at 12%. Moreover, mean pooling, naively combining visual tokens without considering their relevance, yields the worst performance, while K-Means performs comparably to attention-based selection but it is significantly slower—especially when operating in the high-dimensional input space of the LLM—introducing unnecessary computational overhead.

Table 2: Accuracy (Acc.) of different selection methods evaluated on NextQA-valset.

Selection Method	Next-QA (valset) Acc.
Full Model	78.6
Uniform Sampling (6%)	75.5
Mean Pooling (6%)	70.7
K-Means (6%)	76.8
Attention (ours) (6%)	77.0
Uniform Sampling (12%)	76.7
Mean Pooling (12%)	75.5
K-Means (12%)	78.1
Attention (ours) (12%)	78.4

4.1.2 Offline Video Question Answering Datasets

In Table 1 we present results on offline long video understanding benchmarks. rLiVS incorporated in LLaVA-OV-7B achieves state-of-the-art performance in VS-Ego and VS-Movie benchmarks, outperforming the previous best by 2% and 3%, respectively. On MovieChat and CG-Bench, our method achieves strong results, outperforming prior approaches, demonstrating the effectiveness of heavily compressing visual tokens without harming performance. Overall, these results indicate that our method serves as a strong baseline, efficiently handling long inputs, requiring no additional training or fine-tuning—while maintaining minimal memory storage requirements.

Table 3: Evaluation on the Realtime VStream-QA streaming benchmark [38], consisting of RVS-Ego and RVS-Movie subsets. Accuracy (Acc.), Score (Sco.), Latency, VRAM and KV-Cache usage are reported. Results for baselines are taken from [20].

Method	RVS-Ego		RVS-Movie		Latency	VRAM	KV-Cache
	Acc.	Sco.	Acc.	Sco.			
MovieChat [28]	50.7	3.4	36.0	2.3	-	-	-
LLaMA-VID [21]	53.4	3.9	48.6	3.3	-	-	-
Flash-VStream-7B [38]	57.3	4.0	53.1	3.3	2.1s	19GB	-
VideoScan [20]	60.9	4.0	54.1	3.5	2.1s	18GB	-
<i>LLaVA-$\overline{OV}$ 0.5B</i>							
↪ ReKV [9]	54.7	3.7	44.6	3.4	1.6s	19GB	4.0 GB/h
↪ rLiVS (Ours)	57.6	3.8	51.3	3.4	1.5s	11GB	-
<i>LLaVA-$\overline{OV}$ 7B</i>							
↪ ReKV [9]	63.7	4.0	54.4	3.6	2.7s	36GB	18.8 GB/h
↪ rLiVS (Ours)	<u>65.3</u>	4.0	57.7	3.6	1.9s	25GB	-
<i>Qwen2.5-VL 7B</i>							
↪ rLiVS (Ours)	68.1	4.0	<u>56.1</u>	3.6	2.7s	19GB	-

4.1.3 Streaming Video Question Answering Datasets

Finally, we present our results on RVS-Ego and RVS-Movie, long video streaming benchmarks that represent the primary focus of this work and the main setting for evaluating our proposed method. rLiVS demonstrates strong suitability for this setting as shown in Table 3, outperforming the previous best, ReKV, with the same backbone model, by 2–3% and *achieving new state-of-the-art results on the benchmark, while also being nearly 1 second faster than ReKV*. Importantly, our attention-based compression technique—combined with recurrency and caption-guided retrieval and answering—proves robust across both subsets of the streaming benchmark. Unlike ReKV, rLiVS does not require external memory offloading and operates within a 10K context window, enabling both efficient and effective performance, while resulting in 11GB less peak VRAM usage with the same video-LLM backbone. Interestingly, when paired with the lightweight 0.5B model, our method outperforms the previous second-best approach that uses a 7B model, as well as ReKV with the same 0.5B model by 2.9% on RVS-Ego and 6.7% on RVS-Movie, highlighting the strength of caption-based answering even in smaller model variants emphasizing the efficient aspect of our design choices. Finally, by coupling our method with the recent strong multimodal LLM Qwen2.5-VL, we further improve performance on RVS-Ego, reaching 68.1% accuracy. *This highlights the model-agnostic and plug-and-play nature of our proposed rLiVS.*

A key strength of our framework is its ability to adapt token selection dynamically based on task instructions, which serve as top-down signals. Through instruction-guided attention, the model selects a sparse subset of semantically relevant visual tokens, enabling efficient, context-aware understanding. As shown in Appendix H, token selection is highly sensitive to the instruction, with different tasks yielding distinct attention patterns. This adaptivity supports a broad range of video understanding tasks without retraining or architectural changes, highlighting promising directions for scalable and generalizable long video models.

