TEMPORAL QUANTIZATION FOR ADAPTIVE CANDIDATE FRAME CONSTRUCTION IN VIDEO QA

Anonymous authors
Paper under double-blind review

ABSTRACT

Video language models (VLMs) operate under limited visual-token and computational budgets, requiring videos to be represented by a compact set of frames. A common pipeline first constructs a temporally subsampled candidate frame set and then selects the final input frames from it. While conventional pipelines rely on query-agnostic temporal sampling, recent work has increasingly explored adaptive frame acquisition guided by the query and partial observations. Yet, how a fixed observation budget should represent the video timeline before downstream selection remains less explicitly characterized. To address this, we introduce `QUEST`, a training-free method that formulates candidate construction as relevance-weighted temporal quantization, explicitly measuring how well the acquired observations resolve the timeline. By jointly considering estimated query relevance and temporal gaps between observations, `QUEST` allocates finer temporal resolution to relevant yet under-resolved regions while maintaining coverage elsewhere. The resulting candidate set can be used with existing frame selectors without modifying their selection rules. Extensive experiments across multiple VLMs, frame selectors, and video understanding benchmarks demonstrate improved downstream accuracy under matched observation and selection budgets.

1 INTRODUCTION

Video language models (VLMs) process multiple video frames as visual input Bai et al. (2025b;a); Li et al. (2024); Wang et al. (2025b); Zhang et al. (2024b), but processing video remains computationally demanding because each additional frame introduces visual tokens and increases memory and inference costs. Practical VLM pipelines therefore subsample frames along the video timeline to obtain a manageable input. For example, LLaVA-Video Zhang et al. (2024b) and InternVL3.5 Wang et al. (2025b) employ uniform temporal sampling in their standard video inference pipelines. When frame selection is further applied, this temporally subsampled frame set serves as the candidate frame set from which final VLM input is selected.

Given a candidate frame set, frame selection methods select a compact subset of informative frames for VLM inference Zhang et al. (2025b); Tang et al. (2025); Yu et al. (2024); Sun et al. (2025); Liu et al. (2025a). Many operate on predefined candidates obtained through uniform sampling Yu et al. (2024); Sun et al. (2025); Zhang et al. (2025b) or denser predefined sampling Liu et al. (2025a); Tang et al. (2025). As illustrated in Figure 1(a), however, a selector can only choose among the frames made available during candidate construction. Evidence omitted at this stage cannot be recovered downstream, regardless of how effectively the selector ranks or selects the remaining frames. Therefore, candidate construction directly determines the evidence available to subsequent frame selection.

Recent work addresses this limitation by adapting frame acquisition to the query and the observations collected so far, as illustrated in Figure 1(b). A common signal in these methods is query relevance, *i.e.,* how strongly an observed frame is associated with the input query, which can guide where to acquire additional frames. FOCUS Zhu et al. (2025) allocates observations using estimated relevance and statistical uncertainty to identify promising temporal regions, with uncertainty defined over the estimated relevance of temporal regions rather than the temporal gaps between observations. A.I.R. Zou et al. (2025) instead uses repeated VLM assessment to validate promising frames and guide further acquisition, introducing additional VLM inference into the acquisition process. FOCUS and

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.

