PARTITION AND SELECT: EFFICIENT TRAINING-FREE KEYFRAME SELECTION FOR LONG-VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-video understanding is challenging because videos may contain tens of thousands of frames, making it impractical to process them entirely with multimodal large language models (MLLMs). Query-aware keyframe selection addresses this issue, but existing methods often require additional training or process large candidate sets with expensive encoders. We introduce **Part**ition and **S**elect (PartS), a training-free framework for efficient query-aware keyframe selection in long videos. Our method improves efficiency by reducing both the number of candidate frames that must be processed and the cost of encoding each candidate. Specifically, PartS exploits video-codec structure to construct a compact set of visual anchors and adaptively enriches it with frames from query-relevant segments. It then encodes candidates using a lightweight CLIP model, clusters them by visual similarity, and applies query-aware soft frame allocation to distribute a fixed frame budget across clusters according to their relevance. We evaluate PartS on four long-video QA benchmarks and four open-source MLLMs. It achieves 5.1–19.3$\times$ speedups over existing training-free approaches, maintaining competitive performance. These results demonstrate that PartS is an efficient plug-and-play module for long-video understanding.

1 INTRODUCTION

Recent advances in multimodal large language models (MLLMs) (Zhang et al., 2024b; Li et al., 2025a; Bai et al., 2025b; Zhu et al., 2025; Bai et al., 2025a; Qwen Team, 2026) have enabled remarkable performance on a wide range of visual understanding and reasoning tasks. However, extending these capabilities to long video remains challenging. Videos spanning minutes to hours may contain tens of thousands of frames, far exceeding the visual context that current MLLMs can process efficiently. Thus, long-video understanding requires downsampling the video before inference. A common strategy is uniform sampling, which selects equally spaced frames throughout the video, regardless of the query or the distribution of relevant frames over time.

To address this, recent works have explored query-aware keyframe selection. One line of research (Yu et al., 2025; Yao et al., 2025; Lee et al., 2026; Li et al., 2026a) has investigated learning-based approaches, where a model is trained to rank or predict the importance of frames for a given query. Although these methods improve performance over uniform sampling, they introduce additional training overhead and may not generalize well across models and benchmarks, limiting their applicability. In parallel, training-free strategies (Tang et al., 2025; Liu et al., 2025; Sun et al., 2025; Li et al., 2026b; Chen et al., 2026a) have been proposed to avoid the training overhead. However, these methods typically construct their candidate frame sets through fixed-rate uniform sampling and rely on large vision or vision-language encoder models to extract features from the sampled candidates. This significantly increases the inference cost because many frames have to be processed. FOCUS (Zhu et al., 2026) reduces the number of candidate frames through an adaptive sampling strategy, but still relies on large vision-language encoders and remains costly.

In this work, we propose PartS, a training-free framework for efficient query-aware keyframe selection in long videos. First, we introduce an adaptive candidate sampling strategy that exploits video-codec primitives to construct a compact pool of diverse, query-relevant frames, reducing the number

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.

of candidates that must be processed. To further improve efficiency, PartS employs a lightweight CLIP Radford et al. (2021) model for frame encoding, reducing the computational cost per candidate. Second, we introduce a method to select keyframes from this candidate pool. Specifically, PartS groups candidates into clusters based on visual similarity and allocates a fixed frame budget across clusters according to their relevance to the query. Finally, within each cluster, it selects the most query-relevant frames to form the final keyframe set. By jointly addressing the candidate pool size and the cost of processing each candidate, PartS significantly reduces keyframe selection overhead while maintaining competitive performance in long-video understanding tasks.

We evaluate PartS on four long-video question-answering (QA) benchmarks, LongVideoBench (Wu et al., 2024), Video-MME (Fu et al., 2025), MLVU (Zhou et al., 2025) and LVBench (Wang et al., 2025), using four open-source MLLMs: LLaVA-OneVision-7B (Li et al., 2025a), Qwen2.5-VL-7B (Bai et al., 2025b), LLaVA-Video-7B (Zhang et al., 2025b), and Qwen3-VL-7B (Bai et al., 2025a). Experimental results show that PartS substantially reduces candidate-processing cost while consistently improving over uniform sampling and remaining competitive with existing training-free keyframe selection approaches. On LVBench, PartS achieves keyframe selection speedups of 5.1–$19.3\times$ (Figure 1) over existing training-free approaches. These results demonstrate that PartS is an efficient plug-and-play module for improving long-video understanding with off-the-shelf MLLMs, without additional training.

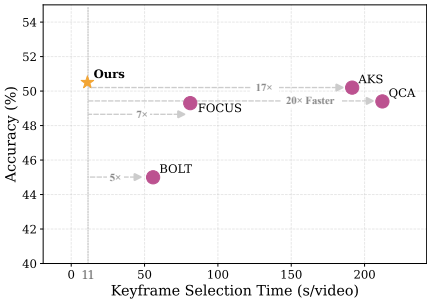


Figure 1: **Accuracy *vs.* keyframe selection time.** Qwen3-VL accuracy versus keyframe selection time (s/video) on LVBench ($K = 32$). PartS maintains competitive accuracy while requiring considerably less selection time.

Our main contributions are summarized as follows:

- We introduce **PartS, a training-free and plug-and-play framework for query-aware keyframe selection** that improves long-video understanding with off-the-shelf MLLMs without additional training.
- We propose an **efficient adaptive candidate sampling strategy** that exploits video-codec primitives to construct a compact pool of diverse, relevant frames, substantially reducing the number of candidates that need to be processed.
- We propose an **efficient frame selection pipeline** that combines a lightweight vision-language encoder with visual clustering and query-aware soft frame allocation, reducing per-frame encoding cost while balancing query relevance and visual coverage.
- We conduct extensive experiments across four long-video benchmarks and four MLLMs. We show that **PartS achieves speedups of 5.1–$19.3\times$ over existing training-free approaches** on LVBench, maintaining competitive performance.

2 Related Work

Extending MLLMs from images to long videos remains challenging: encoding all frames produces a prohibitive number of visual tokens, quickly exceeding context-length limits. A common approach is uniform sampling, used by several video MLLMs (Li et al., 2025b; Zhang et al., 2024b; Bai et al., 2025b; Zhu et al., 2025; Bai et al., 2025a), which selects equally spaced frames to reduce visual input. Although effective for short videos, this strategy often samples irrelevant frames in longer videos, degrading downstream performance. This limitation has motivated work on query-aware frame selection, which can be broadly divided into learning-based and training-free approaches.

Learning-Based Methods. These approaches address this limitation by explicitly learning to identify query-relevant frames or temporal regions. Frame-Voyager (Yu et al., 2025) ranks candidate frames by MLLM prediction loss, while K-Frames (Yao et al., 2025) localizes query-relevant clips and selects keyframes within them. ReFOCUS (Lee et al., 2026) formulates frame selection as a reinforcement learning problem, using MLLM outputs as reward signals. VideoBrain (Zou et al., 2026a) combines a CLIP-based agent to find visually relevant frames with uniform sampling to densely inspect a particular interval for actions and events. FrameOracle (Li et al., 2026a) jointly

Table 1: **Comparison of keyframe selection methods.** We compare methods in terms of training requirements, models used for frame scoring at inference time, iterative VLM calls during selection, diversity modeling, and candidate-pool construction. PartS combines training-free selection with codec-guided adaptive candidate sampling and a lightweight CLIP scorer.

