QUERY-ANSWER-ALIGNED FRAME SELECTION FOR LONG VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) process multimodal inputs by converting text, images, and videos into token sequences that are then processed by a backbone language model. However, the limited visual token budget of these models makes long-form video understanding challenging. For instance, a 5-minute video at 24 frames per second (fps) contains 7,200 frames. However, practical MLLM inference typically uses only a subset of these frames, usually ranging from 8 to 64 frames. These frames are usually sampled uniformly, regardless of their relevance to the question. Recently, training-free, model-agnostic frame-selection methods have been proposed to improve video question answering. In this work, we propose two training-free, model-agnostic frame-selection methods: QV and QAV. QV scores frames using question-frame cosine similarity. QAV additionally incorporates multiple-choice answer options and scores frames by their maximum cosine similarity across the resulting question-option pairs. Both methods first construct a candidate pool by sampling video frames at a fixed rate. They then select the highest-scoring frames, up to a predefined frame budget, and restore their chronological order before passing them to the downstream MLLM. We evaluate QV and QAV on MLVU, Video-MME, and LongVideoBench using three MLLMs: Qwen2-VL-7B, LLaVA-Video-7B, and InternVL3.5-8B. Our results show that QAV improves over uniform sampling across the evaluated model-benchmark settings and achieves competitive performance against the evaluated training-free frame-selection baselines.

1 INTRODUCTION

Multimodal large language models (MLLMs), also known as multimodal models, extend large language models (LLMs) to process both visual and textual inputs. An MLLM typically consists of a front-end encoder and fusion modules (Lin et al., 2023a; Wang et al., 2024; Zhang et al., 2025b) that convert input text, images, and videos into a sequence of tokens (see Fig. 1). For example, VideoLLaMA2 (Cheng et al., 2024) can handle around 2,000 tokens, whereas VILA-V1.5 (Lin et al., 2024) supports approximately 4,000 tokens.

The input-size constraint is less restrictive for images, and these MLLMs have achieved considerable success in understanding their content. However, this constraint poses a significant challenge for long-video understanding. A 5-minute video at 24 frames per second (fps) contains 7,200 frames, which exceeds the number of frames that practical MLLM inference can process. For long-video understanding, the most common approach is to sample frames uniformly at regular intervals. However, this approach can lead to overlooking the frames containing the critical visual cues needed to answer correctly because it does not account for the question being asked.

Fortunately, the answers to most video-related questions are often contained in a few, usually consecutive, video frames (Zhu et al., 2026). Thus, one can divide long-video question answering into two subproblems: (i) selecting the video frames relevant to the question and (ii) generating an answer with an MLLM using the selected frames and the query.

Selecting meaningful frames from a long video can be computationally challenging. This is because decoding and evaluating every frame require significant storage space, CPU, and GPU memory. Moreover, it also increases inference time, particularly for longer videos (Zhu et al., 2026). Thus, a

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

practical frame-selection method should first reduce the candidate frame pool via fixed-rate sampling and then select a small subset of frames most useful for answering the given question.

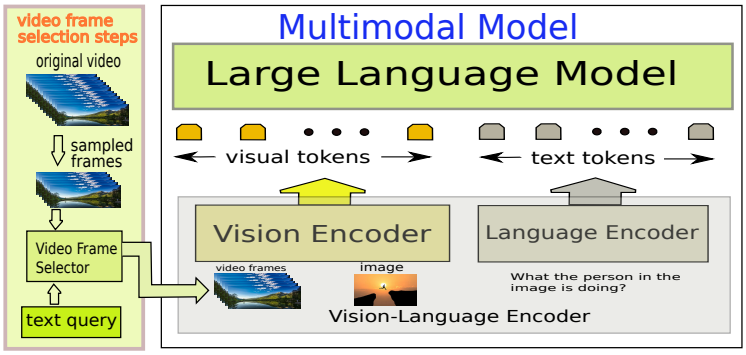


Figure 1: Block diagram of a video frame selector for a multimodal large language model (MLLM). The left column shows the three main steps of the video frame selection module. The frame selection module is training-free, MLLM-agnostic, and plug-and-play.

A common solution to the candidate-pool size problem is query-aware video frame downsampling (Lin et al., 2023b; Maaz et al., 2024; Wang et al., 2022; Zhang et al., 2025a). This is motivated by the fact that blind uniform downsampling to meet an MLLM's token budget can miss question-relevant frames (Tang et al., 2025; Zhang et al., 2025a; Zhu et al., 2026). Query-aware frame selection addresses these limitations by leveraging the question to identify frames more likely to contain the necessary visual cues. Unlike blind uniform sampling, which selects frames based solely on their temporal positions, query-aware selection evaluates candidate frames based on their relevance to the question. For example, Figure 5 in Appendix A.9 compares uniform and question-answer-aligned frame selection. The uniformly sampled frames in Fig. 5a contain only one query-relevant frame. On the other hand, the question-answer-aligned selection in Fig. 5b includes 13 relevant frames.

The left column in Fig. 1 shows a high-level diagram of a query-aware, training-free, MLLM-agnostic, and plug-and-play frame-selection module (see Figures 2 and 3 for details of our proposed methods). There are two independent but synergistic components: one for video frame sub-sampling to reduce the frame-selection search space and the other for query-aware video frame selection. Three recent methods follow the general structure in Fig. 1: Q-Frame (Zhang et al., 2025a), AKS (Tang et al., 2025), and FOCUS (Zhu et al., 2026). AKS and FOCUS use BLIP (Li et al., 2022) to compute frame alignment scores for frame selection. Q-Frame (Zhang et al., 2025a) produces vision tokens and query tokens using CLIP (Radford et al., 2021), and their dot product is then further processed for frame selection. In this work, we introduce two frame-selection methods, described next.

Our proposed methods use regular-interval frame sub-sampling, as in Q-Frame, but differ in two key respects. First, we replace the inner-product ranking in Q-Frame with cosine similarity, which is invariant to embedding magnitudes and empirically improves downstream accuracy in many of our evaluated settings. Second, QAV integrates the answer choices provided in multiple-choice questions. For each candidate frame, QAV computes cosine similarity against every question-answer pair and assigns the frame its maximum score across all pairs (see Section 3.2). This answer-aware scoring directs selection toward frames that are relevant to what the answer looks like, rather than just to what the question asks. To validate our frame-selection methods, we evaluate them using three open-source MLLMs (InternVL3.5-8B (Wang et al., 2025), Qwen2-VL-7B (Wang et al., 2024), and LLaVA-Video-7B (Zhang et al., 2024b)) across three video-understanding benchmarks (Video-MME (Fu et al., 2024), LongVideoBench (Wu et al., 2024), and MLVU (Zhou et al., 2024)).

Our main contributions are fourfold: (1) We propose a cosine-similarity-based query-video frame-scoring method using token vectors generated by pretrained vision-language encoders; (2) We introduce a query-answer-video (QAV) aligned frame-selection method that incorporates multiple-choice answer options as inference-time cues through max-over-options scoring; (3) Our methods are training-free, model-agnostic, and plug-and-play, preserve a fixed downstream MLLM frame budget, and restore selected frames to their original temporal order; and (4) We validate the proposed methods on three open-source MLLMs and three long-video understanding benchmarks, including question-category and video-duration-level comparisons with recent frame-selection baselines.

2 PRELIMINARIES AND RELATED WORK

2.1 PRELIMINARIES

Let $[n]$ denote a set of n integer indices; scalars, vectors, and matrices are denoted by non-bold italic, bold lowercase, and bold uppercase letters, respectively. Let $\mathcal{V}^{(G)} = (\mathbf{V}_i^{(G)} \mid i \in [T])$ denote a video of T frames, $\mathcal{V}^{(S)} = (\mathbf{V}_i^{(G)} \mid i \in [T_s])$ a subsequence of T_s sampled frames with $[T_s] \subseteq [T]$, and $\mathcal{V}^{(M)} = (\mathbf{V}_i^{(S)} \mid i \in [k])$ a subsequence of k selected frames with $[k] \subseteq [T_s]$. A text query Q may have n_a answer choices $A = \{a_i \mid i \in [n_a]\}$, forming question-answer pairs $QA = \{Qa_i \mid i \in [n_a]\}$. An MLLM $\mathcal{M}_{\theta_m}(\cdot,\cdot)$ processes Q (or Qa_i) and $\mathcal{V}^{(M)}$ to produce an answer $\hat{A} = \mathcal{M}_{\theta_m}(Q, \mathcal{V}^{(M)})$. A pretrained vision-language model $\mathcal{VL}_{\theta_{vl}}(\cdot)$, such as CLIP (Radford et al., 2021) or LongCLIP (Zhang et al., 2024a), maps a frame $\mathbf{V}$ and text Q to token vectors $\mathbf{vt}$ and $\mathbf{qt}$ of the same dimension.
Definition 1 (Cosine similarity). For two nonzero vectors $\mathbf{x}$ and $\mathbf{y}$ of the same dimension, their cosine similarity is $\mathrm{CosSim}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y} / (\|\mathbf{x}\|_2 \|\mathbf{y}\|_2)$, where $\|\cdot\|_2$ denotes the Euclidean norm.