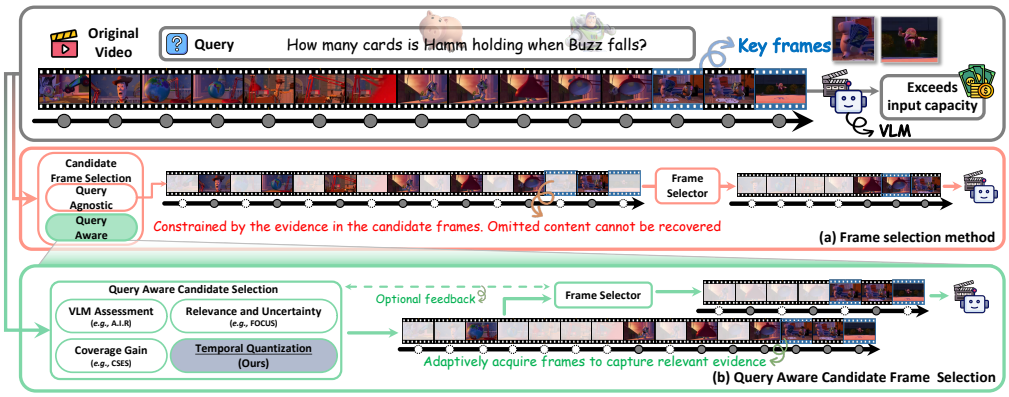

Figure 1: **Overview of frame selection and adaptive candidate construction.** (a) The downstream selector can only choose from the constructed candidate set. (b) Existing adaptive methods use relevance, uncertainty, VLM assessment, or coverage, whereas QUEST allocates observations through relevance-weighted temporal quantization.

A.I.R. demonstrate complementary strategies for adaptive acquisition through relevance uncertainty and VLM-guided assessment. More generally, however, deciding where to observe next requires considering not only what the current observations suggest is relevant, but also which parts of the timeline remain insufficiently resolved. A temporally sparse region may still contain useful evidence that has not yet been observed.

This motivates explicitly accounting for temporal coverage during acquisition. Among recent adaptive acquisition methods, CSES Zhang et al. (2026) further incorporates temporal coverage into adaptive acquisition, prioritizing observations according to their estimated marginal coverage gain. In this work, we approach the same underlying problem from a temporal quantization perspective. Under a fixed observation budget, the acquired frames form a finite temporal representation of the video. Quantization is therefore directly relevant to where frames should be sampled, because the spacing between observations determines the temporal resolution of this representation. We combine this resolution with query relevance through a relevance-weighted temporal covering radius and prioritize the largest such gap. Unlike aggregate coverage gain, this worst-case criterion explicitly targets the most under-resolved query-relevant region. Holding the current relevance estimates fixed, the formulation yields a one-step minimax rule for determining both which interval to sample and where to place the next observation. QUEST reuses the downstream selector's frame-level relevance scorer while leaving its selection rule unchanged.

Our contributions are summarized as follows:

- We formulate candidate frame construction under a fixed observation budget as relevance-weighted temporal quantization, characterizing temporal resolution through query-weighted gaps between acquired frames.
- We introduce QUEST, a training-free adaptive acquisition method that applies the resulting one-step minimax rule to determine which temporal interval to sample and where to place the next observation, while leaving the downstream frame selection rule unchanged.
- We evaluate QUEST across five VLMs, four frame selectors, and five video QA benchmarks, demonstrating improved downstream accuracy under matched observation and selection budgets.

2 Problem Formulation

This section introduces the problem formulation for video QA. We first describe the general setting and then specialize it to the two-stage candidate-construction and frame-selection pipeline studied in this work.

2.1 Video Question Answering

Given a video-question pair (V, q), let $V = \{v_i\}_{i=1}^{T}$ denote a video consisting of T frames and q the corresponding question. A video QA system processes visual evidence from V together with q to

predict an answer $\hat{a}$. Since processing the entire video can be computationally expensive, practical systems often reduce the visual input before inference, for example through frame selection. Let $X \subseteq V$ denote the set of frames ultimately provided to the $\texttt{VLM}(\cdot)$, and the prediction is then $\hat{a} = \texttt{VLM}(X, q)$. The construction of X may involve predefined sampling, adaptive frame acquisition, frame selection, or combinations of these operations.

2.2 Candidate Frame Construction with Downstream Frame Selection

We specialize the general video QA setting to a two-stage pipeline consisting of candidate frame construction followed by downstream frame selection. Given a video V, query q, and observation budget N, a candidate-construction function F_{cand} constructs a candidate frame set

$$S_N = F_{\text{cand}}(V, q, N), \quad S_N \subseteq V, |S_N| = N.$$

The candidate set determines the visual evidence available to the downstream selector. Given a selection budget $K \leq N$, a frame selector F_{sel} then selects

$$S_K = F_{\text{sel}}(S_N, q, K), \quad S_K \subseteq S_N, |S_K| = K,$$

and the selected frames are provided to the VLM to predict the answer $\hat{a} = \texttt{VLM}(S_K, q)$. The two budgets control distinct stages of the pipeline: N determines how many frames are acquired as candidate evidence, whereas K determines how many of these frames are retained for $\texttt{VLM}$ inference. Our focus is the construction of S_N. We therefore compare candidate-construction methods while keeping the observation budget N, selection budget K, and downstream selection rule F_{sel}. This isolates the effect of candidate construction on downstream performance.

