Static or Dynamic: Towards Query-Adaptive Token Selection for Video Question Answering

Anonymous ACL submission

Abstract

Video question answering benefits from the rich information available in videos, enabling a wide range of applications. However, the large volume of tokens generated from longer videos presents significant challenges to memory efficiency and model performance. To alleviate this issue, existing works propose to compress video inputs, but usually overlooking the varying importance of static and dynamic information across different queries, leading to inefficient token usage within limited budgets. To tackle this, we propose a novel token selection strategy, EXPLORE-THEN-SELECT, that adaptively adjust static and dynamic information needed based on question requirements. Our framework first explores different token allocations between key frames, which preserve spatial details, and delta frames, which capture temporal changes. Next, it employs a query-aware attention-based metric to select the optimal token combination without model updates. Our proposed framework is plug-and-play that can be seamlessly integrated within diverse video-language models. Extensive experiments show that our method achieves significant performance improvements (up to 5.8%) among various video question answering benchmarks. The code is accessible at the anonymous link.

1 Introduction

Video Question Answering (VideoQA) has broad applications across various fields (Mogrovejo and Solorio, 2024; Zhang et al., 2024a). Compared to text, videos provide more intuitive and dynamic information, delivering richer context and details by combining visual and temporal elements. Current research primarily leverages powerful large language models to build video-language models (VideoLMs) (Lin et al., 2023; Zhang et al., 2024b), significantly enhancing AI performance in VideoQA tasks. However, the extensive visual information in long videos leads to a dramatic increase in token counts. For instance, if one frame generates 196 tokens (Li et al., 2024a), a 5-minute video sampled at 1fps would produce nearly 60,000 tokens, posing significant challenges to memory requirements and model capabilities.

Figure 1: Different question types vary in their dependence on static and dynamic information in videos. For example, Question 2 relies on fine-grained dynamic information, while Question 1 and 3 only require key frames. The frames needed to answer the questions are highlighted with corresponding colored boxes.

Given the strict token limitations in practical VideoLM deployments, effectively representing essential video information requires a careful allocation between static and dynamic content. Static information, which refers to the visual content within individual frames, is crucial for questions like object recognition, where spatial details dominate. In contrast, dynamic information captures temporal changes and motion patterns across consecutive frames, which are essential for understanding actions or events. Figure 1 illustrates different types of questions, which vary in their reliance on static and dynamic information. Considering these varying dependencies, the challenge lies in optimizing the allocation of limited tokens to preserve the

most relevant aspects of both static and dynamic information, depending on specific question requirements. Although existing studies (Shen et al., 2024; Nie et al., 2024) have explored token compression through changing frame sampling rates or intra-frame downsampling, they fail to address the varying dependencies on static and dynamic information across different question types.

To achieve an effective allocation between static and dynamic information in video token compression, we propose a novel token selection strategy, EXPLORE-THEN-SELECT, that adaptively aligns video tokens with textual queries under a limited token budget. Unlike previous approaches that rely on fixed rules, our strategy autonomously and adaptively combines static and dynamic content based on the nature of the questions (e.g., action description, event sequence, or object recognition), ensuring more precise responses to diverse queries.

Specifically, we categorize video frames into key and delta frames. key frames are fully retained to preserve essential spatial details, such as objects, while delta frames are sparsely processed, keeping only a subset of tokens to capture important temporal changes. To optimize token allocation between these two types of frames, EXPLORE-THEN-SELECT uses a two-stage process. In the **exploration** stage, we construct a search space comprising various combinations of key and delta frames, each yielding a token subsequence of equal constrained length. By adjusting the proportion of key and delta frames, we can prioritize either static details or dynamic changes based on question requirements. In the **selection** stage, we evaluate each combination using an query-aware metric derived from the shallow attention layers of VideoLMs. This metric quantifies the alignment between the query and video tokens, enabling us to select the optimal combination to answer the question.

Notably, our framework is training-free, as neither the exploration nor selection processes require model updates. Leveraging its seamless integration with diverse VideoLMs, we demonstrate the effectiveness of our approach on two widely recognized VideoLMs across multiple benchmarks for both long and short videos. Using our framework, models can achieve improvements of up to 5.8%. Our key contributions are summarized as follows:

- Building on the observation that questions rely differently on static and dynamic video information, we propose a novel EXPLORE-THEN-SELECT framework to adaptively and effectively select video tokens reflecting the optimal balance of static and dynamic information under limited token budgets.

- To address static and dynamic information needs, we design an effective search space of key-delta frame combinations. During the selection phase, we employ a query-aware approach, leveraging an attention-based metric to adaptively evaluate candidates and select the optimal combination for each question.

- We conduct extensive experiments on both long- and short-video benchmarks, demonstrating the effectiveness of our method. Thanks to its plug-and-play design, our approach generalizes well across different models without extra fine-tuning and enables direct control over the token budget for flexible adaptation to resource constraints.

2 Related Work

2.1 Video Language Models

Significant progress has been made in video language model research based on LLMs. These models can be primarily classified into two types: general-purpose vision-language models (Team et al., 2024; Chen et al., 2024b; OpenAI, 2024; Yao et al., 2024; Ye et al., 2023) and specialized video language models (Lin et al., 2023; Zhang et al., 2024c; Li et al., 2024c; Zhang et al., 2025; Liu et al., 2024). Among the former, LLaVA-OneVision (Li et al., 2024a) unifies image and video tasks, while Qwen2-VL (Wang et al., 2024) introduces dynamic resolution support and three-dimensional positional encoding for enhanced visual feature capture. Among specialized models, VideoChat (Li et al., 2023b) targets deep video understanding and interaction, and LongVA (Zhang et al., 2024b) extendeds the context length of language models, transferring their advantages in long-text processing to the video domain.