4.1.4 Ablation of rLiVS Design Choices

Next, we ablate our method’s core design choices through targeted experiments, demonstrating the following: (1) the importance of recurrency when processing long videos split in short clips; (2) the superiority of captions over selected visual tokens—or their combination—for retrieval and answering in long video streams, (3) the performance benefits of LLM-

Table 4: Ablation on the effect of accessing past visual information via recurrency and attention selection.

Method	RVS-Ego		RVS-Movie		MovieChat	
	Acc.	Sco.	Acc.	Sco.	Acc.	Sco.
rLiVS	65.3	4.0	57.7	3.6	78.0	4.0
w/o recurrency	62.5	3.9	53.7	3.5	74.1	3.9

informed attention-based selection compared to naive uniform sampling in streaming scenarios; and (4) the effect of compression rate on downstream performance.

The importance of recurrency in long streams. As previously discussed, [1] proposes processing long videos as independent short clips, storing only the textual outputs (captions) generated by a video-LLM and retrieving relevant information based on cosine similarity to the input question. While this approach is simple and efficient, it suffers from a lack of continuity that becomes increasingly problematic as video length increases. Specifically, the absence of shared visual information across clips and sole reliance on textual representations can hinder the model’s ability to consistently track entities—such as people or objects—across time, especially when captions lack distinguishing context. These limitations naturally impact performance. To empirically prove this, we present results (Table 4) on RVS-Ego, RVS-Movie, and MovieChat, showing that introducing recurrency consistently improves performance by 3–4%, under otherwise identical conditions. Importantly, recurrency in our method serves a dual purpose: (1) it enhances continuity and coherence across short clips, boosting long video understanding, and (2) it guides the LLM’s attention during visual token selection, reinforcing its own impact.

Captions vs Visual Tokens for *retrieval* and *answering*. A natural question is whether textual information (captions) or selected visual tokens provide a better signal for retrieving and answering user queries—that is, which representation is the best choice for effectively preserving crucial information while enabling compression. **For retrieval**, the input query is embedded into the LLM’s input space. Visual tokens, projected ad hoc from the visual encoder, lack learned semantic alignment with queries, leading to less meaningful similarity scores (as illustrated in Appendix F). Proper similarity between them would require projecting both queries and visual tokens into a common space (like CLIP encoders [25]), adding extra computational steps to our pipeline. In contrast, captions share the same modality as queries and can be directly embedded to the LLM’s input space, making them naturally better suited for retrieval via simple cosine similarity. **For answering**, Table 5 presents results using visual tokens, captions, or their combination as retrieved information for each short clip. Among the approaches, using only captions achieves the best performance. Despite visual tokens theoretically offering richer information, they under-perform likely due to the mismatch between video-LLM training data (predominantly short clips of a few seconds) and evaluation contexts (minutes/hour steaming videos). Current video-LLMs struggle to comprehend visual information at larger timescales, even when computationally representable as input. This aligns with findings from [33], which demonstrated that despite the existence of architectural optimizations that enable long video processing (through memory structures or compressed representations), video-LLMs still demonstrate significant performance degradation in these contexts. In contrast, LLM research has extensively explored long-context adaptation, developing effective modeling and training techniques that enable reasoning over extended sequences of textual inputs, such as multi-clip captions. Thus by utilizing only the captions as past information during retrieval and answering, we effectively transform the problem into a text-based long context QA problem, harnessing LLMs’ proven strength in reasoning over these extended contexts.

Attention-based vs Uniform Visual Selection on streaming scenarios. We previously demonstrated (Sec. 4.1.1) that our attention-based selection outperforms uniform sampling in short video QA, even at twice the compression rate, enabling aggressive token reduction with minimal performance loss. We now extend this evaluation to the streaming setting, where selection drives captioning (through recurrency) and thus plays a critical role in overall performance. Our results in Table 6 are consistent with those

Table 5: Comparison of vision vs. language modalities for information retrieval and query response on RVS-Ego/Movie benchmarks.

Retrieved Modality	RVS-Ego		RVS-Movie	
	Acc.	Sco.	Acc.	Sco.
Selected Visual Tokens	58.2	3.9	48.4	3.5
Captions	65.1	4.0	57.7	3.6
Combination	63.0	4.0	54.3	3.5

Table 6: Comparison of selection methods within our pipeline on RVS-Ego and RVS-Movie.