2.2 RELATED WORK

Three lines of work aim to improve video understanding: (1) training stronger MLLMs (Zhang et al., 2025b; Wang et al., 2024; Zhang et al., 2024b), (2) training or adapting the MLLM front-end while keeping the language-model back-end frozen (Wang et al., 2023), or first training the front-end and then fine-tuning the back-end (Li et al., 2024), and (3) developing training-free, model-agnostic, plug-and-play frame-selection methods (Tang et al., 2025; Zhang et al., 2025a; Zhu et al., 2026). Because our methods belong to the third category, we briefly discuss three recent baselines used in our experiments.

The Q-Frame method (Zhang et al., 2025a) proceeds as follows. It first obtains T_S candidate frames by uniformly sampling a video of T frames. Next, it uses the CLIP (Radford et al., 2021) vision-language model to generate visual token vectors $\mathbf{vt}_i$ for $i \in [T_S]$ from the downsampled video frames and a text token vector $\mathbf{qt}$ from the query text. It computes T_S inner products $I_i = \mathbf{qt} \cdot \mathbf{vt_i}$ for $i \in [T_S]$, converts these T_S inner-product scores to a probability distribution using the *softmax* function, adds Gumbel noise to the log-probabilities, and finally selects the k frames with the highest scores. Both AKS (Tang et al., 2025) and FOCUS (Zhu et al., 2026) compute frame-query alignment scores using the BLIP (Li et al., 2022) vision-language model. AKS adaptively samples frames to select the best frame subset, whereas FOCUS (Zhu et al., 2026) first divides a video into disjoint segments. FOCUS then performs a coarse-grained search to identify a few promising segments, followed by a fine-grained search within these segments to select the most promising frames. This strategy significantly reduces the search space. FOCUS's initial video segmentation is based on an empirical evaluation of videos from two benchmark datasets (Fu et al., 2024; Wu et al., 2024). This evaluation shows that video frames exhibit high temporal autocorrelation, suggesting that coarse sampling can identify promising video segments. In this work, we compare these three methods with ours.

3 PROPOSED METHODS

Problem Statement Given a video $\mathcal{V}^{(G)}$, a text question Q, and an MLLM $\mathcal{M}_{\theta_m}(\cdot,\cdot)$ that can process Q and at most k video frames to generate an answer $\hat{A}$, the task is to select a sequence $\mathcal{V}^{(M)}$ of k possibly nonconsecutive frames from $\mathcal{V}^{(G)}$ that maximizes the probability of producing the correct answer to Q.

Need for Video Sub-Sampling To reduce storage, memory, and computation, we sample frames from $\mathcal{V}^{(G)}$ at regular intervals (1 fps in our experiments), forming the candidate set $\mathcal{V}^{(S)}$. Motivated by the temporal redundancy observed by FOCUS (Zhu et al., 2026), we assume that this set contains at least one query-relevant frame.

Best Frame Subset We treat the MLLM as a black box that returns an answer without an accessible confidence score. Let $R_{\mathcal{M}_{\theta_m}}(Q, \mathcal{V}^{(M)})$ denote a confidence-estimation function for the answer

produced from Q and a selected k-frame sequence $\mathcal{V}^{(M)}$. The ideal frame selector is

$$FS^{(ideal)}(Q, \mathcal{V}^{(S)}) = \underset{[k]\subseteq[T_s]}{\arg\max} R_{\mathcal{M}_{\theta_m}}(Q, \mathcal{V}^{(M)}) \tag{1}$$

Equation 1 is impractical to evaluate because the subset search is combinatorial and the confidence-estimation function is unavailable. We therefore use vision-language models to estimate each frame's question relevance as a proxy for its utility (Wang et al., 2023; Liang et al., 2024; Zhang et al., 2025a; Tang et al., 2025; Zhu et al., 2026).

Query and Video Frame Alignment Score Given a set of sampled video frames $\mathcal{V}^{(S)}$, selecting the most promising frames $\mathcal{V}^{(M)}$ requires a method to rate each frame's relevance to the question (Zhu et al., 2026). Existing frame-selection methods commonly use a vision-language model either to produce a relevance score directly or to generate a pair of semantic representations, one from the video frame and the other from the query text, whose alignment is then measured. Both AKS (Tang et al., 2025) and FOCUS (Zhu et al., 2026) estimate frame-query relevance using the BLIP vision-language model. Q-Frame (Zhang et al., 2025a) uses the inner product of a video-frame vector and a text-query vector. However, the inner product can produce inconsistent scores because it depends on the projection length of one vector in the direction of another vector. We will show that our approach based on cosine similarity can identify relevant video frames more effectively than these methods.

Temporal Order Preservation After selection, we restore the frames to their original temporal order before passing them to the MLLM to preserve the sequence of events.

3.1 Query-Frame Alignment Score

To address these issues in selecting the best frame subset, we need a proxy function that does not require evaluating $\mathcal{M}_{\theta_m}(\cdot,\cdot)$. As in prior work, we use a vision-language encoder such as CLIP (Radford et al., 2021) or LongCLIP (Zhang et al., 2024a).

Vision-language encoders offer a computational advantage in identifying relevant frames because they measure each frame's relevance individually. After evaluating all T_s candidate frames, one can select the k frames with the highest potential to support a correct answer. Let $\mathcal{VL}_{\theta_{vl}}(x)$ denote a vision-language model, where x is a video frame $\mathbf{V}$ or a text question Q.

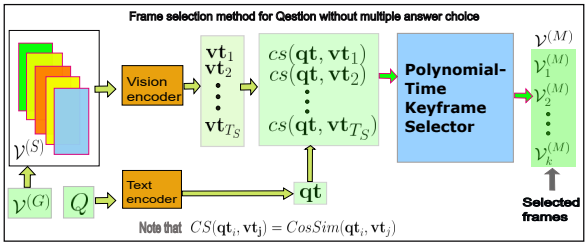


Figure 2: Proposed query-video (QV) aligned frame selection. Sampled frames and the question are encoded using a vision-language encoder. The k frames with the highest question-frame cosine similarity are selected and restored to chronological order.

These vision-language models process image and language inputs separately to produce token representations that encode semantic information from each modality. An image or text token representation can be viewed as an abstract vector representation of the input content. If an image and a text query share related semantic information, their vector representations tend to be aligned in the embedding space. For example, consider a video frame showing a person with drawings at a construction site and a question asking about that person's profession. The frame's image token vector **vt** and the question's text token vector **qt** tend to be aligned, yielding a high $CosSim(\mathbf{vt}, \mathbf{qt})$ score. By evaluating this alignment, the frame-selection method can identify frames that are relevant to the question. Figure 5 in Appendix A.9 illustrates this principle using QAV, which additionally incorporates the answer choices.

Definition 2 *Let a vision-language model $\mathcal{VL}_{\theta_{vl}}(\cdot)$ produce an image token vector $\mathbf{vt} = \mathcal{VL}_{\theta_{vl}}(\mathbf{V})$ from a video frame $\mathbf{V}$ and a text token vector $\mathbf{qt} = \mathcal{VL}_{\theta_{vl}}(Q)$ from a text input Q. We define*

$$R^{(1)}_{\mathcal{M}_{\theta_{vl}}}(Q, \mathbf{V}) = CosSim(\mathbf{qt}, \mathbf{vt}) \tag{2}$$

to estimate the alignment score between a frame $\mathbf{V}$ and a text query Q for an MLLM.

Using Equation 2, we estimate the alignment of a query Q with each frame in $\mathcal{V}^{(S)}$. Let $SQV = \{R^{(1)}_{\mathcal{M}_{\theta_{vl}}}(Q, \mathbf{V_i}) | \mathbf{V_i} \in \mathcal{V}^{(S)}\}$ be the set of alignment scores for all frames in $\mathcal{V}^{(S)}$. We use these scores to select the k most promising frames, as detailed in Algorithm 1 in the appendix. Figure 2 summarizes the frame-selection procedure.

3.2 Query-Answer-Frame Alignment Score

The query-frame alignment method described above may be further improved when the question includes a set of possible answers, i.e., answer choices. To use this information, we pair each answer choice a_i with the question to construct n_a question-answer pairs, where n_a is the number of answer choices. As shown in Fig. 3, we compute n_a alignment scores for each frame and retain its highest score (shown in the yellow column) for the final frame-selection step. Algorithm 2 in the appendix provides the details.