3 Temporal Quantization for Candidate Frame Construction

This section develops the temporal quantization principle underlying our candidate-construction method. We first introduce quantization and worst-case covering, and then apply them to the video timeline to define a query-aware temporal representation objective. Finally, we derive a next-frame sampling rule from this objective.

3.1 Why Temporal Quantization and Worst-case Covering?

From quantization to temporal quantization. Quantization studies how a continuous or high-resolution source can be represented using a finite set of discrete representatives Lloyd (1982); Gersho (1979); Gray & Neuhoff (1998). When applied along a temporal axis, the representatives correspond to samples placed at selected times. We refer to this setting as *temporal quantization*. The spacing between samples determines the temporal resolution of the representation: closely spaced samples provide finer temporal resolution, whereas large gaps leave the corresponding temporal regions more coarsely represented. Thus, temporal quantization directly connects representation quality to where samples are placed along the timeline. Candidate frame construction has the same structure, since a fixed frame budget requires the full video timeline to be represented by only a finite number of candidate frames, as illustrated in Figure 2.

Why worst-case covering? To evaluate a finite temporal representation, we need a criterion that reflects how poorly any part of the timeline is resolved. An average criterion can remain small even when a large gap persists between neighboring observations. For candidate frame construction, such a gap may contain brief query-relevant evidence that is never acquired and therefore cannot be recovered by downstream selection. We therefore adopt a *worst-case covering perspective*, which measures the largest distance from any temporal position to its nearest observation Kesseböhmer & Zhu (2003). As illustrated in Figure 2(b), this criterion directly captures the most coarsely represented region of the timeline.

Incorporating query relevance. Worst-case temporal covering alone treats all temporal regions equally. In video question answering, however, query-relevant regions may warrant finer temporal resolution. Temporal covering and query relevance therefore provide complementary signals: the former indicates where the timeline remains poorly resolved, while the latter indicates where finer resolution is likely to be more useful. Figure 2 summarizes this intuition, showing how temporal gaps and query relevance are combined into the sampling principle developed below.

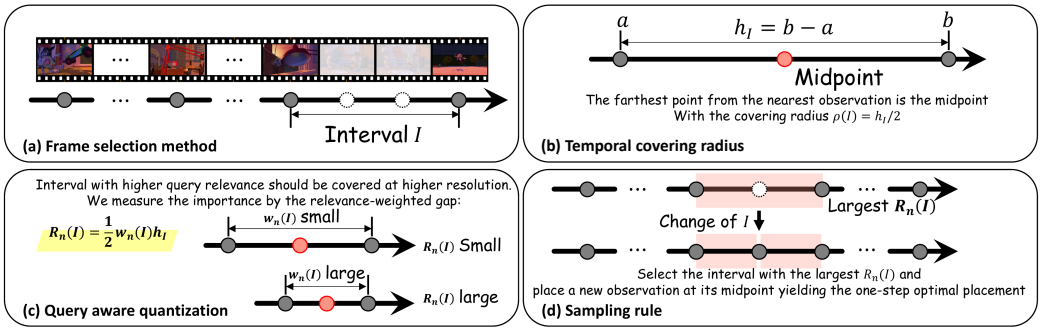

Figure 2: **Intuition of relevance-weighted temporal quantization.** (a) Candidate frames form a finite representation of the video timeline. (b) The covering radius of an interval I is determined by its midpoint. (c) Query relevance weights the temporal gap as $R_n(I) = \frac{1}{2} w_n(I) h_I$. (d) Under fixed interval weights, the one-step optimal rule samples the midpoint of the largest $R_n(I)$.

3.2 Query-Aware Temporal Quantization of Candidate Frames

Building on the intuition above, we now formalize candidate frame construction as query-aware temporal quantization. Because the candidate set is constructed sequentially, we consider an intermediate stage with n sampled candidate frames. Let S_n denote the current candidate set, and S_N the final set when the candidate budget N is reached.