Method	Training Free	Frame-Scoring Model(s)	Iterative VLM Calls	Diversity	Candidate Sampling	Codec Based
Learning-Based Methods						
Frame-Voyager (Yu et al., 2025)	✗	VILA (8B)	✗	✗	Uniform	✗
K-Frames (Yao et al., 2025)	✗	Qwen2.5-VL (3B)	✓	✗	Uniform	✗
ReFOCUS (Lee et al., 2026)	✗	Video-MA2mba	✗	✗	Uniform	✗
VideoBrain (Zou et al., 2026a)	✗	SigLIP2-B/16 (400M)	✓	✗	Uniform	✗
FrameOracle (Li et al., 2026a)	✗	DINOv2	✗	✗	Uniform	✗
Training-Free Methods						
BOLT (Liu et al., 2025)	✓	CLIP-L/14 (430M)	✗	✓	Uniform	✗
AKS (Tang et al., 2025)	✓	BLIP-ITM Large (370M)	✗	✓	Uniform	✗
MDP^3 (Sun et al., 2025)	✓	SigLIP (400M)	✗	✓	Uniform	✗
ASCS (Li et al., 2026c)	✓	CLIP-L/14 (430M)	✗	✓	Uniform	✗
Q-Frame (Zhang et al., 2025a)	✓	Long-CLIP	✗	✗	Uniform	✗
DIG (Li et al., 2026b)	✓	DINOv2 + Qwen2.5-VL (7B)	✓	✗	Uniform	✗
WFS-SB (Chen et al., 2026a)	✓	BLIP-2-ITM (1B)	✗	✓	Uniform	✗
QCA (Peng et al., 2026)	✓	BLIP-2-ITM (1B)	✗	✓	Uniform	✗
A.I.R. (Zou et al., 2026b)	✓	EVA-CLIP-L + Qwen2.5-VL (7B)	✓	✓	Uniform	✗
FOCUS (Zhu et al., 2026)	✓	BLIP-ITM Large (370M)	✗	✗	Adaptive	✗
PartS (Ours)	✓	**CLIP-B/32 (150M)**	✗	✓	**Adaptive**	✓

predicts which frames to select and how many are needed. Because these approaches require task-specific training of the frame-selection mechanism, they may not generalize well.

Training-Free Methods. In contrast to learning-based approaches, these methods seek to improve long video understanding without additional training. BOLT (Liu et al., 2025) balances query relevance and diversity through inverse-transform sampling, while AKS (Tang et al., 2025) formulates frame selection as an optimization problem over query relevance and temporal coverage. MDP^3 (Sun et al., 2025) models selection using a Markov decision determinantal point process to promote diversity, and ASCS (Li et al., 2026c) clusters visually similar frames before sampling according to cluster statistics. Q-Frame (Zhang et al., 2025a) performs keyframe selection and jointly adapts frame resolution to allocate more visual tokens to informative frames, whereas DIG (Li et al., 2026b) selects strategies based on query type. WFS-SB (Chen et al., 2026b) detects semantic boundaries from wavelet-transformed relevance signals and applies maximal marginal relevance, while QCA (Peng et al., 2026) allocates frames across temporal segments according to relevance and visual informativeness. A.I.R (Zou et al., 2026b) identifies relevant events from CLIP similarities and iteratively refines frame selection with MLLM feedback.

Despite avoiding additional training, several methods use fixed-rate uniform sampling, commonly at 1 fps, before relevance scoring. As a result, methods can spend significant computation decoding and scoring frames that are ultimately discarded. Moreover, as summarized in Table 1, several methods use large vision or vision-language backbones, such as CLIP ViT-L/14 (Radford et al., 2021) or BLIP-ITM (Li et al., 2022), to identify relevant events or frames. Thus, the computational cost of frame selection depends not only on the final number of selected frames (i.e., frame budget), but also on both the number of candidates sampled and the cost of scoring each candidate. Most related to this work, FOCUS (Zhu et al., 2026) avoids dense candidate sampling. It partitions the video into short clips and iteratively identifies promising clips that maximize expected cumulative relevance under a budget constraint. Then, it selects the most relevant frames from each clip. This reduces computation compared to uniform sampling; however, it still relies on the heavy BLIP-ITM encoder to compute relevance scores and remains costly.

In contrast, PartS targets efficiency throughout keyframe-selection by reducing both the number of candidate frames processed and the cost of encoding each candidate. We propose an adaptive candidate sampling strategy that exploits video-codec primitives to construct a compact set of diverse and relevant candidate frames, avoiding fixed-rate sampling. We also use the lightweight CLIP ViT-B/32 encoder for feature extraction, reducing per-frame encoding cost. As a result, we significantly reduce computational cost while maintaining competitive long-video QA performance.

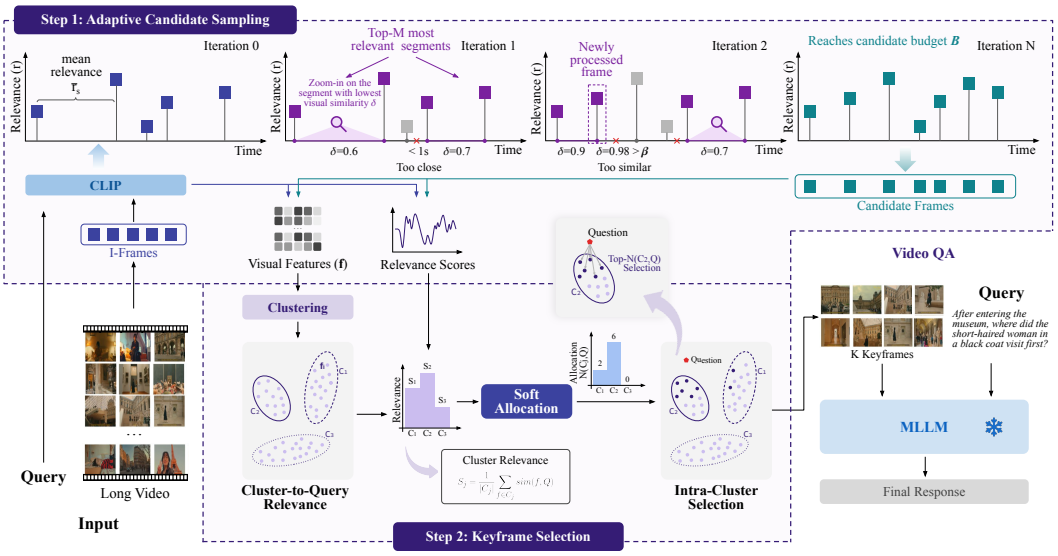


Figure 2: **PartS pipeline.** Given a video and a query, *Step 1* constructs a candidate frame pool by iteratively exploring query-relevant segments and diverse frames. In *Step 2*, it clusters the candidate frames and applies query-aware soft allocation to distribute the frame budget across clusters. Finally, it selects the top-$N(C_j)$ query-relevant frames from each cluster C_j and passes them to the MLLM.

3 Methodology

3.1 Problem Formulation

Let a video $\mathcal{V} = \{f_t\}_{t=1}^{T}$ consist of T frames, and let Q denote a textual query associated with it. Video question answering aims to generate an answer conditioned on both $\mathcal{V}$ and Q. For long videos, however, processing all T frames with an MLLM is computationally infeasible. A standard baseline is uniform sampling, which selects K equally spaced frames independently of Q. This often misses moments important to answering Q, an issue that becomes more evident as video duration increases.

Given a frame budget $K \ll T$, the goal of query-aware keyframe selection is to identify a subset $\mathcal{F} \subseteq \mathcal{V}, |\mathcal{F}| = K$, that grounds the answer to Q. A naive strategy is to select the top-K frames with the highest query relevance. However, relevant content is often concentrated within a small temporal region, then top-K selection may yield redundant frames. Hence, in this work, we formulate keyframe selection as an optimization problem that jointly maximizes *query-relevance* and *diversity*, which discourages redundant frames. Specifically, we aim to maximize the utility function:

$$\mathcal{F}^{\star} = \underset{\mathcal{F}\subseteq\mathcal{V},|\mathcal{F}|=K}{\arg\max} \underbrace{\alpha \cdot r(\mathcal{F}, Q)}_{\text{query-relevance}} + \underbrace{(1-\alpha) \cdot d(\mathcal{F})}_{\text{diversity}}, \tag{1}$$

where $r(\mathcal{F}, Q)$ denotes the relevance of the selected frames to the query, $d(\mathcal{F})$ measures their diversity, and $\alpha \in [0, 1]$ controls the trade-off between the two objectives.