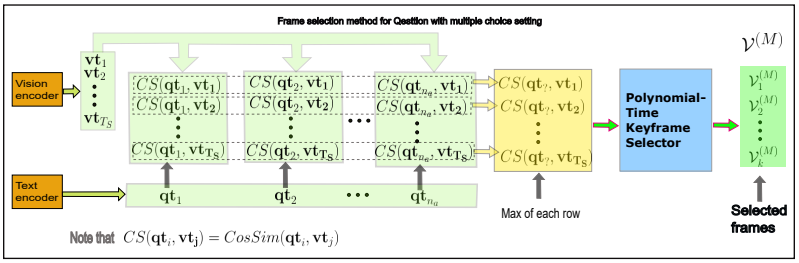


Figure 3: Proposed query-answer-video (QAV) aligned frame selection. Each frame is scored by its maximum cosine similarity across all question-answer pairs (yellow column). The k highest-scoring frames are selected and restored to chronological order.

3.3 Polynomial-time Keyframe Selection for MLLMs

Equation 2 estimates each frame's potential contribution to correctly answering a query Q. The AKS and FOCUS methods also use VLMs to approximate $R^{(1)}_{\mathcal{M}_{\theta_m}}(Q, \mathbf{V})$. Q-Frame uses the inner product, whereas our methods use cosine similarity between $\mathbf{vt}$ and $\mathbf{qt}$, which are generated by VLMs. Using a VLM to estimate each frame's potential contribution to answering a question reduces Eq. 1 to

$$FS^{(vq)}(Q, \mathcal{V}^{(S)}) = \underset{[k]\subseteq[T_s]}{\arg\max} \left(R^{(1)}_{\mathcal{M}_{\theta_{vl}}}(Q, \mathbf{V}_i) \,|\, \mathbf{V}_i \in \mathcal{V}^{(M)} \right) \tag{3}$$

However, the combinatorial search remains computationally intractable.

To obtain a polynomial-time algorithm, a frame-level *cooperative* property among video frames, analogous to the contribution of players in an n-person cooperative game (Shapley, 1953), is necessary. This property was assumed in previous plug-and-play (PaP) methods. We state it explicitly below.

Assumption 1 *If $\mathbf{V}_1, \mathbf{V}_2$, and $\mathbf{V}_3$ are three video frames such that $R^{(1)}_{\mathcal{M}_{\theta_m}}(Q, \mathbf{V}_1) > R^{(1)}_{\mathcal{M}_{\theta_m}}(Q, \mathbf{V}_2) > R^{(1)}_{\mathcal{M}_{\theta_m}}(Q, \mathbf{V}_3)$, then $R^{(2)}_{\mathcal{M}_{\theta_m}}(Q, (\mathbf{V}_1, \mathbf{V}_2)) > R^{(2)}_{\mathcal{M}_{\theta_m}}(Q, (\mathbf{V}_1, \mathbf{V}_3))$.*

The above assumption about three video frames leads to the design of a polynomial-time algorithm for selecting the k video frames with the highest potential to produce the correct answer to a query from the T_s video frames in a video $\mathcal{V}^{(S)}$. We state this formally in the following lemma.

Lemma 1 *Given a function $R^{(1)}_{\mathcal{M}_{\theta_m}}(Q, \mathbf{V})$ for estimating the potential of a frame $\mathbf{V}$ to answer the query Q correctly and a set of frames $\mathcal{V}^{(S)}$, the best k frames under Assumption 1 are the k highest-scoring frames among the T_s frames in $\mathcal{V}^{(S)}$.*

To select k frames from the T_s frames in $\mathcal{V}^{(S)}$, we first compute $R^{(1)}_{\mathcal{M}_{\theta_m}}(Q, \mathbf{V}_i)$ for all frames $\mathbf{V}_i \in \mathcal{V}^{(S)}$ and store the scores in a max-heap. Building the heap takes $O(T_s)$ time. We then repeatedly select the highest-scoring frame and update the heap, which takes $O(\log(T_s))$ steps per selection. Thus, selecting the k highest-scoring frames takes $O(T_s + k\log(T_s))$ steps. When $k = O(T_s)$, the worst-case complexity is $O(T_s \log T_s)$. Algorithms 2 and 1 in the appendix provide detailed steps for implementing Lemma 1.

4 Experimental Evaluation

We compare our two proposed methods (QV and QAV) with uniform sampling and four training-free frame-selection baselines across three benchmarks using three open-source MLLMs.

4.1 Experimental Setup

Our experiments use three public benchmark datasets: **MLVU** (Zhou et al., 2024), **Video-MME** (Fu et al., 2024), and **LongVideoBench** (Wu et al., 2024), which are also used in the literature to evaluate frame-selection methods such as Q-Frame (Zhang et al., 2025a), AKS (Tang et al., 2025), and FOCUS (Zhu et al., 2026). These datasets cover different video durations, domains, and reasoning skills. **MLVU** (Zhou et al., 2024) includes long videos covering seven closed-ended task categories: Topic Reasoning (TR), Anomaly Recognition (AR), Needle QA (NQA), Ego Reasoning (ER), Plot QA (PQA), Action Order (AO), and Action Count (AC). **Video-MME** (Fu et al., 2024) assesses video understanding across short, medium, and long videos in diverse domains. **LongVideoBench** (Wu et al., 2024) focuses on long-context video understanding, with questions that require reasoning across multiple video segments. None of the methods in our experiments uses gradient updates, model fine-tuning, or dataset-specific training. We use three open-source MLLMs: **Qwen2-VL-7B** (Wang et al., 2024), **LLaVA-Video-7B** (Zhang et al., 2024b), and **InternVL3.5-8B** (Wang et al., 2025). Further implementation details appear in Appendix A.

Fair Comparison Protocol For a fair comparison, we evaluate the frame-selection methods under the same downstream MLLM and benchmark settings whenever possible. Specifically, we (i) maintain a fixed budget of $k = 16$ selected frames for uniform sampling, AKS, FOCUS, MaxInfo, and our QV/QAV methods, (ii) test Q-Frame using the same 16-frame budget on all three MLLMs, (iii) use BLIP-based frame selection for AKS and FOCUS and LongCLIP-L for Q-Frame, QV, and QAV, (iv) report the same seven closed-ended MLVU task categories for all MLVU comparisons, and (v) report results for all duration groups, both with and without subtitles, on Video-MME and LongVideoBench. Where possible, we adapt the released evaluation code of Q-Frame, AKS, and FOCUS to the benchmark, model, and preprocessing requirements of our experiments. In the main comparisons, QV and QAV use the 224-pixel LongCLIP-L checkpoint, whereas Q-Frame uses the 336-pixel LongCLIP-L checkpoint specified in its released code. All methods in the main tables use a 16-frame budget, but the retrieval encoder configurations and downstream visual-token budgets are not matched. A separate shared-encoder and matched-budget comparison is reported in Appendix A.7. We use accuracy as the evaluation metric.

4.2 Comparison with SOTA Methods

We compare our methods with four training-free frame-selection baselines: Q-Frame (Zhang et al., 2025a), AKS (Tang et al., 2025), FOCUS (Zhu et al., 2026), and MaxInfo (Li et al., 2026). For a comprehensive comparison, we report results by MLVU task category and video-duration group, with and without subtitles where applicable.

MLVU benchmark As shown in Table 1, QAV obtains the best overall accuracy on all three MLLMs. For Qwen2-VL, QAV obtains 69.2%. Among the prior frame-selection baselines, Q-Frame is the closest competitor at 68.03%, so QAV improves over it by approximately 1.17 percentage

Table 1: Comparison of different methods on MLVU. QV uses the question only; QAV uses the question and answer choices. All methods use a 16-frame budget. Within each MLLM block, the best performance in each column is boldfaced, and the second-best distinct performance is underlined. Ties receive the same formatting.