2.2 Visual Token Compression

Some studies (Bolya et al., 2022) focus on compressing visual tokens in video encoders. For example, RLT (Choudhury et al., 2024) effectively reduces the number of tokens by replacing repeated patches in videos with a single patch. Other works (Li et al., 2024b; Qian et al., 2025; Shen et al.,

Method	Pre-Input	Training-Free	Video-Specific
FastV (Chen et al., 2024a)	✗	✓	✗
ZipVL (He et al., 2024b)	✗	✓	✗
FrameFusion (Fu et al., 2024b)	✗	✓	✓
TokenPacker (Li et al., 2024b)	✓	✗	✗
VideoStreaming (Qian et al., 2025)	✓	✗	✓
SlowFocus (Nie et al., 2024)	✓	✗	✓
LongVU (Shen et al., 2024)	✓	✗	✓
Ours	✓	✓	✓

Table 1: Feature comparison with existing methods. "Pre-Input" refers to methods which reduce tokens before feeding them into large language models, while "Video-Specific" denotes methods which leverage the unique characteristics of video data.

2024; Lan et al., 2024) introduce dedicated modules for token compression, such as BLIP-2 (Li et al., 2023a), which uses a Q-Former module with learnable queries to generate compact semantic representations. Additionally, inspired by KV cache compression in long-text processing (Zhang et al., 2023), some methods apply similar strategies to visual tokens (He et al., 2024b; Chen et al., 2024a; Fu et al., 2024b). These methods optimize token usage efficiency by setting thresholds based on specific metrics to prune visual tokens.

We compare existing methods in Table 1, noting that training-free approaches mainly compress tokens within the KV cache, reducing FLOPs but not addressing the issue of excessive token input to the large language model. In contrast, methods that reduce tokens in advance typically require training. This paper introduces a novel pre-input, training-free framework for more effective compression, considering query-aware static and dynamic information balancing.

3 Preliminary

In this section, we outline the common inference pipeline of VideoLMs as the setup for our approach. It consists of three key steps, including video frame sampling, visual encoding and embedding, and multimodal inference.

Video Frame sampling. Given an input video, N frames are uniformly sampled to form a representation $\boldsymbol{V} \in \mathbb{R}^{N \times C \times H_v \times W_v}$, where $C = 3$ denotes the RGB channels, and H_v and W_v represent the height and width of each frame, respectively.

Visual Encoding and Embedding. The sampled frames are decomposed into non-overlapping spatiotemporal patches, which are processed by a vision encoder to extract spatiotemporal features. These features are projected into the language model's token space via a linear projector, resulting in visual token embeddings $\boldsymbol{F} \in \mathbb{R}^{T \times H \times W \times D}$, where T represents the temporal resolution, H and W denote the spatial resolutions, and D is the token embedding dimension.

Multimodal Processing. The visual token embeddings $\boldsymbol{F}$ are then flattened into a sequence $\boldsymbol{T}_v \in \mathbb{R}^{THW \times D}$, where the sequence length is $L = T \times H \times W$. The sequence $\boldsymbol{T}_v$, instruction embeddings $\boldsymbol{T}_i$, and query embeddings $\boldsymbol{T}_q$ are concatenated into a unified input $\boldsymbol{T} = [\boldsymbol{T}_i, \boldsymbol{T}_v, \boldsymbol{T}_q]$, where $[\cdot]$ denotes token concatenation. Finally, VideoLM processes the unified input sequence $\boldsymbol{T}$ to generate a textual response to the question.

4 Method

4.1 Problem Definition

Due to GPU memory and model capability constraints, the number of visual tokens processed during inference is capped at L_b. The fixed token budget limits frame sampling to a reduced number of frames, resulting in significant loss of rich visual information, particularly in long videos.

In this work, we aim to sample more frames to expand the amount of information we can capture, which generates an excessive number of tokens, leading to a sequence length $L \gg L_b$. Then we compress the tokens to meet the token budget, enabling more effective utilization of rich visual information within the limited length.

To meet our goal, we propose a token-efficient framework that automatically and adaptively selects a limited yet informative set of visual tokens by leveraging the textual query's relevance to both static and dynamic visual information. Our method emphasizes balancing these two types of information, ensuring that the selected tokens maximize their alignment with the query while maintaining memory efficiency.

4.2 Framework Overview

We adopt an EXPLORE-THEN-SELECT framework, as illustrated in Figure 2. In the token exploration stage (Section 4.3), we construct a search space of n visual token subsequences, each of length L_b, where every visual token subsequence reflects a distinct balance of static and dynamic information. In