Directly solving Equation 1 is intractable, as it requires combinatorial search over all subsets of $\mathcal{V}$. Instead, we propose a two-stage approach that efficiently approximates this objective. As illustrated in Figure 2, in *Step 1*, we construct a pool of promising candidate frames. Specifically, we exploit query relevance to identify promising temporal regions and then explore diversity within those regions. In *Step 2*, we select keyframes from the candidate pool. To do so, we cluster the candidate frames to group visually similar frames and apply a query-aware soft allocation mechanism that distributes the frame budget K across clusters according to their relevance to the query. Then, we perform an intra-cluster selection, selecting the most relevant frames of each cluster according to its allocated budget. The final keyframe set is passed to the MLLM.

Throughout the following sections, we denote the maximum candidate-frame budget by B, such that $K \ll B \ll T$. Thus, we first reduce the original video $\mathcal{V}$ to a small candidate set $\mathcal{V}' \subseteq \mathcal{V}$, with $|\mathcal{V}'| \leq B$, and then we select the final set $\mathcal{F}$ of K keyframes.

3.2 ADAPTIVE CANDIDATE SAMPLING

Instead of densely decoding frames through uniform sampling, we propose an exploitation-then-exploration strategy to construct a set of promising candidates that balances the query-relevance and diversity terms in Equation 1. Let $\mathbf{f}_i = \phi(f_i) \in \mathbb{R}^d$ and $\mathbf{q} = \phi(Q) \in \mathbb{R}^d$ denote the ℓ_2-normalized visual and textual embeddings of frame f_i and query Q, respectively. We define the query-relevance of frame f_i as $r_i = \mathbf{f}_i^\top \mathbf{q}$ and the visual similarity between frames f_i and f_j as $\delta_{i,j} = \mathbf{f}_i^\top \mathbf{f}_j$. Given a video $\mathcal{V}$, we initialize the candidate set $\mathcal{V}'$ with its I-frames. Let $\mathcal{I} = \{[f_{i_1}, f_{i_2}]\}$ denote a collection of temporal segments delimited by consecutive I-frames[1]. For a segment $s = [f_{i_1}, f_{i_2}]$, we define its mean relevance as $\bar{r}_s = (r_{i_1} + r_{i_2})/2$ and its endpoint visual similarity as $\delta_s = \delta_{i_1,i_2}$.

At each iteration, we perform *exploitation* by selecting the top-M segments in $\mathcal{I}$ according to $\bar{r}$, thereby restricting refinement to regions that are most likely to contain query-relevant information. We then perform *exploration* within this high-relevance subset by selecting the segment $s_j = [f_{j_1}, f_{j_2}]$ with the lowest endpoint visual similarity δ_{s_j}. This favors regions whose endpoints exhibit larger visual variation, encouraging frame diversity while avoiding refinement of visually redundant segments. For the selected segment, we decode its midpoint $f'_j = f_{\lfloor (j_1+j_2)/2 \rfloor}$ and add it to $\mathcal{V}'$. The selected segment s_j is then split into $s_{\text{left}} = [f_{j_1}, f'_j]$ and $s_{\text{right}} = [f'_j, f_{j_2}]$. Each child segment is reinserted into $\mathcal{I}$ only if its duration is at least one second and its endpoint similarity does not exceed a threshold β, i.e., $\delta_{s_{\text{left}}} \leq \beta$ and $\delta_{s_{\text{right}}} \leq \beta$, where β controls the maximum tolerated visual redundancy between endpoints. The algorithm continues until the candidate budget B is reached or no valid segment remains in $\mathcal{I}$.

3.3 KEYFRAME SELECTION

Frame Clustering. After constructing the candidate set $\mathcal{V}'$, we encode frames $f_i \in \mathcal{V}'$ using CLIP (Radford et al., 2021), obtaining visual embeddings $\mathbf{f}_i = \phi(f_i) \in \mathbb{R}^d$. We then employ K-Means to partition frame embeddings $\mathbf{f}_i$ into c clusters $\mathcal{C} = \{C_1, \ldots, C_c\}$. Frames within the same cluster are expected to contain similar visual content. Thus, distributing the frame budget across clusters discourages the final selection from collapsing onto redundant frames.

Cluster-to-Query Relevance. We compute cluster relevance based on frame-to-query relevance. We compute the relevance of each frame using CLIP similarity and aggregate the resulting scores within each cluster to obtain its *cluster-to-query relevance*. Given a cluster C_j and a query Q, we define:

$$S_j \triangleq \frac{1}{|C_j|} \sum_{f \in C_j} sim(f, Q), \quad sim(f, Q) = \mathbf{f}^\top \mathbf{q}, \tag{2}$$

where $|C_j|$ denotes the cardinality of C_j. The score S_j summarizes the overall relevance of cluster C_j to the query.

Soft Frame Allocation. We propose a soft allocation mechanism that distributes the total frame budget K across clusters proportionally to their relevance to the query. Formally, given a cluster C_j and query Q, the allocated budget is:

$$N(C_j, Q) = \lfloor K \cdot \frac{exp(S_j/\tau)}{\sum_{i=1}^{c} exp(S_i/\tau)} \rfloor, \tag{3}$$

where τ is a temperature parameter that controls the concentration of the allocation distribution. As τ decreases, the allocation increasingly favors highly relevant clusters; as it increases, the distribution approaches uniform allocation, increasing cluster coverage.

Intra-Cluster Frame Selection. For each cluster C_j, we rank its frames by relevance, $sim(f, Q)$ (Equation 2), and select the top-$N(C_j, Q)$ frames. The final keyframe set $\mathcal{F}$ is the union of the selections across all clusters.

$$\mathcal{F} = \bigcup_{C_j \in \mathcal{C}} \underset{f \in C_j}{\text{Top-N}(\text{C}_\text{j}, \text{Q})} \; sim(f, Q). \tag{4}$$

Once the keyframes are selected, we feed them to the MLLM in chronological order.

[1] A compressed video is typically composed of intra-coded (I), predictive (P), and bidirectionally predictive (B) frames. I-frames are encoded independently and commonly serve as reference points for subsequent inter-coded frames (Wiegand et al., 2003), providing a sparse set of temporally distributed frames that can be accessed without densely decoding the entire video. We exploit them to construct an initial candidate pool.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate our method on four long video QA benchmarks: LongVideoBench (Wu et al., 2024), Video-MME (Fu et al., 2025), MLVU (Zhou et al., 2025), and LVBench (Wang et al., 2025). These benchmarks cover videos from a few minutes to hours, with LVBench containing videos longer than 30 minutes. Following common practice, we assess keyframe selection indirectly through downstream MLLM QA accuracy. To complement this, we also evaluate keyframe selection quality directly using query-relevant frame annotations from LVBench and Video-MME-kfs (Li et al., 2026c). We report Key Frame Rate (KFR), which measures the precision of the selected frames against annotated keyframes, and Scene Hit Rate (SHR), which measures coverage of query-relevant scenes. Together, these metrics indicate whether the selected frames capture the frames needed to answer a question. To isolate visual understanding, we do not use subtitles in our experiments.

Implementation Details. We use CLIP ViT-B/32 (Radford et al., 2021) to compute frame-to-query relevance and inter-frame visual similarity, and apply K-Means clustering to partition candidate frames. To demonstrate the generalization of PartS, we use four open-source MLLMs: LLaVA-OneVision-7B (Li et al., 2025a), Qwen2.5-VL-7B (Bai et al., 2025b), LLaVA-Video-7B (Zhang et al., 2025b), and Qwen3-VL-8B (Bai et al., 2025a). For candidate sampling, we set the maximum candidate-frame budget B to 0.5 fps, the maximum visual similarity to $\beta = 0.95$, and pool size $M = 8$. Unless otherwise specified, we use a frame budget of $K = 32$, an allocation temperature of $\tau = 2.0$, and $c = 100$ clusters. Further details are provided in Appendix A. We run all experiments on a single node with two NVIDIA L40S GPUs (48GB each). We will release our code upon acceptance.