Methods	Size	#Frames	TR	AR	NQA	ER	PQA	AO	AC	Avg.
Qwen2-VL (Uniform)	7B	16	**86.7**	**71.5**	72.4	56.3	60.9	42.9	22.8	60.4
Qwen2-VL + Q-Frame	7B	16	80.99	60.50	81.69	**69.60**	72.73	51.35	41.26	68.03
Qwen2-VL + FOCUS	7B	16	80.99	65.00	81.13	62.78	69.76	47.88	**41.75**	66.19
Qwen2-VL + AKS	7B	16	82.1	64.0	78.9	56.8	63.3	44.4	40.3	62.7
Qwen2-VL + MaxInfo	7B	16	83.3	69.5	78.0	58.8	62.5	42.1	31.1	62.2
Qwen2-VL + QV	7B	16	84.4	67.5	**87.0**	65.1	71.6	50.6	28.6	67.7
Qwen2-VL + QAV	7B	16	81.0	70.0	86.2	67.6	**75.0**	**55.2**	29.6	**69.2**
LLaVA-Video + Uniform	7B	16	81.0	59.5	77.2	61.4	68.8	44.8	34.5	63.5
LLaVA-Video + Q-Frame	7B	16	79.47	57.00	82.54	70.17	74.03	52.90	**51.46**	69.23
LLaVA-Video + FOCUS	7B	16	81.0	59.0	83.4	69.0	73.8	49.8	49.5	69.0
LLaVA-Video + AKS	7B	16	82.1	57.5	77.7	61.9	70.3	45.9	48.5	65.5
LLaVA-Video + MaxInfo	7B	16	**84.8**	**65.5**	82.8	65.1	65.5	52.9	45.6	67.2
LLaVA-Video + QV	7B	16	81.7	50.5	**84.5**	70.7	75.0	49.4	47.6	68.8
LLaVA-Video + QAV	7B	16	79.1	57.0	**84.5**	**71.0**	**80.1**	**57.1**	48.1	**71.3**
InternVL3.5 + Uniform	8B	16	82.13	67.00	80.28	62.93	64.56	51.35	37.38	65.07
InternVL3.5 + AKS	8B	16	80.99	**71.00**	81.41	59.77	67.90	51.74	**47.57**	66.82
InternVL3.5 + FOCUS	8B	16	79.47	61.50	84.23	**67.82**	71.99	56.76	**47.57**	69.12
InternVL3.5 + Q-Frame	8B	16	79.09	67.50	85.07	65.06	72.73	53.67	46.12	69.00
InternVL3.5 + MaxInfo	8B	16	**82.5**	67.5	84.2	64.4	67.2	56.0	41.8	67.7
InternVL3.5 + QV	8B	16	76.43	60.50	**86.20**	66.38	72.91	56.37	40.29	68.25
InternVL3.5 + QAV	8B	16	75.67	57.50	**86.20**	66.09	**75.88**	**70.27**	39.81	**70.18**

points. For LLaVA-Video, QAV obtains 71.3%. This accuracy is approximately 2.07 percentage points higher than that of Q-Frame, the strongest prior baseline at 69.23%. For InternVL3.5, QAV obtains 70.18%, improving over the strongest prior baseline, FOCUS at 69.12%, by 1.06 percentage points. All these comparisons use a 16-frame budget.

Video-MME benchmark Video-MME supports evaluation with and without subtitles. As shown in Table 2, we evaluate all methods under both settings. On Qwen2-VL, QAV achieves the best overall accuracy in both cases. Without subtitles, it obtains 60.90%, outperforming the strongest prior baseline, Q-Frame at 58.74%, by 2.16 percentage points. With subtitles, it obtains 62.93%, outperforming the strongest prior baseline, Q-Frame at 62.04%, by 0.89 percentage points. On LLaVA-Video, AKS is marginally better than QAV without subtitles, by 0.12 percentage points (61.93% vs. 61.81%). With subtitles, however, QAV achieves the best overall accuracy, improving over the strongest prior baseline, Q-Frame, by 1.07 percentage points (64.70% vs. 63.63%). On InternVL3.5, FOCUS leads without subtitles at 61.56%, compared with 60.89% for QAV. With subtitles, QAV obtains the best overall accuracy, 64.52%, compared with 63.37% for both FOCUS and Q-Frame.

LongVideoBench LongVideoBench supports evaluation with and without subtitles. As shown in Table 3, we report results under both settings for all methods. On Qwen2-VL, QAV achieves the best overall accuracy in both cases. Without subtitles, it obtains 58.80%, outperforming the strongest prior baseline, FOCUS, by 0.76 percentage points. With subtitles, it obtains 60.20%, outperforming the strongest prior baseline, Q-Frame at 59.01%, by 1.19 percentage points. On LLaVA-Video, QAV also achieves the best overall accuracy in both settings. Without subtitles, it obtains 61.32%, improving over the strongest prior baseline, AKS, by 1.11 percentage points. With subtitles, it obtains 63.49%, improving over the strongest prior baseline, Q-Frame at 60.51%, by 2.98 percentage points. On InternVL3.5, QAV obtains the best overall accuracy at 63.20% without subtitles and 66.57% with subtitles. The strongest prior baseline in both settings is Q-Frame, at 61.63% and 64.47%, respectively.

Table 2: Video-MME results (without / with subtitles). All methods use a 16-frame budget. Within each MLLM block, the best performance in each column and subtitle setting is boldfaced, and the second-best distinct performance is underlined. Ties receive the same formatting.

Model	LLM Size	#Frames	Video-MME (*wo / w subs*)			
			Overall *17min*	Short *1.3min*	Medium *9min*	Long *41min*
Qwen2-VL + Uniform	7B	16	56.07 / 57.44	68.22 / 68.30	52.33 / 55.60	47.67 / 48.40
Qwen2-VL + Q-Frame	7B	16	58.74 / 62.04	68.67 / 70.67	57.22 / 60.78	50.33 / **54.67**
Qwen2-VL + AKS	7B	16	55.19 / 54.70	64.44 / 64.11	53.22 / 53.00	47.89 / 47.00
Qwen2-VL + FOCUS	7B	16	57.07/59.33	67.56/68.44	54.44/56.67	49.22/52.89
Qwen2-VL + MaxInfo	7B	16	55.96 / 55.78	66.67 / 66.89	52.67 / 52.33	48.56 / 48.11
Qwen2-VL + QV	7B	16	57.11 / 61.04	68.78 / 70.22	54.33 / 59.89	48.22 / 53.00
Qwen2-VL + QAV	7B	16	**60.9 / 62.93**	**70.7 / 72.67**	**60.8 / 62.56**	**51.1** / 53.56
LLaVA-Video (Uniform)	7B	16	59.56/62.07	71.33/74.11	58.22/60.22	49.11/51.89
LLaVA-Video + Q-Frame	7B	16	59.30 / 63.63	70.33 / 73.89	57.44 / 62.33	50.11 / **54.67**
LLaVA-Video +AKS	7B	16	**61.93** / 63.11	73.11 / 74.67	60.89 / 61.00	51.78 / 53.67
LLaVA-Video +FOCUS	7B	16	60.52 / 60.70	70.40 / 69.70	58.80 / 59.00	**52.30** / 53.40
LLaVA-Video + MaxInfo	7B	16	60.41 / 60.37	72.67 / 72.67	57.33 / 57.00	51.22 / 51.44
LLaVA-Video + QV	7B	16	58.07 / 62.15	69.56 / 72.78	56.33 / 60.78	48.33 / 52.89
LLaVA-Video + QAV	7B	16	61.81 / **64.70**	**73.22 / 76.67**	**61.11 / 63.56**	51.11 / 53.89
InternVL3.5 + Uniform	8B	16	59.41 / 62.41	69.78 / 74.67	59.33 / 60.11	49.11 / 52.44
InternVL3.5 + AKS	8B	16	58.59 / 61.52	68.33 / 70.67	57.22 / 60.44	50.22 / 53.44
InternVL3.5 + FOCUS	8B	16	**61.56** / 63.37	71.44 / 72.78	61.22 / 61.11	**52.00 / 56.22**
InternVL3.5 + Q-Frame	8B	16	59.70 / 63.37	71.78 / 74.22	58.89 / 61.67	48.44 / 54.22
InternVL3.5 + MaxInfo	8B	16	59.33 / 62.00	69.78 / 72.00	59.89 / 60.22	48.33 / 53.78
InternVL3.5 + QV	8B	16	58.85 / 61.85	71.11 / 73.00	57.44 / 59.00	48.00 / 53.56
InternVL3.5 + QAV	8B	16	60.89 / **64.52**	**72.89 / 75.78**	**61.67 / 63.78**	48.11 / 54.00

In summary, QAV obtains the best overall accuracy in most evaluated settings across the three benchmarks, whereas QV does not consistently outperform the prior baselines. Among the prior frame-selection baselines, the closest competitor depends on the MLLM, benchmark, and subtitle setting, as reported in Tables 1, 2, and 3. These results suggest that answer-aware frame selection provides a competitive alternative to uniform sampling and recent training-free frame-selection baselines under a 16-frame budget, although some baselines remain stronger for particular task categories and video durations.