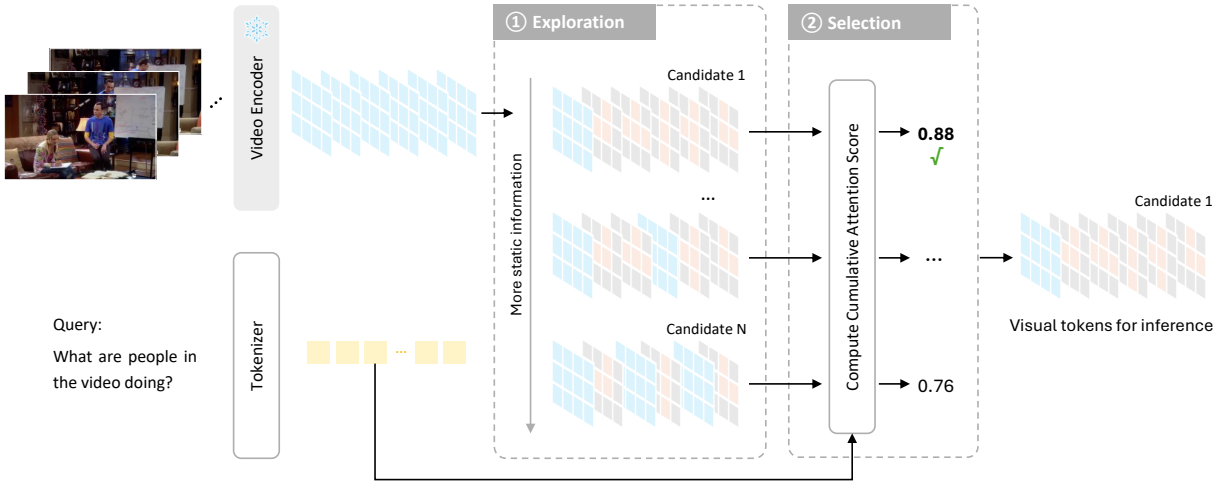

Figure 2: Overview of our EXPLORE-THEN-SELECT framework for token selection. In the exploration stage, we generate different subsequences of different combinations of key and delta-frame tokens. In the selection stage, we evaluate their query-aware metrics based on shallow attention layers and select the optimal one for input to LLMs.

the token selection stage (Section 4.4), we identify the optimal sequence that best aligns with the query requirements. Details will be discussed below.

4.3 Exploration: Search Space Design

This section describes the generation of n token subsequences, each of length L_b, from a token sequence of length L. To balance static and dynamic information in videos based on query requirements, we classify frames into key and delta frames. It is worth noting that since some models employ temporal reduction, here we refer to "frames" as $\boldsymbol{F}$ and the number of frames is T. Based on whether the tokens in a subsequence originate from key or delta frames, we divide them into two subsets: key-frame tokens $\mathcal{T}_{\text{key}}$ and delta-frame tokens $\mathcal{T}_{\text{delta}}$.

Key-frame Token. The key-frame tokens are extracted from the key frames. Assuming N_s key frames are chosen in the video, we uniformly select them from $\boldsymbol{F}$. The temporal indices of these frames are:

$$\mathcal{I} = \left\{ \left\lfloor \frac{kT}{N_s} \right\rfloor + 1 \mid k = 0, 1, \ldots, N_s - 1 \right\}, \quad (1)$$

where the first frame is always selected as a key frame. We preserve all tokens from these key frames to constitute $\mathcal{T}_{\text{key}}$ as:

$$\{\boldsymbol{F}^{i,h,w} \mid i \in \mathcal{I}, h \in [1, H], w \in [1, W]\}, \quad (2)$$

where $\boldsymbol{F}^{i,h,w} \in \mathbb{R}^D$ represents the token embedding at the i-th frame and spatial location (h, w) in $\boldsymbol{F}$. Obviously The total number of key tokens satisfies $|\mathcal{T}_{\text{key}}| = N_s \times H \times W$.

Delta-frame Token. As illustrated in Figure 3, the key frames divide the entire sampled frame sequence into N_s intervals. The frames within these intervals are defined as delta frames, and the delta-frame tokens $\mathcal{T}_{\text{delta}}$ are extracted from them to capture the dynamic information relative to the preceding key frames. Since the subsequence length is L_b, it is evident that the number of delta-frame tokens satisfies $|\mathcal{T}_{\text{delta}}| = L_b - |\mathcal{T}_{\text{key}}|$. These tokens are uniformly distributed across each interval, meaning the number of delta-frame tokens selected from the i-th interval satisfy $|\mathcal{T}_{\text{delta},i}| = \lfloor |\mathcal{T}_{\text{delta}}| / N_s \rfloor$.

Inspired by video codec, to retain as much dynamic information as possible, we select tokens from each interval that exhibit the largest differences compared to the corresponding tokens in the preceding key frame. We first define the token difference metric based on the cosine similarity between two token embeddings:

$$\mathcal{D}(\boldsymbol{f}_i, \boldsymbol{f}_j) = 1 - \frac{\boldsymbol{f}_i \cdot \boldsymbol{f}_j}{\|\boldsymbol{f}_i\| \|\boldsymbol{f}_j\|}. \quad (3)$$

This metric increases as the two embeddings become more dissimilar. Then, we select $|\mathcal{T}_{\text{delta},i}|$ tokens in the interval i that have the largest differences compared to the corresponding tokens in the preceding key frame. We define $\mathcal{T}_{\text{delta},i}$ as:

$$\{\boldsymbol{F}_i^{j,h,w} \mid \text{Top}_{|\mathcal{T}_{\text{delta},i}|} \mathcal{D}(\boldsymbol{F}_i^{0,h,w}, \boldsymbol{F}_i^{j,h,w}), \\ j \in [1, T_i], h \in [1, H], w \in [1, W]\}, \quad (4)$$