Selection Method	RVS-Ego		RVS-Movie	
	Acc.	Sco.	Acc.	Sco.
Uniform Sampling	64.2	3.9	56.0	3.5
Attention (ours)	65.1	4.0	57.7	3.6

in the short clip benchmark: attention-based selection, leveraging LLM’s ability to attend to crucial information, effectively filtering out noise and redundancy, is superior in both subsets by 1-2%, without introducing any additional overhead, since caption generation is part of our pipeline. Note that uniform sampling is integrated into our full pipeline, replacing only the selection strategy. This approach serves as a significantly stronger baseline compared to the uniform sampling baseline of ReKV [9], as demonstrated in the Appendix.

Compression Rate and Downstream Performance.

Our choice of aggressive visual token compression (e.g., retaining only 6%) may raise concerns about potential information loss and downstream performance degradation. However, this reflects a practical trade-off in real-time video understanding. Long-form videos often contain substantial visual redundancy, allowing for effective token filtering without harming performance. To validate this, Table 7 shows how accuracy varies with different compression rates on the NextQA validation set. Retaining 12% of visual tokens nearly matches full performance, while 6% leads to negligible loss³. Our framework remains flexible, with the compression ratio tunable to task requirements.

Looking ahead, adaptive compression—dynamically adjusting retention based on visual content complexity—is a promising direction. While it may improve performance in visually rich scenes, it introduces computational overhead, which we currently avoid to maintain online efficiency. Qualitative results in Appendix J further demonstrate the robustness of our method under extreme compression.

Table 7: Acc. on NextQA valset for different % of visual tokens selected with our proposed attention-based method.

Selected Tokens	Acc.(%)
1%	68.0
6%	77.0
12%	78.4
19%	78.7
25%	79.0
100%	78.6

5 Discussion

Streaming video question answering and understanding is an emerging landscape that can enable the reliable deployment of intelligent agents in various domains requiring visual comprehension. A key design requirement for practical solutions is low memory and compute usage. In this work, we propose a simple approach based on two design choices: first, the instant selection of visual tokens based on the LLM’s attention, resulting in $\sim 95\%$ compression rate of short clip content; second, the use of recurrent processing and caption based question answering to smoothly enable higher-level reasoning over longer time spans. With these components, we achieve state-of-the-art performance on streaming video benchmarks. Our approach is agnostic to the video-LLM deployed and requires no training, setting a strong baseline for future work.

Beyond question answering, our framework shows potential as a general-purpose backbone for long-form video processing and understanding tasks such as summarization, retrieval, and object tracking. Preliminary results on long video summarization (Appendix I) provide initial evidence of the method’s generalizability. Additionally, our analysis in Appendix H demonstrates that the memory selection mechanism is both task-sensitive and instruction-conditioned, underscoring its flexibility and adaptability. We view this as a promising direction for future research, where natural language task descriptions can dynamically guide memory selection and reasoning.

Limitations. Despite strong performance, our method prioritizes efficiency over completeness by attending only to selected content during short clip processing. This may overlook fine-grained details and harm continuity, as the LLM lacks access to the full video history. While our FIFO memory buffer offers short-term context and captions serve as long-term memory, this temporal mechanism may not always capture semantically salient information. A promising direction is to explore more semantic, rather than purely temporal, memory selection strategies that remain efficient.

Our approach is also training-free and fully reliant on pre-trained backbones, inheriting their limitations in visual understanding and temporal reasoning. The recurrent captioning process, conditioned on prior frames and instructions, can introduce redundancy across captions, which may affect retrieval performance. We partially address this by considering both novelty and relevance during retrieval. Future work includes integrating the method into training pipelines and evaluating across a wider range of models and benchmarks to better understand its strengths and limitations.

³An additional benefit is that selecting only 6% of tokens allows more memory slots in the FIFO queue, enhancing long-term understanding.