4.2 COMPARISON WITH STATE OF THE ART

Table 2 compares PartS against recent state-of-the-art methods across four long video QA benchmarks and four MLLMs. We report prior results obtained with the same MLLMs and benchmarks considered in our evaluation whenever available. However, differences in evaluation settings, such as the number of input frames and prompts, may affect performance and therefore limit direct comparison. For a fair comparison, we reproduce three training-free baselines, BOLT, AKS, and FOCUS, under the same experimental setting as ours.

PartS consistently improves over uniform sampling, increasing the average accuracy by 4.1 for models from the LLaVA family and roughly 6.0 for those from the Qwen-VL family. Moreover, PartS achieves the best average performance across benchmarks among the reproduced training-free baselines for all four MLLMs. The gains are particularly pronounced for Qwen2.5-VL and Qwen3-VL, where PartS surpasses the strongest reproduced baseline by 0.9 points on average. Compared with FOCUS, the most closely related method, PartS consistently improves performance across all evaluated MLLMs and benchmarks, with average gains ranging from 0.9 to 2.0 points depending on the MLLM. PartS achieves the best performance or remains competitive with methods that rely on dense 1 fps uniform sampling across all evaluated MLLMs and benchmarks, despite substantially reducing the selection cost (Section 4.3). These results demonstrate that processing a densely sampled candidate pool is not necessary to achieve strong long video QA performance.

4.3 EFFICIENCY ANALYSIS

We analyze the computational efficiency of keyframe selection on LVBench using Qwen3-VL as the downstream MLLM, with a budget of $K = 32$ keyframes for all methods. We report the FLOPs required to encode candidate frames and compute their query-relevance scores, as well as the average latency per video in seconds decomposed into candidate encoding, frame selection, and Qwen3-VL inference. Figure 4 shows that PartS reduces the computational cost of keyframe selection by 62–216$\times$ compared with training-free baselines while maintaining competitive QA accuracy. This reduction in computational cost also translates into lower inference latency. Table 3 shows that candidate encoding dominates runtime, whereas the selection step adds negligible overhead. Since all methods pass the same 32 frames to Qwen3-VL, downstream inference remains fixed at 2.4 s/video. Thus, by reducing candidate-processing cost, PartS lowers keyframe-selection latency to 11.0 s/video, corresponding to 5.1–19.3$\times$ speedups, and reduces end-to-end latency from

Table 2: **Comparison with state-of-the-art keyframe selection methods.** We report overall accuracy (%), considering only training-free methods, as indicated by the **TF** column. † denotes baselines reproduced under our evaluation setup. Gray entries are results reported in the original papers and are not directly used for ranking. Among methods evaluated under our setup, the best and second-best results are shown in **bold** and underlined, respectively. Values in green indicate the average improvement of PartS over uniform sampling using the same MLLM and frame budget.

Method	Candidate Pool	# Frames	TF	LongVideoBench	Video-MME	MLVU	LVBench	Avg.
LLaVA-OV-7B								
MDP^3 (Sun et al., 2025)	Uniform / 1 fps	8	✓	59.0	59.1	69.8	–	–
A.I.R. (Zou et al., 2026b)	Uniform / 1 fps	≤ 32	✓	60.7	61.4	69.3	–	–
WFS-SB (Chen et al., 2026a)	Uniform / 1 fps	8	✓	59.8	59.3	67.2	–	–
Uniform Sampling	Uniform / 32	32	✓	56.2	58.5	63.3	39.6	54.4
BOLT† (Liu et al., 2025)	Uniform / 1 fps	32	✓	57.6	59.6	**66.9**	44.0	57.0
AKS† (Tang et al., 2025)	Uniform / 1 fps	32	✓	**60.7**	60.8	65.9	46.4	**58.5**
QCA† (Peng et al., 2026)	Uniform / 1 fps	32	✓	60.3	59.7	66.4	**46.6**	58.3
FOCUS† (Zhu et al., 2026)	Adaptive	32	✓	60.2	58.4	65.9	45.8	57.6
PartS (ours)	Adaptive	32	✓	60.4	**61.1**	66.6	45.9	**58.5** (+4.1)
Qwen2.5-VL-7B								
DIG (Li et al., 2026b)	Uniform / 2 fps	32	✓	60.4	–	67.2	–	–
A.I.R. (Zou et al., 2026b)	Uniform / 1 fps	≤ 32	✓	61.4	65.0	67.5	–	–
ASCS (Li et al., 2026c)	Uniform / 1 fps	32	✓	63.3	63.1	–	–	–
WFS-SB (Chen et al., 2026a)	Uniform / 1 fps	32	✓	64.4	64.4	70.4	–	–
Uniform Sampling	Uniform / 32	32	✓	58.6	62.2	59.4	39.1	54.8
BOLT† (Liu et al., 2025)	Uniform / 1 fps	32	✓	61.8	**63.4**	67.8	43.2	59.1
AKS† (Tang et al., 2025)	Uniform / 1 fps	32	✓	62.4	62.7	65.6	**48.5**	59.8
QCA† (Peng et al., 2026)	Uniform / 1 fps	32	✓	62.0	62.1	68.3	47.3	59.9
FOCUS† (Zhu et al., 2026)	Adaptive	32	✓	61.7	62.0	66.2	45.5	58.9
PartS (ours)	Adaptive	32	✓	**63.9**	63.3	**68.6**	47.4	**60.8** (+6.0)
LLaVA-Video-7B								
WFS-SB (Chen et al., 2026a)	Uniform / 1 fps	8	✓	61.1	61.7	66.9	–	–
Uniform Sampling	Uniform / 32	32	✓	58.7	62.8	66.0	40.5	57.0
BOLT† (Liu et al., 2025)	Uniform / 1 fps	32	✓	60.9	**64.3**	**70.0**	43.6	59.7
AKS† (Tang et al., 2025)	Uniform / 1 fps	32	✓	61.3	64.1	68.7	45.3	59.9
QCA† (Peng et al., 2026)	Uniform / 1 fps	32	✓	61.8	63.8	68.1	47.9	60.4
FOCUS† (Zhu et al., 2026)	Adaptive	32	✓	61.0	61.7	67.1	47.1	59.2
PartS (ours)	Adaptive	32	✓	**62.1**	64.1	69.5	**48.7**	**61.1** (+4.1)
Qwen3-VL-8B								
Uniform Sampling	Uniform / 32	32	✓	62.1	64.8	61.7	41.3	57.5
BOLT† (Liu et al., 2025)	Uniform / 1 fps	32	✓	62.6	65.8	**70.4**	45.0	61.0
AKS† (Tang et al., 2025)	Uniform / 1 fps	32	✓	63.0	65.9	67.8	50.2	61.7
QCA† (Peng et al., 2026)	Uniform / 1 fps	32	✓	64.3	65.3	69.4	49.4	62.1
FOCUS† (Zhu et al., 2026)	Adaptive	32	✓	63.5	63.3	67.8	49.3	61.0
PartS (ours)	Adaptive	32	✓	**66.0**	**66.0**	69.4	**50.5**	**63.0** (+5.5)

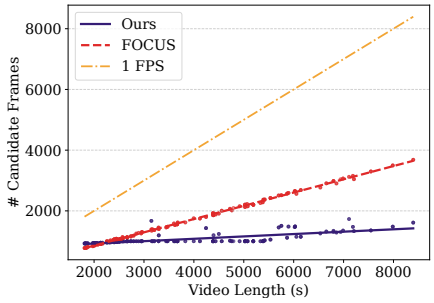


Figure 3: **Candidate-frame scaling.** Number of candidate frames selected as video duration increases on LVBench. PartS scales more slowly than FOCUS and 1-FPS sampling.