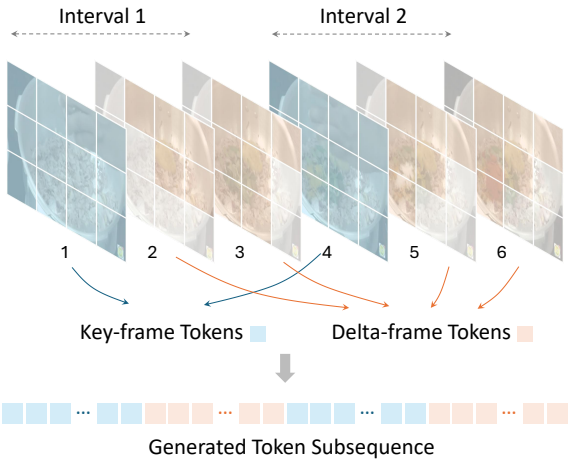

Figure 3: An example of token subsequence generation with 6 total frames and 2 key frames.

where $\boldsymbol{F}_i^{0,h,w}$ denotes the token embedding at position (h, w) in the preceding key frame of current interval i, and $\boldsymbol{F}_i^{j,h,w}$ denotes the token embedding at position (h, w) in the j-th delta frame of interval i. T_i refers to the number of delta frames in i-th interval.

Token Subsequence Generation. Then we merge $\mathcal{T}_{\text{key}}$ and $\mathcal{T}_{\text{delta}}$ according to their original order to obtain the L_b-long token subsequence $\hat{\boldsymbol{T}}_v$.

To generate n candidate token subsequences, we vary N_s from 1 to n. As N_s decreases, the number of delta-frame tokens increases, thereby preserving more dynamic information under the same token budget. Conversely, the number of key-frame tokens increases, preserving more static information. In this way, we can generate token subsequences with varying proportions of static and dynamic information to adapt to the requirements of different queries.

Notably, our frame division is inspired by the GOP structure in video codec (Lee et al., 2006), where I-frames capture full scenes and P/B-frames encode temporal changes. Besides, similar to adjusting GOP sizes, varying the proportion of key and delta frames allows us to control the emphasis on static or dynamic cues.

4.4 Selection: Quick Evaluation

After obtaining n token subsequences of length L_b, we perform an evaluation and select the optimal subsequence based on the chosen metric. Previous studies have identified certain characteristics of visual tokens in attention mechanisms. For instance, Chen et al. (2024a) shows that most vision tokens can be removed at the second layer without significant performance loss, and Wan et al. (2024) observes that vision tokens are generally less attended. Based on these findings, we consider that the attention mechanism at the second layer already provides meaningful clues of token importance. Besides, we hypothesize that higher cumulative attention scores on visual tokens indicate a better utilization of the visual information.

To enable quick evaluation, we compute the attention score matrix $\boldsymbol{S}$ at the second layer of the VideoLMs, using textual query tokens as the query input, and instruction and vision tokens as the key input:

$$\boldsymbol{Q} = \boldsymbol{W}_{\text{Q}} \boldsymbol{H}_q, \tag{5}$$

$$\boldsymbol{K} = \boldsymbol{W}_{\text{K}} \cdot \text{Concat}(\boldsymbol{H}_i, \boldsymbol{H}_v), \tag{6}$$

$$\boldsymbol{S} = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right), \tag{7}$$

where $\boldsymbol{H}_q$, $\boldsymbol{H}_i$, and $\boldsymbol{H}_v$ denote the hidden features of the textual query, instruction, and visual inputs. And d_k is the dimension of key vectors in the attention mechanism. To quantify the attention allocated to visual tokens, we compute the summation of attention scores of the visual tokens. Specifically, to ensure comprehensive consideration of each text query token, we first extract the maximum values along the query dimension from $\boldsymbol{S}$, yielding an attention score vector $\boldsymbol{s}$ for each visual token. Then we sum the attention scores of the visual tokens:

$$\boldsymbol{s} = \max_i \boldsymbol{S}_{ij}, \tag{8}$$

$$s = \sum_{j=N_i}^{N_i+N_v} \boldsymbol{s}_j, \tag{9}$$

where $\boldsymbol{S}_{ij}$ represents the attention score of the i-th query token to the j-th visual token, N_i denotes the number of instruction tokens, and N_v is the number of visual tokens. Finally, from the n candidates, we select the input with the highest sum of visual token attention scores as the optimal input:

$$\bar{\boldsymbol{T}}_v = \underset{m \in \{1,2,\ldots,n\}}{\arg\max}\ s^{(m)}, \tag{10}$$

where $s^{(m)}$ denotes the summed attention score for the m-th token subsequence.