4.2.1 Performance for Subcategories

MLVU benchmark MLVU contains seven closed-ended and two open-ended task categories, as shown in Fig. 4 in Appendix A; we evaluate the seven closed-ended categories. On Qwen2-VL, uniform sampling performs best on the two holistic categories, TR and AR, exceeding the best result among our methods by 2.3 and 1.5 percentage points, respectively. Our methods lead on NQA, PQA, and AO, while Q-Frame leads on ER and FOCUS performs best on AC. On LLaVA-Video, Q-Frame leads on AC; our methods lead on NQA, ER, PQA, and AO; and MaxInfo leads on TR and AR. On InternVL3.5, our methods lead on NQA, PQA, and AO; MaxInfo leads on TR, AKS on AR, and FOCUS on ER, while AKS and FOCUS tie on AC. All methods in this comparison use a 16-frame budget.

4.2.2 Effect of Video Length

Video-MME benchmark Video-MME has three duration groups and can be evaluated with or without subtitles, giving six settings. As shown in Table 2, QAV achieves the best result in five settings on Qwen2-VL; Q-Frame leads on long videos with subtitles, at 54.67% compared with 53.56% for QAV. On LLaVA-Video, QAV leads in the four short- and medium-video settings, while FOCUS leads on long videos without subtitles and Q-Frame leads on long videos with subtitles. On InternVL3.5, QAV leads in the four short- and medium-video settings, while FOCUS leads on long videos in both subtitle settings.

Table 3: LongVideoBench results (without / with subtitles). All methods use a 16-frame budget. Within each MLLM block, the best performance in each column and subtitle setting is boldfaced, and the second-best distinct performance is underlined. Ties receive the same formatting.

Model	LLM Size	#Frames	LongVideoBench (*wo / w subs*)				
			Overall	8s–15s	15s–60s	3m–10m	15m–60m
Qwen2-VL (+Uniform)	7B	16	54.10 / 55.80	66.70 / 65.10	65.10 / 68.00	52.70 / 57.50	47.50 / 47.70
Qwen2-VL + FOCUS	7B	16	58.04 / 58.19	66.14 / **67.73**	66.28 / 66.86	56.07 / 58.01	**54.26** / 52.48
Qwen2-VL +AKS	7B	16	54.90 / 54.45	62.43 / 60.85	61.63 / 62.21	56.55 / 54.85	49.11 / 49.65
Qwen2-VL + Q-Frame	7B	16	57.97 / 59.01	66.67 / 67.20	69.19 / 71.51	58.98 / **62.14**	50.89 / 50.18
Qwen2-VL + MaxInfo	7B	16	53.48 / 54.53	65.08 / 65.61	68.02 / 69.19	52.43 / 53.64	45.92 / 46.99
Qwen2-VL + QV	7B	16	58.12 / 59.01	**68.25** / 65.61	69.19 / 70.93	57.52 / 59.47	51.77 / 52.84
Qwen2-VL + QAV	7B	16	**58.80** / **60.20**	65.60 / 65.10	**69.80** / **72.7**	**59.70** / 61.20	52.50 / **54.10**
LLaVA-Video + Uniform	7B	16	56.90 / 59.30	67.20 / 69.31	70.35 / 72.09	53.64 / 57.28	51.69 / 53.48
LLaVA-Video + Q-Frame	7B	16	59.39 / 60.51	67.20 / 69.84	69.77 / 72.67	58.50 / 60.68	54.26 / 53.55
LLaVA-Video + FOCUS	7B	16	58.86 / 60.06	62.96 / 64.55	68.61 / 67.44	58.25 / 59.71	54.97 / 56.56
LLaVA-Video + MaxInfo	7B	16	57.52 / 57.59	67.72 / **70.37**	69.77 / 70.35	54.61 / 56.31	52.48 / 50.35
LLaVA-Video + AKS	7B	16	60.21 / 59.69	**70.90** / 67.72	69.77 / 73.26	58.74 / 58.01	54.79 / 54.08
LLaVA-Video + QV	7B	16	59.99 / 62.83	67.20 / 69.31	**73.26** / **75.00**	58.01 / 60.92	54.96 / **58.33**
LLaVA-Video + QAV	7B	16	**61.32** / **63.49**	67.20 / 69.31	72.09 / 73.84	**59.71** / **63.59**	**57.22** / 58.29
InternVL3.5 (+Uniform)	8B	16	58.71 / 61.78	68.78 / **73.54**	**76.16** / 73.84	56.55 / 60.68	51.60 / 54.96
InternVL3.5 + AKS	8B	16	59.76 / 62.53	63.49 / 71.43	68.60 / 65.70	59.47 / 63.83	56.03 / 57.62
InternVL3.5 + FOCUS	8B	16	60.28 / 62.68	61.38 / 66.14	68.60 / 73.26	63.35 / 64.81	55.14 / 56.74
InternVL3.5 + MaxInfo	8B	16	58.41 / 61.03	68.78 / **73.54**	74.42 / 75.00	57.77 / 59.47	50.53 / 53.72
InternVL3.5 + Q-Frame	8B	16	61.63 / 64.47	68.78 / **73.54**	72.09 / 72.67	62.86 / 66.26	55.14 / 57.62
InternVL3.5 + QV	8B	16	61.48 / 64.92	68.78 / **73.54**	72.67 / **76.16**	63.11 / 66.75	54.43 / 57.27
InternVL3.5 + QAV	8B	16	**63.20** / **66.57**	**69.31** / **73.54**	72.09 / 74.42	**64.08** / **67.96**	**57.80** / **60.82**

LongVideoBench LongVideoBench has four duration groups and can be evaluated with or without subtitles, giving eight settings. As shown in Table 3, our methods achieve the best result in five settings on Qwen2-VL, while FOCUS leads in two and Q-Frame in one. On LLaVA-Video, our methods lead in six settings; AKS and MaxInfo lead on the shortest videos without and with subtitles, respectively. On InternVL3.5, our methods lead or tie for the best result in seven settings; uniform sampling leads on 15–60-second videos without subtitles.

5 Conclusion and Future Work

In this paper, we presented training-free, model-agnostic frame selection methods for long-video understanding under a fixed MLLM frame budget. The first method, **QV**, ranks candidate frames by cosine similarity between frame and question embeddings. The second method, **QAV**, extends QV by scoring each frame against all question-answer pairs and taking the per-frame maximum. Experiments on MLVU, Video-MME, and LongVideoBench with three open-source MLLMs show that query- and answer-aligned selection passes more useful visual evidence to the downstream model than uniform sampling and recent training-free baselines in many settings. Future work includes extending QAV to support open-ended questions, exploring lightweight encoder adapters, and using an adaptive frame budget rather than a fixed k.

AI use statement

We used generative AI tools to improve the grammar and clarity of the manuscript. Generative AI was also used for debugging and validation of evaluation scripts. The research questions and proposed methods were developed by the authors. The authors reviewed the materials and take responsibility for the results and conclusions.

Reproducibility statement

Section 3 describes the proposed QV and QAV methods, and Appendix A.3 provides their pseudocode. The benchmark datasets, downstream MLLMs, and evaluation protocol are described in Section 4.1, with additional implementation and computing-resource details in Appendix A.1. The configurations for the encoder, answer-option, answer-aware baseline, shared-encoder, and frame-budget ablations are described in their respective appendix sections.

REFERENCES

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

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

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pp. 12888–12900. PMLR, 2022. URL https://proceedings.mlr.press/v162/li22n.html.

Pengyi Li, Irina Abdullaeva, Alexander Gambashidze, Andrey Kuznetsov, and Ivan Oseledets. Maxinfo: A training-free key-frame selection method using maximum volume for enhanced video understanding. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 7198–7207, 2026.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 26763–26773, 2024.

Hao Liang, Jiapeng Li, Tianyi Bai, Xijie Huang, Linzhuang Sun, Zhengren Wang, Conghui He, Bin Cui, Chong Chen, and Wentao Zhang. Keyvideollm: Towards large-scale video keyframe selection. arXiv preprint arXiv:2407.03104, 2024.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. 2023a. https://arxiv.org/abs/2311.10122.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 26689–26699, 2024.

Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, Jiaming Han, Siyuan Huang, Yichi Zhang, Xuming He, Hongsheng Li, and Yu Qiao. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. 2023b. https://arxiv.org/abs/2311.07575.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and et al. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning, pp. 8748–8763. PMLR, 2021.

Lloyd S. Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker (eds.), Contributions to the Theory of Games II, volume 28 of Annals of Mathematical Studies, pp. 307–317. Princeton University Press, Princeton, NJ, 1953.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 29118–29128, 2025.

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

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265, 2025. URL https://arxiv.org/abs/2508.18265.

Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, Sen Xing, Guo Chen, Junting Pan, Jiashuo Yu, Yali Wang, Limin Wang, and Yu Qiao. Internvideo: General video foundation models via generative and discriminative learning, 2022. URL https://arxiv.org/abs/2212.03191.

Yizhou Wang, Ruiyi Zhang, Haoliang Wang, Uttaran Bhattacharya, Yun Fu, and Gang Wu. Vaquita: Enhancing alignment in llm-assisted video understanding. arXiv preprint arXiv:2312.02310, 2023.