References

- [1] 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] 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.
- [3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- [4] Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. Simple unsupervised keyphrase extraction using sentence embeddings. In Anna Korhonen and Ivan Titov, editors, *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 221–229, Brussels, Belgium, October 2018. Association for Computational Linguistics.
- [5] Guo Chen, Yicheng Liu, Yifei Huang, Yuping He, Baoqi Pei, Jilan Xu, Yali Wang, Tong Lu, and Limin Wang. Cg-bench: Clue-grounded question answering benchmark for long video understanding. *arXiv preprint arXiv:2412.12075*, 2024.
- [6] Marvin M Chun and Nicholas B Turk-Browne. Interactions between attention and memory. *Current Opinion in Neurobiology*, 17(2):177–184, 2007. Cognitive neuroscience.
- [7] Nelson Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24:87 – 114, 2001.
- [8] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.
- [9] Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, TaoZhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video KV-cache retrieval. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [10] Gueter Josmy Faure, Jia-Fong Yeh, Min-Hung Chen, Hung-Ting Su, Shang-Hong Lai, and Winston H Hsu. Hermes: temporal-coherent long-form understanding with episodes and semantics. *arXiv preprint arXiv:2408.17443*, 2024.
- [11] Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*, 2023.
- [12] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-Imm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13504–13514, 2024.
- [13] Xiaohu Huang, Hao Zhou, and Kai Han. Llm-vtp: Llm-reasoned visual token pruning for efficient multi-modal video understanding.
- [14] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13700–13710, 2024.
- [15] Victor A.F. Lamme and Pieter R. Roelfsema. The distinct modes of vision offered by feedforward and recurrent processing. *Trends in Neurosciences*, 23(11):571–579, 2000.
- [16] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7331–7341, 2021.

- [17] 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*, 2024.
- [18] Haopeng Li, Andong Deng, QiuHong Ke, Jun Liu, Hossein Rahmani, Yulan Guo, Bernt Schiele, and Chen Chen. Sports-qa: A large-scale video question answering benchmark for complex and professional sports. *arXiv preprint arXiv:2401.01505*, 2024.
- [19] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.
- [20] Ruanjun Li, Yuedong Tan, Yuanming Shi, and Jiawei Shao. Videoscan: Enabling efficient streaming video understanding via frame-level semantic carriers. *arXiv preprint arXiv:2503.09387*, 2025.
- [21] 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*, pages 323–340. Springer, 2024.
- [22] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.
- [23] George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63(2):81–97, March 1956.
- [24] 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.
- [25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.
- [26] Zhenmei Shi, Yifei Ming, Xuan-Phi Nguyen, Yingyu Liang, and Shafiq Joty. Discovering the gems in early layers: Accelerating long-context llms with 1000x input token reduction. *arXiv preprint arXiv:2409.17422*, 2024.
- [27] Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26160–26169, 2025.
- [28] Enxin Song, Wenhao Chai, GuanHong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024.
- [29] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.
- [30] Maria Tsimgpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.
- [31] Jiawei Wang, Liping Yuan, Yuchen Zhang, and Haomiao Sun. Tarsier: Recipes for training and evaluating large video description models. *arXiv preprint arXiv:2407.00634*, 2024.
- [32] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14549–14560, 2023.

- [33] Weihang Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, et al. Lvbench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*, 2024.
- [34] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.
- [35] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition, 2022.
- [36] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9777–9786, 2021.
- [37] Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22128–22136, 2025.
- [38] 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.
- [39] 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 Computer Vision and Pattern Recognition Conference*, pages 13691–13701, 2025.

NeurIPS Paper Checklist

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 explain the contributions as shown in method and experiments.

Guidelines:

- The answer NA 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 NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

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

Answer: [\[Yes\]](#)

Justification: In Section.5.

Guidelines:

- The answer NA 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: [\[NA\]](#) .

Justification: We do not have theoretical results.

Guidelines:

- The answer NA 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: in the main paper and the Appendix we list all details needed.

Guidelines:

- The answer NA 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: We will release the code after finalizing the patent.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: in Section.4 and in the Appendix.

Guidelines:

- The answer NA 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: due to the computational cost for running each experiments.

Guidelines:

- The answer NA 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: In the Appendix.

Guidelines:

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

9. Code of ethics

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

Answer: [Yes]

Justification: We conform to the NeurIPS Code of Ethics. We include details to reproduce the method and we will add model cards once released.

Guidelines:

- The answer NA 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: We include a broader impact section in the Appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We do not train new models or release new data.

Guidelines:

- The answer NA 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: yes we cite the data we use and models and check the license.

Guidelines:

- The answer NA 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: [NA]

Justification: we do not release new assets.

Guidelines:

- The answer NA 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: [NA]

Justification: not involved.

Guidelines:

- The answer NA 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: [NA]

Justification: No crowdsourcing involved.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLMs are only used for checking text formatting.