Model	Settings			EgoSchema	VideoMME				MLVU
	Method	Sample	Budget		Short	Medium	Long	Overall	
VideoChat2	-	16	-	54.4	48.3	37.0	33.2	39.5	-
LongVA	-	128	-	-	61.1	50.4	46.2	52.6	-
mPLUG-Owl3	-	128	-	-	70.0	57.7	50.1	59.3	-
LongVU	-	1fps	-	67.6	-	-	-	60.6	65.4
Qwen2-VL-7B	Original	64	-	66.2	71.1	59.4	50.8	60.4	50.6
	Retrieval	256	64	63.6	71.0	61.3	52.2	61.5	49.4
	Similarity	256	64	66.6	71.4	60.6	51.8	61.3	53.0
	Ours	256	64	**67.8**	**72.4**	**63.1**	**53.2**	**62.9**	**54.4**
	Original	32	-	64.7	68.9	55.2	48.7	57.6	46.8
	Retrieval	128	32	61.7	70.0	58.6	51.6	60.0	46.8
	Similarity	128	32	65.6	70.1	58.7	**51.8**	60.2	47.2
	Ours	128	32	**66.7**	**71.4**	**61.0**	51.7	**61.4**	**52.2**
LLaVA-OneVision-7B	Original	64	-	60.1	70.6	55.8	47.8	58.0	50.8
	Retrieval	256	64	57.7	64.0	53.4	47.0	54.8	44.6
	Similarity	256	64	59.6	71.0	57.9	50.8	59.9	48.4
	Ours	256	64	**60.3**	**71.9**	**58.3**	**51.4**	**60.6**	**51.2**
	Original	32	-	60.4	**71.3**	57.4	48.0	58.9	46.8
	Retrieval	128	32	57.9	63.2	53.9	46.0	54.4	44.0
	Similarity	128	32	60.2	70.8	57.1	49.7	59.2	50.2
	Ours	128	32	**60.5**	70.2	**58.0**	**51.6**	**59.9**	**51.0**

Table 2: Results on long video benchmarks show that our method achieves significant improvements over the baselines, particularly on the advanced Qwen2-VL, with up to a 5.8% gain on the VideoMME medium subset.

5 Experiments

5.1 Experiment Settings

Benchmarks. To comprehensively evaluate performance, we select benchmarks for both long and short videos. We use VideoMME, EgoSchema, and MLVU for long videos, and MSVD-QA and ActivityNet-QA for short videos.

VideoMME (Fu et al., 2024a) contains 900 videos (11 seconds to 1 hour) and 2,700 QA pairs. EgoSchema (Mangalam et al., 2023) includes over 5,000 questions based on videos averaging 3 minutes in length. MLVU (Zhou et al., 2024) provides over 500 QA pairs on videos ranging from 3 minutes to 2 hours. MSVD-QA (Xu et al., 2017) includes 1,970 short clips (10 seconds on average), with a test split of approximately 13,000 questions. ActivityNet-QA (Yu et al., 2019) provides 800 videos and 8,000 QA pairs in the test set, averaging around 10 questions per video.

We adopt multiple-choice accuracy as the metric for VideoMME, EgoSchema, and MLVU, and employ GPT-based scoring (OpenAI, 2024) for the open-ended MSVD-QA and ActivityNet-QA.

Baselines. We validate our plug-and-play method on two representative models: Qwen2-VL (Wang et al., 2024), featuring dynamic resolution and multimodal rotary position embeddings, and LLaVA-OneVision (Li et al., 2024a), supporting multi tasks, both in their 7B versions. Results for Qwen2.5-VL (Bai et al., 2025) are included in Appendix A.1.

As shown in Table 1, prior methods either compress only within the KV cache, leaving long input sequences unaddressed, or require training models, making direct comparison with our training-free approach unfair. Thus we consider three baselines: 1) Original: uniform frame sampling within the token budget; 2) Retrieval: oversample frames, then prune based on cosine similarity between frame and query embeddings to fit the token limit; 3) Similarity: oversample frames, then prune based on cosine similarity between adjacent token embeddings. In practice, both "Retrieval" and "Similarity" strategies are commonly adopted in compression modules (Qian et al., 2025; Song et al., 2024; He et al., 2024a). For reference, we also report results from several training-based video understanding methods (Li et al., 2023b; Zhang et al., 2024b; Ye et al., 2024; Shen et al., 2024) in the first block of Table 2, though they are not directly comparable due to training cost differences. To further validate the advantages of our method, we include a comparison with our reproduced training-free LongVU in Appendix A.2.

Implementation Details. All experiments are conducted on two 40GB A100 GPUs. For multiple-choice questions, the model generates one token (three for MLVU), while for open-ended questions, outputs are limited to 30 tokens. The prompts used are detailed in Appendix B. Sampling is disabled to ensure deterministic results.

Note that video resolution affects the number of frame tokens generated by Qwen2-VL, making a fixed token budget yield varying frame counts across videos and complicating comparisons. To address this, we set a frame-based budget T_b, so the token limit is $L_b = T_b \times H \times W$, where $H \times W$ is the token count per frame. This approach streamlines implementation and ensures fair comparison.

5.2 Main Results

Long Video Results. Table 2 shows results on long video benchmarks for two settings: 256-frame sampling with a 64-frame budget (256-64) and 128-frame sampling with a 32-frame budget (128-32). Our method outperforms baselines across all benchmarks and most subsets. Qwen2-VL-7B significantly outperforms baselines by up to 4.2% on EgoSchema, 2.5% on VideoMME, and 5.0% on MLVU (256-64), and by up to 5.0%, 3.8%, and 5.4% (128-32), with a 5.8% gain on VideoMME medium subset. While our method also achieves notable improvements on LLaVA-OneVision-7B, the gains are less pronounced than on Qwen2-VL, likely due to noise from its one-dimensional positional encoding. The three-dimensional positional embedding of Qwen2-VL-7B offers more stable results, hilighting the importance of positional embedding design. Overall, these results demonstrate the effectiveness of our method, and reveal some model-specific behaviors and limitations.