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

Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-CLIP: Unlocking the long-text capability of clip. In European conference on computer vision, pp. 310–325. Springer, 2024a.

Shaojie Zhang, Jiahui Yang, Jianqin Yin, Zhenbo Luo, and Jian Luan. Q-frame: Query-aware frame selection and multi-resolution adaptation for video-llms. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 22056–22065, 2025a.

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

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

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

Zirui Zhu, H Xu, Y Luo, Y Liu, K Sarkar, Z Yang, and Y You. FOCUS: Efficient keyframe selection for long video understanding. In International Conference on Learning Representations (ICLR), 2026. URL https://openreview.net/forum?id=1OQKqLFcbB.

A Appendix

A.1 Details of Experimental Setup

Our experiments evaluate multiple-choice video question answering on MLVU, Video-MME, and LongVideoBench. Given a video (V), a question (Q), and answer choices (A), a frame selection method selects frames subject to a budget k. The selected frames are restored to their original temporal order and then sent to the downstream MLLM along with the question and all answer choices. In Tables 1, 2, and 3, uniform sampling, AKS, FOCUS, Q-Frame, MaxInfo, QV, and QAV all use a budget of $k = 16$ frames. This matches the frame budget, but does not imply identical visual-token counts across methods or MLLMs. The 44-frame multi-resolution Q-Frame configurations are reported only in the separate ablations below. We report question-answering accuracy (%); the controlled ablation tables specify their evaluation coverage.

Benchmark Datasets We evaluate three public benchmarks for long-video multiple-choice question answering. These datasets cover different video durations, domains, and reasoning skills. **MLVU** (Zhou et al., 2024) consists of long videos that cover several video understanding tasks. We evaluate the seven closed-ended categories: Topic Reasoning (TR), Anomaly Recognition (AR), Needle QA (NQA), Ego Reasoning (ER), Plot QA (PQA), Action Order (AO), and Action Count (AC). **Video-MME** (Fu et al., 2024) assesses video understanding across various lengths, short, medium, and long, spanning diverse domains. We report results both with and without subtitles. **LongVideoBench** (Wu et al., 2024) focuses on understanding long-context videos across various durations, with questions that require reasoning across multiple video segments. We report results for LongVideoBench in both subtitle-free and subtitle-augmented settings.

MLLM Models We evaluate three open-source MLLMs: **Qwen2-VL-7B-Instruct** (Wang et al., 2024), **LLaVA-Video-7B** (Zhang et al., 2024b), and **InternVL3.5-8B** (Wang et al., 2025). These are abbreviated as Qwen2-VL, LLaVA-Video, and InternVL3.5 in the result tables. All models are evaluated without finetuning or architectural modifications. The retrieval encoder used for frame selection is separate from the downstream MLLM's visual encoder. Consequently, using the same retrieval encoder alone does not establish a matched downstream visual-token budget; this is controlled explicitly in Appendix A.7.

Implementation Details For QV and QAV, we construct a candidate frame pool $\mathcal{V}^{(S)}$ by sampling at approximately 1 fps and use the 224-pixel LongCLIP-L (Zhang et al., 2024a) checkpoint as the retrieval encoder, except in the encoder ablation. In QV, each candidate frame is encoded into a visual embedding, while the question Q is encoded into a text embedding. Each frame is scored by the cosine similarity between these embeddings. QAV additionally pairs the question with each answer choice and scores each frame by its maximum cosine similarity across the resulting question-option pairs. Both methods select the highest-scoring frames, up to the budget k, and restore their chronological order. The downstream MLLM receives the question and all answer choices in both cases; the correct-answer identity is not provided to either selector. No gradient updates, model fine-tuning, or dataset-specific training are used.

For the 16-frame Q-Frame results in the main tables, we use the LongCLIP-L 336-pixel retrieval checkpoint, question-only unnormalized image-text dot products, and Gumbel-based ranking with temperature $\tau = 0.8$. Candidate frames are sampled at approximately 1 fps, with uniform subsampling to at most 128 candidates. The fixed-frame configuration allocates up to 16 frames to the high-resolution branch and none to the medium- or low-resolution branches. Selected images are resized to half their source dimensions before downstream model-specific preprocessing; fewer than 16 available candidates are not padded. These runs retain our evaluation prompts, answer parsing, and subtitle handling, rather than reproducing the complete upstream evaluation pipeline.

Compute Resources Experiments were conducted on a Linux server with four NVIDIA H100 NVL GPUs, each with 96 GB of GPU memory, dual Intel Xeon Silver 4514Y CPUs, and 1.5 TB of system memory. Each evaluation job used one H100 GPU, and independent jobs were run in parallel when multiple GPUs were available. No model training or finetuning was performed; compute was used only for frame selection and MLLM inference. Evaluation time depends on the MLLM,

benchmark, frame budget, frame resolution, and subtitle setting; we do not treat the separate ablations as equal-compute comparisons.

A.2 SUBCATEGORIES OF MLVU BENCHMARK

As shown in Fig. 4, MLVU contains seven closed-ended task categories and two open-ended task categories. Our study evaluates the seven closed-ended tasks. Table 4 gives the question counts in the 2,174-question evaluation set used for the controlled MLVU ablations. These are evaluation-set counts, not counts for the entire published benchmark.

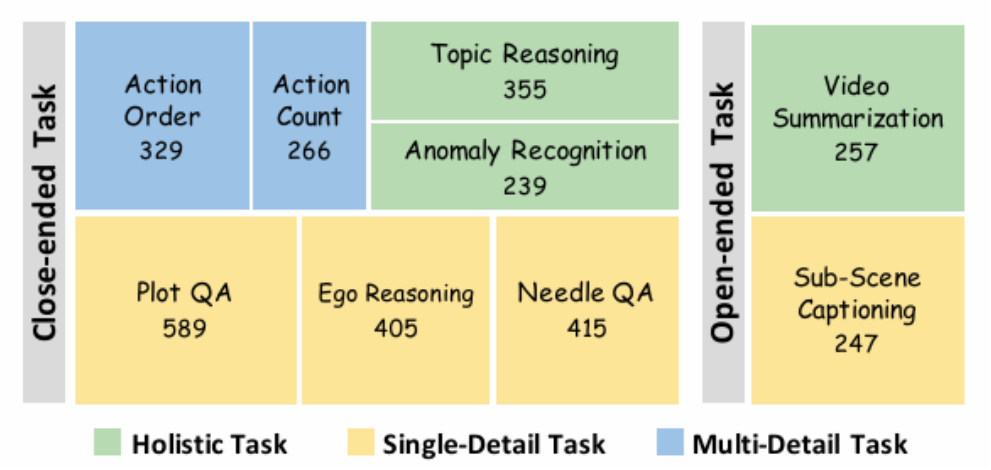


Figure 4: This is part of Figure 1 in the MLVU benchmark paper (Zhou et al., 2024). It shows nine task subcategories. Our study evaluates only the seven closed-ended tasks.

Table 4: Closed-ended task categories and question counts in the controlled MLVU ablation set. Open-ended tasks are not evaluated.

Task	TR	AR	NQA	ER	PQA	AO	AC	Total
Questions	263	200	355	352	539	259	206	2,174

A.3 Algorithms for Proposed Frame Selection

Algorithms 1 and 2 summarize the two proposed frame selection procedures used in our experiments. Both methods first construct a compact candidate set by subsampling the input video at a fixed rate, set to 1 fps in our experiments, score each candidate frame with a pretrained vision-language encoder, select the highest-scoring $\min(k, T_s)$ candidates, and restore the selected frames to their original temporal order before passing them to the downstream MLLM; $CosSim$ is defined in Definition 1 and already includes norm-based normalization.