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Technical Appendices and Supplementary Material

A Experimental Details

In this section, we provide a detailed description of our evaluation setup and methodological choices to ensure the reproducibility of our experiments:

- *Visual Tokens Selection:* As described in the main paper, our attention-based selection method selects k out of N visual tokens globally. The backbone video-LLM we use (LLaVA-OV) is trained on 32 video frames, each contributing 196 tokens, resulting in a total of 6,272 visual tokens. We split the 6,272-token context window into two halves: the first is used to encode past information (by injecting previously selected visual tokens), and the second is reserved for the raw visual tokens of the current short clip. We choose k such that it evenly divides the 3,136 tokens of the second half (current short clip) in order to fully utilize the context. During selection, we consider only on the second half, corresponding to the current short clip (i.e., 3,136 tokens). For both long-video offline and streaming benchmarks, we select 196 tokens from this set, which is $\frac{1}{16}$ (or approximately 6.25%) of the total. For simplicity, we report this as 6% throughout the paper. In our uniform baseline, everything remains the same and we just sample 196 out of the 3,136 tokens of the current short clip.
- *Resources:* Across all experiments, we utilize one or two NVIDIA A100 GPUs with 40GB of memory. Latency and VRAM measurements are conducted on a single GPU.
- *LLM Evaluation:* As reported in the main paper, we follow prior work [9, 38] for LLM-based evaluation and use gpt-3.5-turbo. The prompt templates employed are provided in Section K.
- *Qwen2.5-VL setup:* This model dynamically embeds frames based on input resolution, resulting in different number of tokens per frame. In our experiments, we use the same number of tokens per short clip as LLaVA-OV to ensure fair comparison.

B Layer Selection for Attention-Based Visual Token Selection

In our method, visual tokens are selected based on their attention scores to the generated captions. Rather than using attention from all layers, we compute scores from a small subset. Modern fast attention implementations (e.g., FlashAttention-2 [8]) avoid explicitly materializing the full N^2 attention matrix (with N as the sequence length) to reduce memory and computational cost. To maintain efficiency and avoid additional overhead, we use attention from only a small ℓ out of L total number of layers (28 in LLaVA-OV). In the main paper experiments, we set $\ell = 4$ and we select these layers via uniform sampling across the depth of the network. As shown in Table 8, this simple strategy provides robust performance comparable to using all layers, with minimal variance. In contrast, selecting layers from localized regions (e.g., early, middle, or late) yields less stable results, and using a single layer results in significantly worse performance.

C Methods for Aggregating Attention Scores in Token Selection

In our method, we opted for average score over different attention heads for token selection; however, we noted that alternative score aggregation strategies could also be explored. Here we present results on the NextQA-valset using *max* as the aggregation method and selecting 6% of the visual tokens (to directly compare with the results of the main paper). As shown in Table 9, both methods perform similarly, with a negligible difference of only 0.1% in accuracy.

D Context Length for Retrieving & Answering

In the main paper, we use a context length of 10k tokens for retrieving and answering user queries. This choice is motivated by two key factors: (1) **Latency:** In streaming video applications, where user

⁴These layers are used across experiments in the main paper.

Table 8: Accuracy on NextQA validation set for different layer selections, keeping 6% of visual tokens.

Layers	Acc. on NextQA-valset (%)
Full Model	78.6
<i>all layers</i>	76.8
<i>[5, 9, 14, 20]⁴</i>	77
<i>[3, 12, 18, 24]</i>	76.8
<i>[1, 10, 15, 25]</i>	77.2
<i>[4, 5, 16, 28]</i>	76.7
<i>[3, 4, 5, 6]</i>	75.2
<i>[13, 14, 15, 16]</i>	76.6
<i>[24, 25, 26, 27]</i>	76.6
<i>[3]</i>	73.3
<i>[15]</i>	76.4
<i>[26]</i>	75.7

Table 9: Accuracy on NextQA validation set for different layer aggregation methods of attention heads, keeping 6% of visual tokens.

Aggregation Method	Acc. on NextQA-valset (%)
Full Model	78.6
Avg.	77.0
Max	77.1

queries arrive alongside frames in real-time, low-latency responses are critical. Increasing the context length substantially slows down both retrieval and LLM inference. (2) **Noise Filtering:** Long video streams are highly redundant, and larger contexts increase the risk of including irrelevant information, which can degrade performance. We empirically validate these considerations in Table 10 on RVS-Movie and RVS-Ego streaming benchmarks, showing that while increasing context from 6k to 10k improves accuracy, a further increase to 20k offers no additional benefit—and in some cases, even hurts performance—while incurring higher computational costs. Overall, a 10k context length offers a favorable trade-off between effectiveness and efficiency. Finally, we note that ReKV [9] retrieves 64 frames (each consisting of 196 tokens, since the same model is used), resulting in a larger context and, thus, in higher latency (2.7s) as reported in the main paper.