Short Video Results. Short-video benchmarks inherently contain fewer frames, simpler scenes, and primarily coherent motion, making them less affected by token length limitations. As a result, the trade-off between static and dynamic information is less pronounced, and performance gains tend to be smaller compared to long-video settings. Nonetheless, we evaluate our method's generalization on short-video benchmarks by sampling 64 frames and setting the budget to 16 for videos averaging 10 seconds. As shown in Table 3, our method consistently outperforms all baselines on Qwen2-VL-7B, achieving up to 3.8% higher accuracy and 0.2 higher scores. On LLaVA-OneVision-7B, it achieves strong results on ActivityNet-QA and performs comparably to the "Retrieval" baseline on MSVD-QA. These results demonstrate the robustness and generalization ability of our method even under short-video scenarios.

Model	Method	MSVD-QA		ActivityNet-QA	
		Acc	Score	Acc	Score
Qwen2-VL	Original	66.0	3.59	50.3	2.82
	Retrieval	64.4	3.52	48.6	2.74
	Similarity	66.5	3.60	51.4	2.87
	Ours	**66.8**	**3.61**	**52.4**	**2.90**
LLaVA-OneVision	Original	54.3	3.09	52.6	2.90
	Retrieval	**54.8**	**3.12**	50.1	2.77
	Similarity	54.3	3.10	52.4	2.89
	Ours	54.7	3.11	**53.0**	**2.92**

Table 3: Results on short video benchmarks. Although primarily focused on long videos, our method show stable and generalizable performance on short videos.

Method	EgoSchema	VideoMME	MLVU
Orignal	64.7	57.6	46.8
Explore + Random	66.3	60.7	50.2
Explore + Select	**66.7**	**61.4**	**52.2**

Table 4: Ablation study of our method. Results demonstrate the effectiveness of both stages, with each component yielding improvements over the baseline.

5.3 Ablation Studies

Stage Ablation. As shown in Table 4, we conduct a two-stage ablation study on our method. The ablation experiments were performed on Qwen2-VL-7B, sampling 128 frames with a budget of 32 frames. First, we validated the effectiveness of the exploration stage. As indicated by the "Explore + Random" row in the table, generating multiple token subsequences followed by random selection results in improvement compared to the original operation, demonstrating the rationality of our search space design. Then we verify the effectiveness of the selection phase. On all benchmarks, our selection method achieves improvement over the random selection.

Metric Ablation. Table 5 presents two ablation studies on our metric design using Qwen2-VL-7B (128-frame sampling, 32-frame budget). The first block compares including or excluding the query token in the construction of $\boldsymbol{K}$ in Equation (6), finding only marginal differences; for simplicity, we exclude the query token in our final design. The second block compares max and mean operations

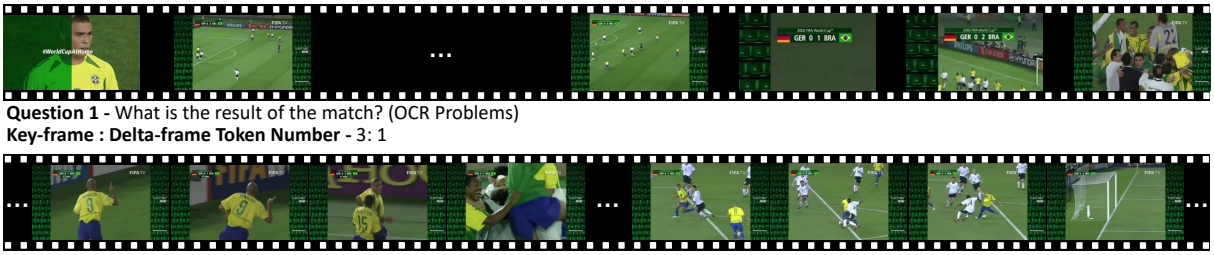

Figure 4: In our qualitative analysis cases, our method allocates key and delta-frame tokens at a ratio of 3:1 for Question 1 which is a OCR problem, and a ratio of 5:11 for Question 2 which pertains to action recognition.

Model	Method	EgoSchema	VideoMME
Qwen2-VL	w/ query	66.3	**61.6**
	w/o query	**66.7**	61.4
	mean	66.0	60.9
	max	**66.7**	**61.4**

Table 5: Ablation study on metric design. The first block shows that including the query token in $\boldsymbol{K}$ has negligible impact, so it is omitted. The second block finds that the max operation in Equation (8) outperforms the mean on both datasets.

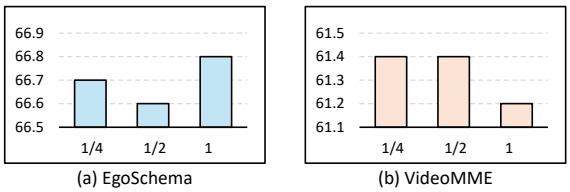

Figure 5: Search space size analysis. The x-axis represents the search space size. There are n subsequences in the space, and their key frame number ranges from $\{1, 2, \ldots, n\}$. Assuming the budget frame is N_b, "1" refers to $n = N_b$, "1/2" indicates $n = \lfloor N_b/2 \rfloor$, "1/4" represents $n = \lfloor N_b/4 \rfloor$. Larger search spaces benefit EgoSchema but hurt VideoMME and increase time cost. A balanced setting uses half the budget size.

for query aggregation in Equation (8), showing that the max operation consistently yields better results, thus supporting our metric choice.

5.4 Further Analysis

Qualitative Analysis. Figure 4 shows two questions for the same video, which are correctly answered employing our method. Question 1, an OCR problem predominantly reliant on static information, prompts the method to allocate a key-to-delta-frame token ratio of 3:1. Conversely, action-ralated Question 2, necessitating the identification of a player scoring a goal, leads to the adoption of a key-to-delta-frame token ratio of 5:11.