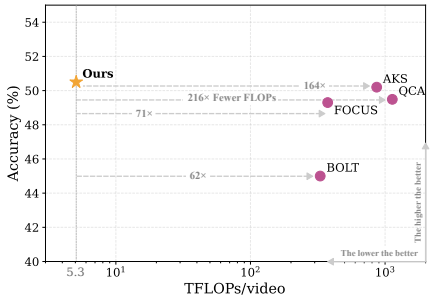


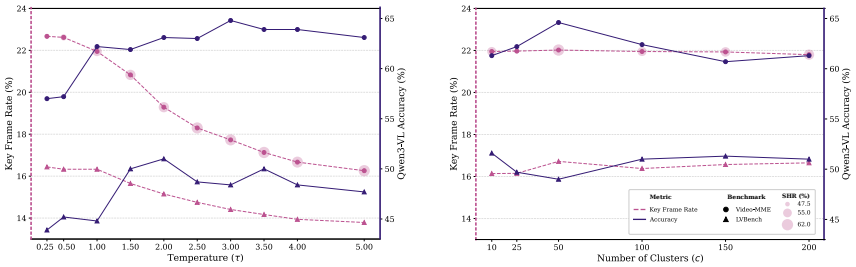
Figure 4: **Accuracy *vs.* Keyframe selection FLOPs.** Qwen3-VL accuracy versus computational cost of keyframe selection (TFLOPs/video) on LVBench ($K = 32$).

58.2–214.5 s/video to 13.4 s/video. To complement this analysis, we further report energy consumption and carbon emissions in Appendix B.

The efficiency gains arise from two factors: the number of candidate frames processed and the cost of encoding each candidate. First, PartS avoids uniformly sampling frames over the entire video to construct a candidate pool. Figure 3 shows that the number of candidate frames our method processes grows more slowly with video duration than with both 1-FPS and FOCUS sampling

Table 3: **Inference latency analysis on LVBench.** Average per-video latency (s/video) for candidate encoding, frame selection, and downstream Qwen3-VL inference with a fixed budget of $K = 32$ frames. KF Latency denotes the total keyframe selection latency. Speedup indicates how much faster PartS is relative to each baseline in terms of KF latency.

Method	Keyframe Selection				MLLM Inference	Total Latency
	Candidate Encoding	Selection	KF Latency (↓)	Speedup (↓)		
QCA	211.7	0.4	212.1	19.3 ×	2.4	214.5
AKS	191.1	0.4	191.5	17.4×	2.4	193.9
FOCUS	80.9	0.2	81.2	7.3×	2.4	83.5
BOLT	55.7	**0.1**	55.8	5.1×	2.4	58.2
PartS (ours)	**10.4**	0.6	**11.0**	1.0×	2.4	**13.4**



(a) Allocation temperature (b) Number of clusters

Figure 5: Impact of (a) allocation temperature τ, and (b) number of clusters c on keyframe selection and Qwen3-VL-8B performance on Video-MME and LVBench.

strategies. Second, PartS computes frame-query relevance using the lightweight CLIP ViT-B/32 encoder, avoiding the cost of repeatedly invoking larger vision-language encoders during candidate scoring. These gains make PartS particularly suitable for long-video understanding, where large candidate pools and expensive scoring models can become computational bottlenecks.

4.4 Ablations

Hyperparameter Sensitivity. We analyze the sensitivity of PartS to the allocation temperature τ and the number of clusters c in Figure 5, using 20% subsets of Video-MME and LVBench. We report Qwen3-VL QA accuracy and keyframe rate (KFR) on both benchmarks, as well as scene hit rate (SHR) on Video-MME, which provides scene annotations.

The temperature τ controls the trade-off between concentrating the frame budget on highly relevant clusters and distributing it more evenly across the video. As τ increases, KFR consistently decreases, while SHR increases, highlighting the relevance-coverage trade-off. As expected, downstream QA accuracy is higher at intermediate values of τ, as small values concentrate the budget on a few highly relevant regions, whereas large values sacrifice relevance for broader coverage. In particular, $\tau = 2.0$ provides a good trade-off among KFR, SHR, and downstream QA accuracy across both benchmarks, and we therefore use it as the default setting.

In contrast, PartS is less sensitive to the number of clusters c. Both KFR and SHR remain relatively stable over a range of c, while downstream accuracy exhibits marginal variation. These results show that allocation temperature matters more than clustering granularity, while PartS remains robust across a broad range of cluster configurations. We note that the same trends hold on both Video-MME and LVBench.

Robustness to Frame Budget. We further evaluate the robustness of PartS to different frame budgets $K \in \{16, 32, 64\}$, while keeping all other hyperparameters fixed. We report results in Table 4. PartS consistently outperforms uniform sampling across every frame budget, MLLM, and benchmark, demonstrating that its effectiveness is not tied to a particular choice of K. The gains are especially pronounced on LVBench, where PartS improves over uniform sampling by 3.9 to 9.2 points across the evaluated settings. Notably, with only $K = 16$ frames, PartS already surpasses uniform sampling with $K = 64$ for all four MLLMs on LVBench, indicating that selecting informative frames can be more effective than simply increasing the frame budget.

Table 4: **Robustness to frame budget**. Accuracy (%) for $K \in \{16, 32, 64\}$ using identical hyperparameters across budgets. Our method outperforms uniform sampling across all budgets.

		Video-MME			LVBench		
Model	**Method**	K=16	K=32	K=64	K=16	K=32	K=64
LLaVA-OV-7B	Uniform	57.6	58.5	59.1	36.5	39.6	40.9
	PartS (Ours)	**60.0**	**61.1**	**60.9**	**43.4**	**45.9**	**46.4**
LLaVA-Video-7B	Uniform	60.1	62.8	63.7	40.2	40.5	43.4
	PartS (Ours)	**62.0**	**64.1**	**65.6**	**44.1**	**48.7**	**49.3**
Qwen2.5-VL-7B	Uniform	57.7	62.2	64.2	36.0	39.1	41.4
	PartS (Ours)	**61.4**	**63.3**	**64.8**	**44.9**	**47.4**	**49.8**
Qwen3-VL-8B	Uniform	61.4	64.8	67.6	38.7	41.3	44.3
	PartS (Ours)	**63.0**	**66.0**	**68.4**	**46.1**	**50.5**	**52.4**

Table 5: **Performance across video durations on Video-MME.** We report accuracy (%) on short, medium, and long videos. Within each MLLM, the best and second-best results are shown in **bold** and underlined, respectively.

Method	**Short** (≤2 min)	**Medium** (4–15 min)	**Long** (30–60 min)	**Overall**	**Method**	**Short** (≤2 min)	**Medium** (4–15 min)	**Long** (30–60 min)	**Overall**
LLaVA-OV-7B					**Qwen2.5-VL-7B**				
Uniform	70.7	56.1	48.8	58.5	Uniform	73.0	60.9	52.8	62.2
BOLT	70.1	59.8	48.9	59.6	BOLT	73.2	61.8	55.1	**63.4**
AKS	**71.6**	58.9	**52.0**	60.8	AKS	73.1	61.3	53.8	62.7
QCA	70.8	58.4	49.8	59.7	QCA	**73.9**	61.5	50.9	62.1
FOCUS	68.4	57.2	49.6	58.4	FOCUS	71.6	61.7	52.9	62.0
PartS (ours)	71.3	**60.6**	51.4	**61.1**	**PartS** (ours)	72.7	**62.1**	**55.2**	63.3
LLaVA-Video-7B					**Qwen3-VL-8B**				
Uniform	75.1	61.6	51.7	62.8	Uniform	76.4	63.3	54.7	64.8
BOLT	**75.7**	**63.1**	**54.2**	**64.3**	BOLT	75.9	64.4	**57.0**	65.8
AKS	**75.7**	63.0	53.6	64.1	AKS	**78.1**	64.2	55.2	65.9
QCA	74.8	62.7	54.0	63.8	QCA	77.1	62.1	56.7	65.3
FOCUS	71.4	60.6	53.2	61.7	FOCUS	74.0	60.9	55.1	63.3
PartS (ours)	75.2	62.9	**54.2**	64.1	**PartS** (ours)	76.0	**65.2**	56.9	**66.0**

4.5 Performance Across Video Lengths

We analyze performance across video durations on Video-MME (Table 5). The gain of PartS over uniform sampling increases as the length of the video increases. Across the four MLLMs, PartS performs on par with uniform sampling on short videos and improves accuracy by 2.2 and 2.4 points on average for medium and long videos, respectively. Compared with FOCUS, the most closely related adaptive sampling method, PartS consistently achieves higher accuracy across all duration splits and MLLMs. Moreover, PartS remains competitive with or outperforms the strongest reproduced baselines, particularly on medium and long videos. Together with the efficiency results in Section 4.3, these findings show that PartS becomes increasingly advantageous for longer videos, reducing selection cost while preserving or improving downstream QA performance. We further analyze performance across Video-MME task categories in Appendix C.