Table 10: Comparison of accuracy and score under different context lengths on RVS-Movie and RVS-Ego.

Context Length	RVS-Ego		RVS-Movie		Latency
	Acc.	Score	Acc.	Score	
20k	65.3	4.0	56.0	3.5	3.9s
10k	65.3	4.0	57.7	3.6	1.9s
6k	62.7	3.9	57.0	3.5	1.6s

E Positional Information

A potential concern with our proposed method is the manipulation or loss of spatial positional information during visual token selection. In rLiVS, we retain only the temporal order of selected tokens, potentially altering their original spatial layout when reinserting them into the LLM context. However, our initial experiments suggest that this design choice does not hinder performance. As shown in Table 7, selecting only 19% of the visual tokens—based on our relevance scoring and preserving temporal order—achieves comparable performance to using the full set of visual tokens on the NextQA validation set. Interestingly, selecting 25% of tokens even outperforms the full-token baseline, indicating that reducing redundancy and filtering out less informative content is more beneficial than strictly preserving spatial structure. This finding aligns with prior observations (e.g.,

[13]) that video data contains substantial redundancy, which can degrade performance if not mitigated. Additionally, since vision-language models (VLMs) are primarily pre-trained for caption generation, we posit that leveraging this objective as an intermediate step acts as an effective noise filter. During pretraining, these models learn to focus on the most informative visual tokens—those that align with the semantics of captions. Our method builds directly on this property by selecting tokens that contribute most to caption generation, thereby enhancing relevance and reducing noise. Finally, we note that by choosing not to address the challenge of accurately injecting positional information, our method remains orthogonal to any specific video-LLM architecture. Developing improved strategies to explicitly retain or reintroduce spatial information—and thereby better model spatio-temporal dynamics during processing—remains an open research direction, which we leave for future work.

F Visual tokens vs Captions based Retrieval

In our ablation studies, we argue that visual tokens projected ad hoc from the visual encoder lack learned semantic alignment with user questions. As a result, they produce uninformative similarity scores and lead to confusing retrieval behavior. In contrast, captions—being in the same modality as user queries—naturally lie within the same LLM input space, making them more suitable for retrieval. Figure 2 illustrates the distribution of cosine similarity scores between visual tokens and queries, as well as between captions and queries, aggregated across videos and questions in RVS-Movie. We observe that the similarity distribution between visual tokens and queries is narrowly centered around zero, with values ranging from approximately -0.02 to 0.06. In contrast, the similarity scores between captions and queries span a much broader range—from about 0.4 to 0.9—nearly an order of magnitude wider—potentially indicating a greater capacity to distinguish relevant from irrelevant content during retrieval.

To verify this point, we present performance results on both RVS-Ego and RVS-Movie using three similarity-based retrieval methods: visual tokens, captions, and random selection⁵. As shown in Table 11, retrieval using visual tokens performs on par with random selection, reinforcing the claim that visual tokens provide limited value. In contrast, caption-based retrieval consistently yields improvements of approximately 2–3%, underscoring its effectiveness.

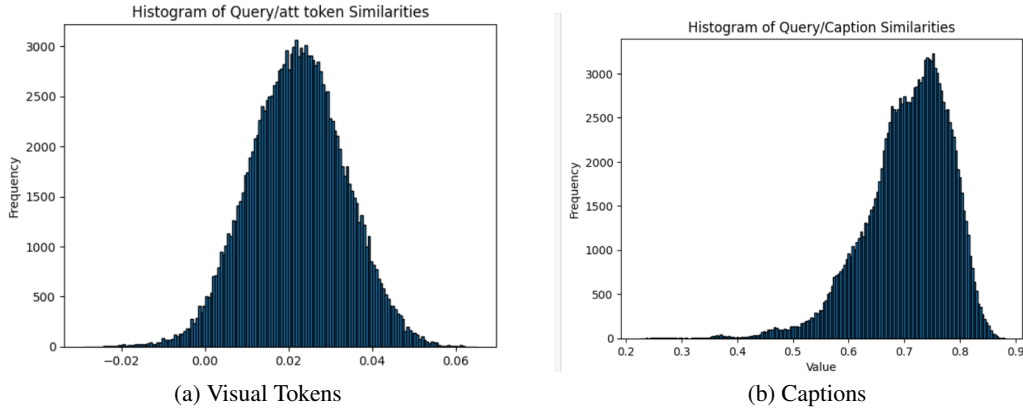


Figure 2: Distribution of cosine similarities between visual tokens and queries (left), and between captions and queries (right), averaged over videos and questions in RVS-Movie.