Search Space Size. We study search space size impact using Qwen2-VL-7B, sampling 128 frames with a 32-frame budget. Figure 5 shows performance improves on EgoSchema when search space matches the budget but declines on VideoMME. We attribute this to excessive key frames, causing sparse delta-frame token selection and deviation from the training distribution, reducing effectiveness. Additionaly, time cost rises with search space size. To balance these, we set the search space to half the budget frame number.

Time Overhead. The compression inevitably incurs time overhead, but our focus is on memory efficiency and information retention. Our overhead mainly comes from metric computation. Using shallow attention layers with question-length queries, sequential processing of 16 candidates in a 28-layer model doubles first-token latency but leaves subsequent decoding unaffected, benefiting open-ended questions. Parallel processing can further reduce latency. On LLaVa-OneVision-7B (128-frame sampling with a 32-frame budget), compared to the 2.24s overhead incurred by LongVU, our approach costs only 0.43s without flash-attention.

6 Conclusion

Given that long videos possess tokens far exceeding the capacity that models can process, we advance token compression strategies by unveiling the following crucial fact: different question types exhibit varying dependencies on dynamic and static information. Based on this discovery, we propose a novel token selection strategy for video token compression. Our method splits video frames into key and delta frames, and adaptively determines the optimal token allocations among key and delta frames guided by each specific query. Experiments demonstrate the effectiveness and generalizability of our method across multiple models and datasets.

Limitations

In this paper, we propose a novel token selection strategy for token compression in video question answering tasks, addressing the varying dependencies of questions on dynamic and static video information. While the effectiveness of our method has been validated across multiple datasets, certain limitations remain. Firstly, due to differences in positional encoding mechanisms across models, some encoding schemes may impact the model's ability to accurately judge video length and temporally localize events. Nevertheless, we believe our approach holds insight for developing compression modules in pre-trained and fine-tuned video models. Additionally, although our method incurs no additional memory overhead (superior to pruning in the key-value cache), it does introduce time overhead. This overhead mainly stems from metric computation. We utilize the output of a shallow (second-layer) attention mechanism to compute the metric, where only the attention map between query tokens and vision tokens is computed. This overhead only happens during the initial token inference and does not affect subsequent token generation. It is worth noting that such additional time cost is a common challenge for most compression methods.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2022. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024a. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer.

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

Rohan Choudhury, Guanglei Zhu, Sihan Liu, Koichiro Niinuma, Kris M Kitani, and László Jeni. 2024. Don't look twice: Faster video transformers with run-length tokenization. *arXiv preprint arXiv:2411.05222*.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. 2024a. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. 2024b. Framefusion: Combining similarity and importance for video token reduction on large visual language models. *arXiv preprint arXiv:2501.01986*.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. 2024a. Ma-lmm: 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.

Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. 2024b. Zipvl: Efficient large vision-language models with dynamic token sparsification and kv cache compression. *arXiv preprint arXiv:2410.08584*.

Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma. 2024. Vidcompress: Memory-enhanced temporal compression for video understanding in large language models. *arXiv preprint arXiv:2410.11417*.

Jeehong Lee, IlHong Shin, and HyunWook Park. 2006. Adaptive intra-frame assignment and bit-rate estimation for variable gop length in h. 264. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(10):1271–1279.

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

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2023b. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. 2024b. Tokenpacker: Efficient visual projector for multimodal llm. *arXiv preprint arXiv:2407.02392*.

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

Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. 2023. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*.

Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. 2024. Kangaroo: A powerful video-language model supporting long-context video input. *arXiv preprint arXiv:2408.15542*.

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

David Mogrovejo and Thamar Solorio. 2024. Question-instructed visual descriptions for zero-shot video answering. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9329–9339.

Ming Nie, Dan Ding, Chunwei Wang, Yuanfan Guo, Jianhua Han, Hang Xu, and Li Zhang. 2024. Slowfocus: Enhancing fine-grained temporal understanding in video llm. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

OpenAI. 2024. Gpt-4o system card. https://openai.com/index/gpt-4o-system-card/.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. 2025. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37:119336–119360.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. 2024. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. 2024. 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.

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

Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. 2024. Look-m: Look-once optimization in kv cache for efficient multimodal long-context inference. *arXiv preprint arXiv:2406.18139*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. 2017. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1645–1653.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2024. mplug-owl3: Towards long image-sequence understanding in multi-modal large language models. *arXiv preprint arXiv:2408.04840*.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.

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

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. 2025. Video-ollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*.

Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. 2024a. A simple LLM framework for long-range video question-answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21715–21737, Miami, Florida, USA. Association for Computational Linguistics.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. 2024b. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2024c. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.

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

A Additional Experiments

The appendix provides supplementary experiments, results on the advanced Qwen2.5-VL model and comparison with training-free LongVU.

A.1 Experiments on Qwen2.5-VL

Qwen2.5-VL is the latest vision-language model in the Qwen series, officially released in 2025 February. Building upon the foundation of Qwen2-VL, Qwen2.5-VL introduces significant enhancements in long-video comprehension. Notably, it incorporates absolute time encoding, enabling the model to handle videos of extended durations with second-level event localization. To provide a more comprehensive evaluation of our method, we report experimental results on the Qwen2.5-VL-7B model using the same experimental settings as in the main text.