5 Conclusion

We introduced PartS, a training-free, plug-and-play framework for efficient query-aware keyframe selection in long videos. PartS improves efficiency by reducing both the number of candidate frames through adaptive sampling and the cost of processing each candidate through lightweight CLIP-based scoring. Combined with our frame selection strategy, this significantly reduces keyframe selection cost while consistently outperforming uniform sampling and remaining competitive with existing methods across four long-video QA benchmarks and four open-source MLLMs. On LVBench, PartS achieves 5.1–19.3× faster keyframe selection than competing training-free methods. These results establish PartS as an efficient module for improving long-video understanding with off-the-shelf MLLMs, without additional training. Future work will explore adaptive frame budgets, richer temporal cues, and extensions to other long-video understanding tasks, including video captioning.

AI USAGE STATEMENT

We used GPT-5 solely for language assistance during the preparation of this paper, including improving grammar and clarity and polishing the author's original content. The model was not used to generate scientific ideas, to design the method, to conduct experiments, to analyze results, or to draw conclusions. All technical content, claims, and interpretations were developed and verified by the authors, who take full responsibility for the final manuscript.

ETHICS STATEMENT

This work evaluates publicly available MLLMs solely on publicly available video QA benchmarks. We fully comply with the terms of use and licensing agreements associated with each model and benchmark. We do not conduct any fine-tuning or modifications to the models that could introduce unintended risks. However, we recognize that our work reflects the existing limitations and potential risks of the evaluated models. In addition, our proposed method is designed to improve the efficiency of long video understanding, potentially reducing the cost of analyzing large volumes of video. As such, it may be misused to enable or scale systems and could amplify harms such as privacy violations or disproportionate impacts on marginalized communities.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Wang Chen, Yuhui Zeng, Yongdong Luo, Tianyu Xie, Luojun Lin, Jiayi Ji, Yan Zhang, and Xiawu Zheng. Wavelet-based frame selection by detecting semantic boundary for long video understanding. In *CVPR*, pp. 24052–24061, June 2026a.

Wang Chen, Yuhui Zeng, Yongdong Luo, Tianyu Xie, Luojun Lin, Jiayi Ji, Yan Zhang, and Xiawu Zheng. Wavelet-based frame selection by detecting semantic boundary for long video understanding. In *CVPR*, pp. 24052–24061, June 2026b.

Benoit Courty, Victor Schmidt, Goyal-Kamal, Marion Coutarel, Boris Feld, Jérémy Lecourt, LiamConnell, SabAmine, inimaz, supatomic, Mathilde Léval, Luis Blanche, Alexis Cruveiller, ouminasara, Franklin Zhao, Aditya Joshi, Alexis Bogroff, Amine Saboni, Hugues de Lavoreille, Niko Laskaris, Edoardo Abati, Douglas Blank, Ziyao Wang, Armin Catovic, alencon, Michal Stechly, Christian Bauer, Lucas-Otavio, JPW, and MinervaBooks. mlco2/codecarbon: v2.4.1, May 2024. URL https://doi.org/10.5281/zenodo.11171501.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *CVPR*, pp. 24108–24118, 2025.

Hosu Lee, Junho Kim, Hyunjun Kim, and Yong Man Ro. Refocus: Reinforcement-guided frame optimization for contextual understanding. In *CVPR*, pp. 8291–8302, 2026.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *TMLR*, 2025a.

Chaoyu Li, Tianzhi Li, Fei Tao, ZHENYU ZHAO, Ziqian Wu, Maozheng Zhao, Juntong Song, Cheng Niu, and Pooyan Fazli. Frameoracle: Learning what to see and how much to see in videos. In *ICML*, 2026a. URL https://openreview.net/forum?id=Vd5bjrFQBh.

Jialuo Li, Bin Li, Jiahao Li, and Yan Lu. Divide, then ground: Adapting frame selection to query types for long-form video understanding. In *CVPR*, pp. 11369–11380, 2026b.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, pp. 12888–12900. PMLR, 2022.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *Science China Information Sciences*, 68, 2025b.

Zongyao Li, Kengo Ishida, Satoshi Yamazaki, Xiaotong Ji, and Jianquan Liu. Kfs-bench: Comprehensive evaluation of key frame sampling in long video understanding. In *WACV*, pp. 5643–5652, 2026c.

Shuming Liu, Chen Zhao, Tianqi Xu, and Bernard Ghanem. Bolt: Boost large vision-language model without training for long-form video understanding. In *CVPR*, pp. 3318–3327, 2025.

Jun Peng, Baiyang Song, Jie Li, Hui Li, Yiyi Zhou, Rongrong Ji, and Yonghong Tian. Qca: Query-and content-aware keyframe selection for long video understanding. *arXiv preprint arXiv:2607.00983*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pp. 8748–8763, 2021.

Hui Sun, Shiyin Lu, Huanyu Wang, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Ming Li. MDP3: A training-free approach for list-wise frame selection in video-llms. In *ICCV*, pp. 24090–24101, 2025.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *CVPR*, pp. 29118–29128, 2025.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *ICCV*, pp. 22958–22967, 2025.

Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7): 560–576, 2003.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. In *NeurIPS*, pp. 28828–28857, 2024.

Yifeng Yao, Yike Yun, Jing Wang, Huishuai Zhang, Dongyan Zhao, Ke Tian, Zhihao Wang, Minghui Qiu, and Tao Wang. K-frames: Scene-driven any-k keyframe selection for long video understanding. *arXiv preprint arXiv:2510.13891*, 2025.

Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xu Xiaolei, Zhenbang Sun, Bingni Zhang, Jiawei Wu, et al. Frame-voyager: Learning to query frames for video large language models. In *ICLR*, 2025.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Reality check on the evaluation of large multimodal models, 2024a. URL `https://arxiv.org/abs/2407.12772`.

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 *ICCV*, pp. 22056–22065, October 2025a.

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

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun MA, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *TMLR*, 2025b.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *CVPR*, pp. 13691–13701, 2025.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

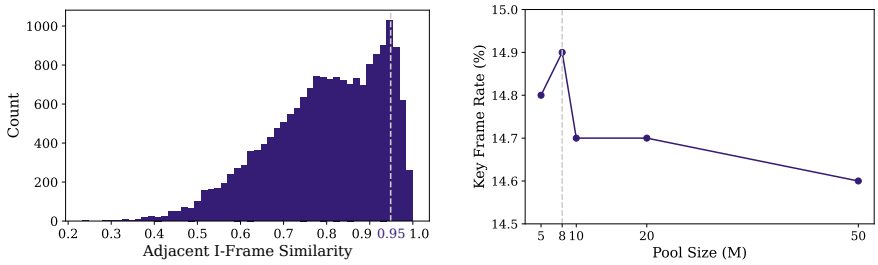
Zirui Zhu, Hailun Xu, Yang Luo, Yong Liu, Kanchan Sarkar, Zhenheng Yang, and Yang You. Focus: Efficient keyframe selection for long video understanding. In *ICLR*, 2026.