G Our uniform baseline vs ReKV’s [9]

In the main paper, we compared attention-based and uniform visual token selection on the streaming benchmarks RVS-Ego and RVS-Movie, showing that our attention-based method consistently outperforms uniform sampling by approximately 1–2%. Notably, the uniform baseline in that comparison was strengthened with two key components of our approach: recurrency and short-clip level token selection—both of which significantly improve performance. Here, we present a simpler uniform

⁵Across methods, captions are retrieved for answering.

Table 11: Comparison of accuracy and score under different retrieval strategies (based on captions, visual tokens or random) on RVS-Movie and RVS-Ego.

Retrieval	RVS-Ego		RVS-Movie	
	Acc.	Score	Acc.	Score
Random	64.6	3.9	55.2	3.5
Visual Tokens	63.7	3.9	55.0	3.5
Captions	65.3	4.0	57.7	3.6

sampling baseline more aligned with the setting of [9], where tokens are sampled uniformly across the entire video. To adapt this to our framework, we uniformly select 32 frames globally to fill the context, consistent with the frame length that our video-LLM backbone was trained. As shown in Table 12, this naive baseline performs significantly worse—about 10% lower—than our strengthened uniform variant, highlighting the importance of our proposed design choices.

Table 12: Comparison of accuracy and score of global uniform sampling (sampling frames) vs our recurrent short-clip caption-based baseline on RVS-Movie and RVS-Ego.

Baseline	RVS-Ego		RVS-Movie	
	Acc.	Score	Acc.	Score
Global Uniform	56.2	3.7	44.1	3.1
Recurrent short-clip caption-based Uniform	64.2	3.9	56.0	3.5

H Attention-based Selection based on Input Instruction

H.1 Core Mechanism

We propose a flexible top-down attention mechanism in which input instructions guide the selection of visual tokens through a caption- or response-generation step. The overall pipeline is structured as follows:

$$\text{Instruction} \rightarrow \text{Generated Caption/Response} \rightarrow \text{Attention-Based Token Selection}$$

Tokens are selected based on their attention scores during text generation, aligning with task-specific semantic relevance. As discussed, this mechanism draws inspiration from human memory systems, wherein attention is modulated by task-driven goals rather than uniform processing across all inputs.

While prior work has leveraged attention scores for token reduction [13, 26, 37], these approaches typically operate within individual layers, pruning tokens based on localized attention during early processing stages. In contrast, our method introduces several key innovations. First, we compute *cross-layer attention aggregation*, identifying tokens that receive consistently high attention across multiple layers. Second, we apply *sparse token retention*, keeping only a small subset of visual tokens (approximately 6%) based on the aggregated scores. Third, through *recurrent token reuse*, the selected tokens are propagated across successive short video segments, enabling temporal continuity without any model fine-tuning. This design not only reduces computational cost but also preserves semantic coherence across clips, facilitating efficient and context-aware video reasoning.

H.2 Task Adaptability

Although our main experiments focus on caption-guided token selection for visual question answering (VQA), the proposed framework is inherently instruction-adaptive and capable of generalizing across a variety of video understanding tasks through modification of the input instruction. For example:

- **Person Tracking:** “Track the movement of the person in the red shirt throughout this video sequence.”

- **Action Recognition:** “Identify and describe the specific athletic movements performed.”
- **Object Detection:** “Locate and describe all vehicles appearing in this scene.”

Each task-specific instruction elicits distinct model outputs, leading to unique attention patterns and token selections that reflect the semantic priorities of the task. To qualitatively assess this behavior, we conducted a comparative analysis of token selection overlaps under varying instructions.

Case Study: Instruction Sensitivity We selected two representative videos and evaluated the overlap of selected visual tokens using a generic captioning prompt versus task-specific instructions:

- **Video 1:** A girl with sunglasses is the main foreground object.
 - *Instruction:* “Track the locations of the girl wearing the sunglasses in the video.”
 - *Token Overlap with Generic Prompt:* 44%
- **Video 2:** Two people playing cards, with interest focused on the background.
 - *Instruction:* “Locate and describe the objects appearing in the background of the video.”
 - *Token Overlap with Generic Prompt:* 8%

The generic prompt used in both cases was: “Describe what’s happening in the video,” consistent with the one used during main captioning experiments. The results demonstrate that token selection is highly sensitive to the given instruction, validating the dynamic adaptability of our attention-based mechanism.