Algorithm 1 Query-aligned video frame selection (QV)

Require: Video $\mathcal{V}^{(G)}$, question Q, frame budget k, sampling interval Δt=1 s, vision-language encoder $\mathcal{VL}_{\theta_{vl}}$

Ensure: Chronologically ordered selected frame sequence $\mathcal{V}^{(M)}$

$\mathcal{V}^{(S)} \leftarrow \textsc{SampleFrames}(\mathcal{V}^{(G)}, \Delta t)$ ▷ 1 frame per second
$T_s \leftarrow |\mathcal{V}^{(S)}|$
$\mathbf{qt} \leftarrow \mathcal{VL}^{\text{text}}_{\theta_{vl}}(Q)$
for $i = 1$ **to** T_s **do**
    $\mathbf{vt}_i \leftarrow \mathcal{VL}^{\text{image}}_{\theta_{vl}}(\mathbf{V}^{(S)}_i)$
    $SQV_i \leftarrow \text{CosSim}(\mathbf{qt}, \mathbf{vt}_i)$
end for
$\mathcal{H} \leftarrow \textsc{BuildMaxHeap}(\{(SQV_i, i)\}_{i=1}^{T_s})$
$\mathcal{I} \leftarrow \emptyset$
for $r = 1$ **to** $\min(k, T_s)$ **do**
    $(SQV, i) \leftarrow \textsc{ExtractMax}(\mathcal{H})$
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{i\}$
end for
$\mathcal{I} \leftarrow \textsc{SortAscending}(\mathcal{I})$ ▷ restore temporal order
return $\mathcal{V}^{(M)} = \left(\mathbf{V}^{(S)}_i \mid i \in \mathcal{I}\right)$

Algorithm 2 Query-answer-aligned video frame selection (QAV)

Require: Video $\mathcal{V}^{(G)}$, question Q, answer choices $A = \{a_j\}_{j=1}^{n_a}$, frame budget k, sampling interval Δt=1 s, vision-language encoder $\mathcal{VL}_{\theta_{vl}}$
Ensure: Chronologically ordered selected frame sequence $\mathcal{V}^{(M)}$

$\mathcal{V}^{(S)} \leftarrow \textsc{SampleFrames}(\mathcal{V}^{(G)}, \Delta t)$ ▷ 1 frame per second
$T_s \leftarrow |\mathcal{V}^{(S)}|$
for $j = 1$ **to** n_a **do**
  $Qa_j \leftarrow Q \,\|\, a_j$ ▷ concatenate question with j-th answer choice
  $\mathbf{qt}_j \leftarrow \mathcal{VL}_{\theta_{vl}}^{\text{text}}(Qa_j)$
end for
for $i = 1$ **to** T_s **do**
  $\mathbf{vt}_i \leftarrow \mathcal{VL}_{\theta_{vl}}^{\text{image}}(\mathbf{V}_i^{(S)})$
  for $j = 1$ **to** n_a **do**
    $S_{ij} \leftarrow \text{CosSim}(\mathbf{qt}_j, \mathbf{vt}_i)$
  end for
  $SQaV_i \leftarrow \max_{j \in [n_a]} S_{ij}$ ▷ best alignment across all choices
end for
$\mathcal{H} \leftarrow \textsc{BuildMaxHeap}(\{(SQaV_i, i)\}_{i=1}^{T_s})$
$\mathcal{I} \leftarrow \emptyset$
for $r = 1$ **to** $\min(k, T_s)$ **do**
  $(SQaV, i) \leftarrow \textsc{ExtractMax}(\mathcal{H})$
  $\mathcal{I} \leftarrow \mathcal{I} \cup \{i\}$
end for
$\mathcal{I} \leftarrow \textsc{SortAscending}(\mathcal{I})$ ▷ restore temporal order
return $\mathcal{V}^{(M)} = \left(\mathbf{V}_i^{(S)} \mid i \in \mathcal{I}\right)$

A.4 Encoder Ablation for Proposed Frame Selection

Table 5 compares CLIP and LongCLIP as the vision-language encoders used in the proposed QV and QAV frame selectors. The downstream MLLM, dataset, and frame budget are kept constant. The results indicate that both selectors can operate effectively with either encoder. In this MLVU setting, LongCLIP provides modest average improvements for both QV and QAV, although the benefits vary across different task categories.

Table 5: Encoder ablation for proposed frame selection on MLVU. All rows use Qwen2-VL-7B as the downstream MLLM, with a fixed budget of $k = 16$ selected frames. QV uses the question only; QAV uses max-over-options scoring with the question and answer choices.

Method	Encoder	#Frames	TR	AR	NQA	ER	PQA	AO	AC	Avg.
QV	CLIP	16	83.3	66.0	87.2	65.1	71.1	49.4	28.6	67.1
QV	LongCLIP	16	84.4	67.5	87.0	65.1	71.6	50.6	28.6	67.7
QAV	CLIP	16	82.1	70.0	86.1	66.7	74.4	54.4	28.6	68.8
QAV	LongCLIP	16	81.0	70.0	86.2	67.6	75.0	55.2	29.6	69.2

A.5 Effect of Answer-Option Content

We examine how the information supplied to the frame selector affects downstream question-answering accuracy. We use Qwen2-VL-7B-Instruct as the downstream MLLM and LongCLIP-L as the retrieval encoder. Across all conditions, we use the same candidate frames sampled at 1 fps, select $k = 16$ frames, and restore their chronological order. The downstream model, text prompt, and decoding procedure are held fixed; only the text used to score candidate frames changes.

We compare four conditions:

- **QV**: frames are scored using question-frame cosine similarity.

- **QAV-All**: standard QAV, using the maximum cosine similarity across all question-option pairs.
- **QAV-Correct**: frames are scored using the question paired with only the correct answer option.
- **QAV-Distractors**: frames are scored using the maximum cosine similarity across question-option pairs containing only incorrect options.

In every condition, the downstream MLLM receives the original question and all original answer choices. All four conditions are evaluated on the same 2,169 MLVU questions, excluding five questions in which the correct answer text appears in more than one option. Ground-truth labels are used only to construct the QAV-Correct and QAV-Distractors retrieval inputs. These two conditions are label-dependent diagnostics, not deployable selection methods. Standard QAV-All does not receive the correct-answer identity.

Table 6: Answer-option content ablation on MLVU. Δ denotes the change relative to QV in percentage points, computed before rounding.

Condition	# Correct	Accuracy (%)	Δ (pp)
QV	1465	67.54	0.00
QAV-All	1496	68.97	+1.43
QAV-Correct	1565	72.15	+4.61
QAV-Distractors	1387	63.95	−3.60

As shown in Table 6, QAV-All improves over QV by 1.43 percentage points without access to the correct-answer identity. Correct-option-only selection achieves 72.15%, whereas distractor-only selection achieves 63.95%. These results show that using all answer options improves accuracy over question-only selection in this setting. Using only the correct option achieves the highest overall accuracy, whereas using only incorrect options performs worse than question-only selection. However, the latter condition excludes the correct option from frame selection. Its lower accuracy may therefore reflect both the removal of useful correct-option information and the influence of incorrect options.

A.6 Answer-Aware Baseline Ablation

We examine whether incorporating answer options also benefits other frame-selection methods. We evaluate question-only and answer-aware configurations of AKS, FOCUS, and Q-Frame on the same MLVU evaluation set, using Qwen2-VL-7B-Instruct as the downstream MLLM.

For each method, let $h(v_i, t)$ denote its frame-text relevance score. The question-only condition uses $h(v_i, q)$, whereas the answer-aware condition uses

$$s_i = \max_{a \in \mathcal{A}} h(v_i, q \parallel a),$$

where $\mathcal{A}$ contains all answer options and $\parallel$ denotes concatenation with a separating space. The correct-answer identity is never provided to the selector. Unlike replacing every method's scorer with cosine similarity, this experiment retains the method-specific scoring functions.

AKS and FOCUS use BLIP-L image-text matching probabilities and select 16 frames. AKS uses an approximately 1-fps candidate pool and the released recursive selection core, with thresholds $t_1 = 0.8$ and $t_2 = -100$. We set its maximum recursion depth to 4 so that the allocation $\lfloor 16/2^d \rfloor$ remains nonzero at every permitted depth d, without padding or post-selection truncation. FOCUS retains its adaptive search, with 16-second coarse sampling, 1-second fine sampling, a zoom ratio of 0.25, and the existing adaptive minimum-gap setting. Both methods retain BLIP's 35-token text limit; long question-option texts may therefore be truncated.

Q-Frame uses LongCLIP-L with 224-pixel retrieval preprocessing and unnormalized image-text dot products. It retains Gumbel-based ranking with temperature $\tau = 0.8$, at most 128 candidate frames, and a multi-resolution allocation of 4 high-, 8 medium-, and 32 low-resolution frames, totaling 44 frames.

Within each method, both conditions share the encoder, selection settings, preprocessing, downstream prompt, and greedy decoding settings. Stochastic selection uses paired per-question seeds. Selected frames are presented chronologically, and the downstream MLLM receives the original question and all answer options in both conditions. Visual-token counts match within each question pair.

For reference, we also reproduce the QV and QAV accuracies from Table 1, using LongCLIP-L and 16 selected frames. In the "Ours (QV/QAV)" row, the question-only and question-plus-options columns correspond to QV and QAV, respectively.

Table 7: Effect of answer-option conditioning on MLVU. Δ is the within-method change in percentage points computed before rounding. The reference row (†) reproduces Table 1; its Δ is the difference between the displayed accuracies.