Junbo Zou, Ziheng Huang, Shengjie Zhang, Liwen Zhang, and Weining Shen. Videobrain: Learning adaptive frame sampling for long video understanding. In *ICML*, 2026a. URL https://openreview.net/forum?id=1TXn0PeBZJ.

Yuanhao Zou, Shengji Jin, Andong Deng, Youpeng Zhao, Jun Wang, and Chen Chen. A.i.r.: Enabling adaptive, iterative, and reasoning-based frame selection for video question answering. In *ICLR*, 2026b. URL https://openreview.net/forum?id=SZVpOKw0YD.

A Additional Experimental Details

Adaptive Candidate Sampling Hyperparameters. We report in Figure A.1 the analyses used to determine the hyperparameters for candidate-pool construction. To assess the quality of our candidate sampling strategy, we leverage the keyframe annotations provided by LVBench and report the keyframe rate (KFR). As shown in Figure A.1a, adjacent I-frames exhibit high visual similarity, reflecting the strong temporal redundancy in videos. We therefore set the visual similarity tolerance to $\beta = 0.95$, which removes most highly redundant frames. We further analyze the effect of the candidate pool size M in Figure A.1b. KFR varies only marginally across different values of M, indicating that the candidate sampling strategy is relatively insensitive to this hyperparameter. Based on this observation, we set $M = 8$ as the default throughout our experiments.



(a) Visual similarity tolerance (β) (b) Pool Size (M)

Figure A.1: **Candidate-pool analysis.** (a) Distribution of visual similarity between adjacent frames from LVBench, used to guide the choice of the similarity tolerance β. (b) Effect of the candidate pool size M on the keyframe rate (KFR) on LVBench.

Frame-to-Query Relevance. Since CLIP's text context length is limited to 77, encoding the whole query could easily exceed this limit. We address this issue by splitting the query into a question and candidate answers. Formally, given a query Q composed of a question and (when available) a set of candidate answers, we encode the question and the set of candidate answers using CLIP, obtaining text embeddings $\mathbf{q}_0$ for the question and $\{\mathbf{q}_j\}_{j=1}^{m}$ for m candidates. We define frame-to-query similarity as the maximum cosine similarity between frame embedding $\mathbf{f}_i$ and the set of text embeddings $\{\mathbf{q}_j\}_{j=0}^{m}$:

$$sim(f_i, Q) = \max_{j \in \{0,\cdots,m\}} \frac{\mathbf{f}_i^\top \mathbf{q}_j}{\|\mathbf{f}_i\| \, \|\mathbf{q}_j\|}. \tag{5}$$

It is worth noting that candidate options $\{q_j\}_{j\geq 1}$ are part of the standard prompt provided to the MLLM at inference time for multiple-choice video QA. Thus, our method does not rely on any privileged information beyond what is available to the model at inference time. Moreover, the maximum aggregation function assigns the same weight to all candidates; therefore, it does not condition the relevance score on the ground-truth answer. In open-ended QA settings, where candidate options are not available, the formulation of frame-to-query similarity reduces to frame-question similarity by setting $m = 0$.

B Energy Consumption and Carbon Footprint Analysis

We analyze the energy consumption and carbon emissions on LVBench using CodeCarbon (Courty et al., 2024). As shown in Table A.1 shows that our method is more energy-efficient than the competing keyframe selection approaches. It consumes only 17.5 kWh, compared with 45.1 kWh to 398.0 kWh for baselines. Similarly, PartS produces only 1.7 kg of CO_2e, compared with 4.4 to 39.2 kg for baselines, representing a 2.6×–23.1× reduction in carbon emissions. These results further demonstrate the computational efficiency of our keyframe selection strategy.

Table A.1: **Energy consumption and carbon emissions of keyframe selection methods.** Our method substantially reduces both energy consumption and CO_2e emissions compared with existing training-free approaches.

Method	CO_2e (kg) ↓	Energy (kWh) ↓
QCA	39.2	398.0
AKS	32.7	333.4
FOCUS	14.2	144.1
BOLT	4.4	45.1
PartS (Ours)	**1.7**	**17.5**

C Performance across tasks.

We compare methods across Video-MME task categories using Qwen3-VL and plot results in Figure A.2. PartS achieves balanced performance across reasoning and perception tasks and remains competitive with baselines across categories. Moreover, it outperforms FOCUS on most tasks, with gains of 10.2 and 12.7 points on temporal reasoning and temporal perception, respectively. These tasks require identifying and combining cues distributed across different moments of the video, highlighting the effectiveness of PartS for temporal understanding. Counting remains the most challenging category for all methods, likely because sparse observations make it difficult to distinguish and track multiple similar instances across time.

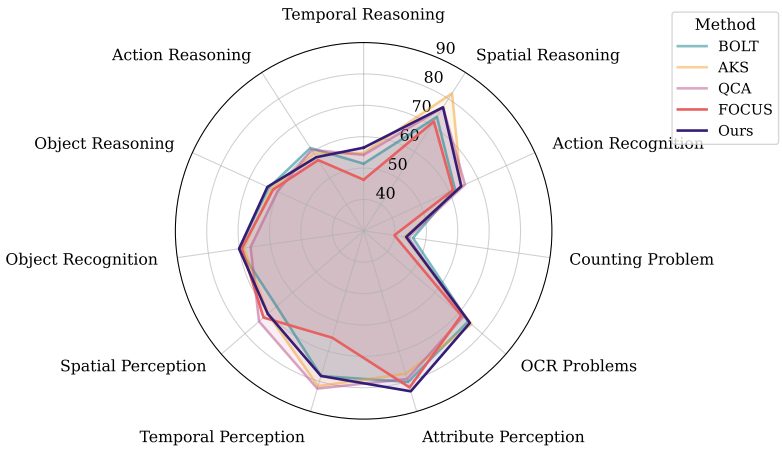


Figure A.2: **Performance across tasks on Video-MME.** Accuracy (%) of Qwen3-VL using keyframes selected by each method.

D Impact of Adaptive Candidate Sampling

We ablate the impact of our adaptive candidate sampling strategy under a matched latency budget, results are presented in Table A.2. Applying PartS to uniformly sampled candidates already consistently improves over standard uniform sampling, demonstrating the effectiveness of our query-aware frame selection independently of candidate construction. Replacing the uniform candidate pool with our adaptive sampling strategy yields a further improvement across all four MLLMs and all four benchmarks. Since both variants operate under the same latency budget, these improvements indicate that the gains do not arise from processing more candidates, but from allocating the available computational budget more effectively toward query-relevant regions. These results show that our adaptive candidate sampling and our frame selection methods provide complementary benefits.

Table A.2: **Impact of adaptive candidate sampling.** We compare PartS using uniformly sampled candidates and t our adaptive candidate sampling strategy under the same latency budget. Results report accuracy (%) across four MLLMs and long-video QA benchmarks. Best results for each model are shown in **bold**.