I Long Video Summarization Task

In this section, we conduct a preliminary evaluation of our method’s task generalizability through long video summarization. Using the MLVU validation split [39], we compute the holistic video summarization score and compare our approach to Video-XL [27], a recent training-based streaming video understanding model that also employs recurrent mechanisms. As shown in Table 13, rLiVS outperforms Video-XL in holistic summarization quality.

Table 13: Holistic video summarization scores on MLVU (validation).

Method	Model Size	MLVU Summarization Score
Video-XL	7B	3.40
rLiVS (Ours)	7B	3.65

These results demonstrate that rLiVS not only matches but outperforms specialized, trained architectures like Video-XL while remaining training-free and easily extensible. We include a more detailed breakdown of this comparison in the revised version of the paper.

J Qualitative Analysis

To assess the effectiveness of our aggressive token compression in complex and dynamic scenarios, we conducted a qualitative evaluation on a subset of videos from the SportsQA dataset [18]. Although full quantitative results are pending due to dataset access constraints, we compare caption outputs generated using our method (6% token retention) against those generated using all visual tokens.

Despite the significant reduction in token count, the model retains its ability to produce semantically rich, fine-grained descriptions, closely matching the full-token baseline. Below are selected examples from the *aerobic gymnastics* category:

- **Full:** “The video showcases a synchronized gymnastics routine performed by a group of athletes at the 10th European Championships for Aerobic Gymnastics.”
- **6%:** “The video is about a gymnastics routine performed by the Italian team at the 10th European Championships.”

- **Full:** "The video showcases a rhythmic gymnastics performance by three athletes."
6%: "The video showcases a rhythmic gymnastics routine performed by three athletes."
- **Full:** "The video showcases a synchronized gymnastics routine performed by two athletes at the 5th Aerobic Gymnastics Asian Championships."
6%: "The video is about two gymnasts performing a synchronized routine at the 5th Aerobic Gymnastics Asian Championships."
- **Full:** "The video is about a rhythmic gymnastics performance by two athletes at the 7th FIG Aerobics World Age Group Competitions in Incheon, Korea."
6%: "The video showcases a rhythmic gymnastics routine performed by two athletes."

These examples indicate that our model maintains high semantic fidelity, even under aggressive compression, and generalizes well to fast-changing scenes with multiple actors and fine-grained actions. This suggests the viability of our fixed but efficient token selection approach in high-complexity domains such as sports.

K Prompt Templates

We show here the prompts we use for caption generation and for the LLM evaluation:

Caption Generation. "Describe what's happening in the video."

LLM evaluation. For this, we follow previous works [38, 9] and utilize the following commonly used template:

```
messages=[
{
"role": "system",
"content":
"You are an intelligent chatbot designed for evaluating the correctness of
generative outputs for question-answer pairs. "
"Your task is to compare the predicted answer with the correct answer and
determine if they match meaningfully. Here's how you can accomplish the
task:"
"___"
"##INSTRUCTIONS: "
"- Focus on the meaningful match between the predicted answer and the
correct answer.\n"
"- Consider synonyms or paraphrases as valid matches.\n"
"- Evaluate the correctness of the prediction compared to the answer."
},
{
"role": "user",
"content":
"Please evaluate the following video-based question-answer pair:\n\n"
f"Question: {question}\n"
f"Correct Answer: {answer}\n"
f"Predicted Answer: {pred}\n\n"
"Provide your evaluation only as a yes/no and score where the score is an
integer value between 0 and 5, with 5 indicating the highest meaningful
match. "
"Please generate the response in the form of a Python dictionary string
with keys 'pred' and 'score', where value of 'pred' is a string of 'yes' or
'no' and value of 'score' is in INTEGER, not STRING."
"DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION. Only provide the
Python dictionary string. "
"For example, your response should look like this: {'pred': 'yes',
'score': 4.8}."
}
]
```

L Broader Impact

The broader impact of our approach can be discussed from three main aspects: the environmental impact, accessibility, and ethical consideration. Our method significantly reduces computational costs and energy consumption, subsequently lowering the carbon footprint associated with long video understanding. Along with the efficiency, our approach makes streaming video question answering readily accessible to practitioners and researchers with limited resources. However, as in the case of most AI research, our approach can potentially be misused, and it is crucial to ensure a responsible application. As such, preventing its use in unauthorized surveillance or privacy violations is important. In general, developing comprehensive guidelines and regulations on the ethical use of large video language models is essential to mitigate these risks.