Method	Frames	Question only (%)	Question + options (%)	Δ (pp)
AKS	16	64.95	67.57	+2.62
FOCUS	16	66.01	68.26	+2.25
Q-Frame	44	66.93	68.45	+1.52
Ours (QV/QAV)†	16	67.7	69.2	+1.5

Answer-option conditioning improves accuracy for all three methods in this setting, yielding 57, 49, and 33 additional correct answers for AKS, FOCUS, and Q-Frame, respectively. These results indicate that the benefit of answer-option information is not exclusive to QAV.

A.7 Shared-Encoder and Matched-Budget Ablation

We examine whether differences in frame-selection performance persist when the retrieval encoder and downstream visual budget are controlled. We evaluate AKS, FOCUS, Q-Frame, QV, and QAV on the same MLVU evaluation set, using LongCLIP-L with 224-pixel retrieval preprocessing as the shared encoder and Qwen2-VL-7B-Instruct as the downstream MLLM. All methods share a candidate pool sampled at 1 fps, without a method-specific candidate cap.

In the matched-budget comparison, every method selects 16 frames, which are resized to 336×336 pixels and presented chronologically. The recorded downstream visual-token count is exactly 2,304 for every question and method. The downstream text prompt, answer parser, and greedy decoding settings are identical. AKS, FOCUS, Q-Frame, and QV use only the question for frame selection; QAV uses question-option pairs. The downstream MLLM receives the question and all answer options in every condition. No selector receives the correct-answer identity.

These are controlled adaptations, not reproductions of the complete published pipelines. AKS retains its recursive selection core and per-video min-max normalization of cosine scores, with $t_1 = 0.8$, $t_2 = -100$, and maximum depth 4 for the 16-frame budget. FOCUS retains its selection core, using LongCLIP cosine scores mapped to $[0, 1]$. Q-Frame retains unnormalized feature dot products and Gumbel-based ranking with temperature $\tau = 0.8$. QV uses question-frame cosine similarity, whereas QAV uses the maximum cosine similarity across question-option pairs. QV and QAV are rerun in this shared evaluator; their results here are separate from the main benchmark runs.

We additionally evaluate Q-Frame with 44 multi-resolution frames. The first 4, next 8, and remaining 32 relevance-ranked frames use one-half, one-quarter, and one-eighth of the original image dimensions, respectively, before downstream image processing. This condition has a source-dependent visual-token budget and is reported separately from the matched 16-frame comparison.

Table 8 shows that QAV achieves the highest observed accuracy in the matched-budget comparison, improving over QV by 2.25 percentage points and FOCUS, the strongest question-only method, by 0.87 percentage points. Differences are computed before rounding. These results support the benefit of answer-option conditioning in this setting after controlling the encoder and downstream visual budget. The 44-frame Q-Frame result is a separate multi-resolution comparison rather than a matched-budget result. Neither Q-Frame row in this controlled ablation is interchangeable with the 336-pixel-retrieval Q-Frame runs in the main tables.

Table 8: Shared-encoder ablation on MLVU using LongCLIP-L and Qwen2-VL-7B-Instruct. The first five rows share the same frame and visual-token budgets. The 44-frame multi-resolution condition is reported separately. Bold indicates the best accuracy among the matched-budget rows.

Method	Frames	Visual tokens	Accuracy (%)
AKS	16	2,304	66.01
FOCUS	16	2,304	67.34
Q-Frame	16	2,304	65.46
QV	16	2,304	65.96
QAV	16	2,304	**68.22**
Q-Frame (multi-resolution)	44	Variable	66.19

A.8 Ablation Study of k

We examine how the number of selected frames affects QAV's question-answering accuracy. We evaluate frame budgets $k \in \{8, 16, 32, 64\}$ on MLVU, Video-MME, and LongVideoBench, using LongCLIP-L for frame selection and LLaVA-Video-7B as the downstream MLLM. For Video-MME and LongVideoBench, we report results both with and without subtitles.

Table 9: Question answering accuracy (%) w.r.t. different numbers of selected frames k on MLVU. LLaVA-Video-7B is used as the MLLM and LongCLIP-L is used for QAV frame selection.

k	TR	AR	NQA	ER	PQA	AO	AC	Avg.
8	76.8	51.5	82.5	67.6	77.2	48.6	36.9	66.9
16	79.1	57.0	84.5	71.0	80.1	57.1	48.1	71.3
32	**81.8**	55.5	84.5	70.2	78.9	69.1	51.5	72.8
64	**81.8**	**59.0**	**85.1**	**73.0**	**80.5**	**74.1**	**56.3**	**75.2**

Table 10: Question answering accuracy (%) w.r.t. different numbers of selected frames k on Video-MME. LLaVA-Video-7B is used as the MLLM and LongCLIP-L is used for QAV frame selection.

k	**Video-MME** (*wo / w subs*)			
	Overall	Short	Medium	Long
8	59.44 / 62.59	71.11 / 74.44	58.33 / 60.67	48.89 / 52.67
16	61.81 / 64.70	73.22 / 76.67	61.11 / 63.56	51.11 / 53.89
32	62.74 / 65.81	75.00 / 77.44	62.22 / 65.89	51.00 / 54.11
64	**63.63 / 67.22**	**75.33 / 77.89**	**63.89 / 67.44**	**51.67 / 56.33**

Table 11: Question answering accuracy (%) w.r.t. different numbers of selected frames k on LongVideoBench. LLaVA-Video-7B is used as the MLLM and LongCLIP-L is used for QAV frame selection.

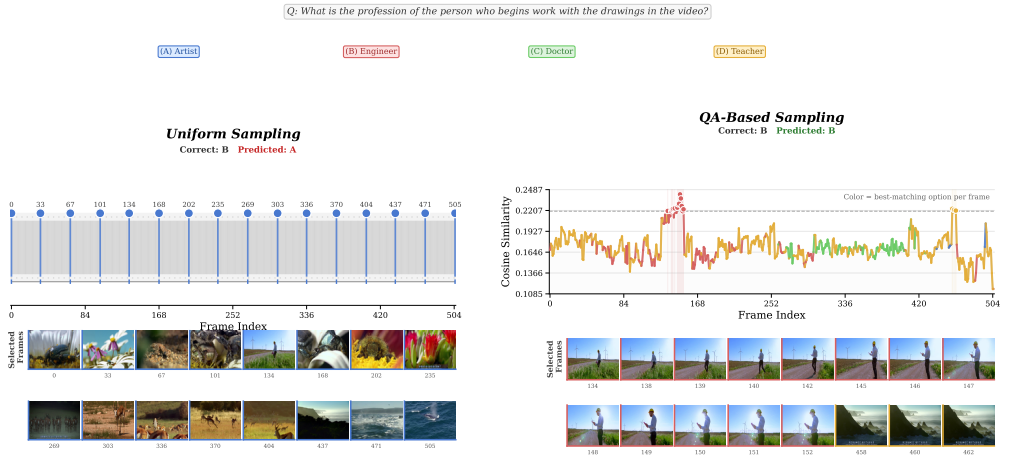
k	**LongVideoBench** (*wo / w subs*)				
	Overall	8s–15s	15s–60s	3m–10m	15m–60m
8	60.43 / 61.78	**68.78 / 69.31**	**73.84 / 76.16**	57.04 / 59.71	56.03 / 56.38
16	61.33 / 63.87	67.20 / **69.31**	72.67 / 73.26	59.95 / 64.32	56.91 / 58.87
32	**62.15 / 65.07**	67.20 / **69.31**	70.93 / 73.26	**61.41 / 65.53**	**58.33** / 60.82
64	61.11 / 64.25	67.20 / **69.31**	69.19 / 71.51	59.47 / 62.86	57.80 / **61.35**

As shown in Table 9, overall MLVU accuracy increases from 66.9% with 8 frames to 75.2% with 64 frames. However, individual task categories do not improve monotonically; for example, AR and PQA accuracy decrease when increasing the budget from 16 to 32 frames. On Video-MME (Table 10), overall accuracy increases across the tested budgets, from 59.44% to 63.63% without subtitles and from 62.59% to 67.22% with subtitles.

LongVideoBench exhibits a different pattern (Table 11). Overall accuracy peaks at $k = 32$, reaching 62.15% without subtitles and 65.07% with subtitles, before decreasing at $k = 64$. The effect of increasing the frame budget is therefore benchmark- and task-dependent: additional frames can improve accuracy, but the largest tested budget does not always provide the best performance.

A.9 Qualitative Frame-Selection Example

Figure 5 compares uniform sampling with QAV for the same video and multiple-choice question. Both methods select 16 frames. Uniform sampling selects frames at regular intervals, whereas QAV uses question-option alignment scores. In this example, the displayed prediction is incorrect for uniform sampling and correct for QAV.



(a) Uniform interval-based frame selection.

(b) Question-answer-aligned frame selection (QAV).

Figure 5: Comparison of uniform sampling and QAV under the same 16-frame budget.