Long Video Results. Table 6 presents the long video benchmark results on Qwen2.5-VL-7B under different sampling and budget settings. Across both 256-64 and 128-32 settings, our method consistently achieves the best performance on most benchmarks. Specifically, under the 256-64 setting, our approach outperforms all baselines on EgoSchema, VideoMME, and MLVU, achieving the highest accuracy of 61.6%, 65.7%, and 58.4%, respectively. Notably, on VideoMME and MLVU, our method yields improvements of up to 3.1% and 8.4% over the baselines. Similarly, in the 128-32 setting, our method continues to lead, with top results on EgoSchema (60.6%), VideoMME and MLVU (51.6%). These results demonstrate the effectiveness and robustness of our approach, and further validate its strong generalization capability across different models.

Short Video Results. Although our method is primarily focused on long video understanding, it also delivers strong results on short video tasks. For instance, on Qwen2.5-VL evaluated with ActivityNet-QA under the 64-frame sampling and 16-frame budget setting, our method achieves the best performance among all baselines. As shown in Table 7, it attains the highest accuracy of 54.3% and a score of 3.07, outperforming the baselines by up to 2.2% in accuracy and 0.11 in score.

A.2 Comparison with Training-free LongVU

To further demonstrate the advantages of our approach, we compare it with LongVU (Shen et al., 2024) by reproducing its compression method in a training-free setting. Following the original paper, we use DINOv2 (Oquab et al., 2023) with a 0.83 threshold for frame reduction and apply a $\lfloor 2/3 \rfloor$ downsampling ratio. However, we find that meeting a precise token budget with LongVU requires careful tuning of thresholds and heuristics, offering only indirect control over compression. In contrast, our method uses top-K selection, enabling direct and accurate control of the token count. As shown in Table 8, our method consistently outperforms the reproduced LongVU across all models and benchmarks, while providing more reliable and practical token budget management.

B Prompt Details

We utilize the template provided by the model for the instruction prompt part. We only introduce the textual organization format in the questioning part.

B.1 Prompts for Multiple-Choice Questions

We add the sentence "Respond with only the letter (A, B, C, or D) of the correct option." at the beginning of the multiple-choice questions. Here is an example for questions in VideoMME:

Model	Settings			EgoSchema	VideoMME				MLVU
	Method	Sample	Budget		Short	Medium	Long	Overall	
Qwen2.5-VL-7B	Original	256	-	60.3	75.0	61.8	51.0	62.6	50.0
	Retrieval	256	64	60.9	75.4	66.7	54.8	65.6	56.2
	Similarity	256	64	60.8	74.0	64.7	54.3	64.3	53.6
	Ours	256	64	**61.6**	**75.8**	**65.2**	**56.1**	**65.7**	**58.4**
	Original	128	-	59.1	73.1	60.0	49.6	60.9	47.2
	Retrieval	128	32	60.2	**74.6**	**64.8**	53.3	**64.2**	48.4
	Similarity	128	32	60.0	73.3	60.9	51.6	61.9	47.6
	Ours	128	32	**60.6**	74.1	63.2	**53.9**	63.7	**51.6**

Table 6: Long video benchmark results on Qwen2.5-VL-7B. Our method consistently achieves the best performance across most benchmarks, with improvements of up to 3.1% on VideoMME and 8.4% on MLVU over the baselines, demonstrating strong effectiveness and generalization.

Model	Method	ActivityNet-QA	
		Accuracy	Score
Qwen2.5-VL	Original	52.1	2.96
	Retrieval	52.7	2.98
	Similarity	53.1	2.99
	Ours	**54.3**	**3.07**

Table 7: Short video benchmark results on Qwen2.5-VL. Our method achieves the highest accuracy and score, outperforming all baselines.

Model	Method	EgoSchema	VideoMME
Qwen2-VL	Original	66.2	60.4
	LongVU	67.2	62.3
	Ours	**67.8**	**62.9**
LLaVA-OneVision	Original	60.1	58.0
	LongVU	**60.3**	59.3
	Ours	**60.3**	**60.6**
Qwen2.5-VL	Original	60.3	62.6
	LongVU	61.6	64.5
	Ours	**61.6**	**65.7**

Table 8: Comparison with training-free LongVU (256-64). Our method consistently outperforms the reproduced LongVU across models and benchmarks, while offering more precise control over the token count.

Respond with only the letter (A, B, C, or D) of the correct option.
Which elements are depicted in the painting introduced by the video?
A. A little girl and a red balloon.
B. A little boy and a red balloon.
C. A little girl and a blue balloon.
D. An adult and a blue balloon.

Here is an example for EgoSchema:

Respond with only the letter (A, B, C, D or E) of the correct option.
Identify the recurring actions in the video and briefly discuss their significance to the overall narrative.
A. C constantly organizing a plastic box, suggesting her obsession with tidiness
B. C and the boy taking turns throwing objects out of the window, showcasing a game
C. C pouring water and conversing with the boy, highlighting routine and communication
D. The boy trying to get C's attention by throwing a toy on the blanket repeatedly
E. C teaching the boy how to fold blankets properly and arrange his toys

---

And here is an example for MLVU:

Respond with only the letter (A, B, C, D, E or F) of the correct option.
In what setting does the video take place?
(A) Castle
(B) Forest
(C) Desert
(D) Countryside
(E) Ocean
(F) Campus

B.2 Prompts for Open-Ended Questions

We add the sentence "Answer the question according to the video." at the beginning of the open-ended questions. Here is an example:

Answer the question according to the video.
who did circles on the back tire of his motorcycle in the parking lot?