Model	Method	LongVideoBench	Video-MME	MLVU	LVBench	Avg.
LLaVA-OneVision-7B	Uniform	56.2	58.5	63.3	39.6	54.4
	PartS + Uniform	57.9	59.8	64.1	43.7	56.4
	PartS + Adaptive	**60.4**	**61.1**	**66.6**	**45.9**	**58.5**
Qwen2.5-VL-7B	Uniform	58.6	62.2	59.4	39.1	54.8
	PartS + Uniform	60.1	62.8	66.2	44.5	58.4
	PartS + Adaptive	**63.9**	**63.3**	**68.6**	**47.4**	**60.8**
LLaVA-Video-7B	Uniform	58.7	62.8	66.0	40.5	57.0
	PartS + Uniform	59.7	63.4	66.9	45.9	59.0
	PartS + Adaptive	**62.1**	**64.1**	**69.5**	**48.7**	**61.1**
Qwen3-VL-8B	Uniform	62.1	64.8	61.7	41.3	57.5
	PartS + Uniform	63.6	65.2	67.3	48.1	61.1
	PartS + Adaptive	**66.0**	**66.0**	**69.4**	**50.5**	**63.0**

E ALGORITHM

We provide the pseudocode of our method in Algorithms 1 and 2. Algorithm 1 describes the adaptive candidate sampling. It initializes the candidate pool $\mathcal{V}'$ with I-frames and forms temporal segments between consecutive I-frames longer than one second. Each segment is scored by the mean query relevance of its boundary frames and their visual similarity. At each iteration, the algorithm first identifies the M segments with the highest query relevance and, among them, selects the segment whose boundary frames have the lowest visual similarity. The selected segment $[f_{j_1}, f_{j_2}]$ is refined by sampling its temporal midpoint f'_j, which is added to $\mathcal{V}'$. The segment is subsequently divided into two child segments, $[f_{j_1}, f'_j]$ and $[f'_j, f_{j_2}]$. Each child is added back to the refinement pool only if it is longer than one second and its boundary-frame similarity is below the threshold β. The process stops when the candidate budget B is reached or no eligible segment remains.

Algorithm 2 then selects the final K keyframes from the resulting candidate pool. First, it employs K-Means to partition the candidate frames, using their visual embeddings $\mathbf{f}_i$, into c clusters. For each cluster C_j, we compute a cluster-to-query relevance score S_j as the average cosine similarity between its frames and the query embedding $\mathbf{q}$. These scores are sorted in descending order, and the algorithm iterates over the resulting ranked list to determine the allocation budget N_j for each cluster C_j. If the allocated budget exceeds the cluster size, all frames in that cluster are selected, and the remaining budget is carried over to the next cluster. Otherwise, if the allocated budget is smaller than the cluster size, the algorithm selects the top-N_j frames with the highest frame-to-query similarity scores within that cluster. The procedure continues until the frame budget K is reached, and the selected frames are returned as the final keyframe set $\mathcal{F}$.

F EVALUATION PROMPT

To further facilitate reproducibility, we provide below the prompts used in our experiments. Our prompts follow the template adopted by LMMs-Eval. (Zhang et al., 2024a).

LongVideoBench:

```
{Question}
A. {Option A}
B. {Option B}
...
Answer with the option's letter from the given choices directly.\n
```

Algorithm 1 Adaptive Candidate Sampling

Input: Video $\mathcal{V} = \{f_i\}_{i=1}^{L}$, query Q, max. candidate frame budget B, pool size M and max. visual similarity β.
Output: List $\mathcal{V}'$ of candidate frames.

$\mathcal{V}' \leftarrow$ I-Frames($\mathcal{V}$) ▷ Initialize candidate frame pool.
$\mathcal{I} \leftarrow [\,]$
for $i \in [0, \ldots, len(\mathcal{V}') - 1]$ **do**
  if duration$(f_i, f_{(i+1)}) > 1$ second **then**
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{[f_i, f_{(i+1)}]\}$
  end if
end for
Compute relevance $r_i = sim(f_i, Q) \quad \forall f_i \in \mathcal{V}'$;
Compute mean relevance $\bar{r}_s = (r_i + r_{i+1})/2 \quad \forall [f_i, f_{i+1}] \in \mathcal{I}$;
Compute visual similarity $\delta_s \quad \forall [f_i, f_{i+1}] \in \mathcal{I}$;
while $len(\mathcal{V}') < B$ and $len(\mathcal{I}) > 0$ **do**
  $\mathcal{I}' \leftarrow \underset{s \in \mathcal{I}}{\text{Top-M}} \quad \bar{r}_s$
  $[f_{j_1}, f_{j_2}] \leftarrow \underset{s \in \mathcal{I}'}{\arg\min} \quad \delta_s$
  $\mathcal{I} \leftarrow \mathcal{I} - \{[f_{j_1}, f_{j_2}]\}$ ▷ Remove $[f_{j_1}, f_{j_2}]$ from $\mathcal{I}$.
  $f'_j \leftarrow f_{\lfloor (j_1 + j_2)/2 \rfloor}$ ▷ Zoom-in on the relevant segment with lowest visual similarity.
  $\mathcal{V}' \leftarrow \mathcal{V}' \cup \{f'_j\}$
  $s_{\text{left}} \leftarrow [f_{j_1}, f'_j]$
  $s_{\text{right}} \leftarrow [f'_j, f_{j_2}]$
  Compute mean relevance $\bar{r}_{\text{left}}$ and $\bar{r}_{\text{right}}$;
  Compute visual similarity δ_{left} and δ_{right};
  if $\delta_{\text{left}} \leq \beta$ and duration$(s_{\text{left}}) > 1$ **then**
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{s_{\text{left}}\}$
  end if
  if $\delta_{\text{right}} \leq \beta$ and duration$(s_{\text{right}}) > 1$ **then**
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{s_{\text{right}}\}$
  end if
end while
return $\mathcal{V}'$

Algorithm 2 Keyframe Selection

Input: Video $V = \{f_i\}_{i=1}^{L}$, query Q, number of frames to select K, number of clusters c and allocation temperature τ.
Output: List $\mathcal{F}$ of selected frames, with $|\mathcal{F}| = K$.

$S_{frame}, S, \mathcal{F} \leftarrow [\,], [\,], [\,]$ ▷ Initialize frame-to-query, cluster-to-query and frame lists.
$V' \leftarrow$ Adaptive Candidate Sampling(V, Q); ▷ Candidate frames from V.
Compute CLIP embeddings $\{\mathbf{f}_i | f_i \in V'\}$ and $\mathbf{q}$; ▷ Leverages the embeddings computed in the Adaptive Candidate Frame sampling step.
Obtain clusters $\mathcal{C} = \{C_1, \ldots, C_c\}$ executing K-Means over $\{\mathbf{f}_i\}$;
for $C_j \in \mathcal{C}$ **do**
    $S_j \leftarrow 0$
    for $i \in C_j$ **do**
        $score \leftarrow \frac{\mathbf{f}_i^\top \mathbf{q}}{\|\mathbf{f}_i\| \, \|\mathbf{q}\|}$ ▷ Compute frame-to-query similarity.
        $S_{frame} \leftarrow S_{frame} \cup \{score\}$
        $S_j \leftarrow S_j + score$
    end for
    $S_j \leftarrow S_j / len(C_j)$ ▷ Cluster-to-query similarity.
    $S \leftarrow S \cup \{S_j\}$
end for
$carry, j \leftarrow 0, 1$
Sort S in decreasing order;
while $len(\mathcal{F}) < K$ and $j \leq c$ **do** ▷ For each cluster, compute frame allocation.
    $N_j \leftarrow \lceil K \cdot \frac{exp(S_j/\tau)}{\sum_{i=1}^{c} exp(S_i/\tau)} \rceil + carry$
    if $len(C_j) \leq N_j$ **then**
        $\mathcal{F}_j \leftarrow C_j$
        $carry \leftarrow N_j - len(C_j)$
    else
        $S^c_{frame} \leftarrow$ Similarity scores S_{frame} for frames in C_j;
        $\mathcal{F}_j \leftarrow$ Indices of the top-N_j frames with highest S^c_{frame} scores;
        $carry \leftarrow 0$
    end if
    $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{F}_j$
    $j \leftarrow j + 1$
end while
return $\mathcal{F}$

Video-MME:

```
Select the best answer to the following multiple-choice question
based on the video and the subtitles.
{Question}
A. {Option A}
B. {Option B}
...
Respond with only the letter (A, B, C, or D) of the correct option.
Answer with the option's letter from the given choices directly.
\n
```

MLVU:

```
{Question}
A. {Option A}
B. {Option B}
...
Only give the best option.  Best option:\n
```

LVBench:

```
{Question}
A. {Option A}
B. {Option B}
...
Answer the question with the option letter.\n
```