Temporal intervals between candidate frames. Recall that $V = \{v_i\}_{i=1}^{T}$ denotes a video of T frames. We write the current candidate set as $S_n = \{v_{i_j}\}_{j=1}^{n}$, where i_j is the frame index of the j-th candidate and $i_1 < \cdots < i_n$. Since the first and last video frames are included in S_n, consecutive candidate frames partition the timeline into temporal intervals. For two consecutive sampled frame indices $a < b$, let $I = [a, b]$ denote the interval between them and let $h_I = b - a$ denote its temporal gap.

Worst-case temporal covering. Within an interval $I = [a, b]$, the temporal position farthest from its nearest candidate frame is the midpoint. Following the worst-case covering perspective of Kesseböhmer & Zhu (2003), we define the covering radius of I as $\rho(I) := \frac{h_I}{2}$. This radius measures the worst temporal resolution within the interval: a larger h_I indicates that some temporal locations lie farther from the nearest candidate frame and are therefore more coarsely represented. Under a purely temporal criterion, minimizing the largest covering radius therefore is equivalent to minimizing the largest gap between consecutive candidate frames.

Query-aware temporal covering. Temporal covering alone treats all intervals equally, although their relevance to the query may differ. We therefore assign each interval I a nonnegative weight $w_n(I)$, representing its estimated query relevance from the current observations. The specific estimation of $w_n(I)$ is described in Section 4. Combining query relevance with temporal covering, we define the relevance-weighted covering radius as

$$R_n(I) := w_n(I)\rho(I) = \frac{1}{2} w_n(I) h_I.$$

Here, $w_n(I)$ captures how relevant the interval is estimated to be to the query, while h_I reflects the temporal gap between observations. A large $R_n(I)$ therefore identifies a region that is both query-relevant and insufficiently resolved.

Worst-case query-aware objective. We define the current worst-case query-aware covering error as the largest relevance-weighted covering radius, $E_n := \max_I R_n(I)$. Thus, E_n measures the largest relevance-weighted temporal gap in the current candidate representation. The next subsection uses this objective to determine where an additional candidate frame should be sampled based only on the information available at the current stage.

3.3 Sequential Candidate Construction from Partial Observations

Sequential construction from current evidence. The candidate set must be constructed without access to the relevance of unsampled frames. At step n, only the current observations and the resulting

interval weights $w_n(I)$ are available. We therefore choose the next candidate frame using the current objective E_n, seeking a placement that minimizes the worst-case query-aware covering error after one additional observation.

Prospective sampling under current relevance. Consider placing one additional candidate frame at position z within a current interval $I = [a, b]$. Because the new frame is unavailable before sampling, we hold the current interval weight $w_n(I)$ fixed and assign it to both subintervals created by z. Their temporal gaps are $z - a$ and $b - z$, so the larger relevance-weighted covering radius becomes

$$R_n^+(I, z) := \frac{w_n(I)}{2} \max\{z - a, b - z\}.$$

Since all other intervals remain unchanged, the resulting worst-case error is

$$E_n^+(I, z) := \max\left\{ \max_{I' \neq I} R_n(I'),\ R_n^+(I, z) \right\}.$$

We seek the interval I and position z that minimize $E_n^+(I, z)$.

Proposition 3.1 (One-step minimax candidate placement)**.** *Fix the current candidate set S_n and the interval weights $w_n(I)$. Let*

$$I_n^\star \in \arg\max_I R_n(I) = \arg\max_I w_n(I) h_I$$

among intervals containing an unsampled frame. If $I_n^\star = [a, b]$, then the one-step worst-case covering error is minimized by placing the next observation at the interval midpoint $z_n^\star = (a + b)/2$.

Proof. For a fixed interval $I = [a, b]$, all other interval errors are independent of the sampling position z. Minimizing the resulting worst-case error within I therefore reduces to minimizing $\max\{z - a, b - z\}$, which is minimized when the two subintervals have equal length, *i.e.,* at $z = (a + b)/2$. We then choose which interval to sample. Sampling any interval other than one with the largest current $R_n(I)$ leaves that maximum error unchanged. Sampling $I_n^\star$ at its midpoint, in contrast, halves its relevance-weighted covering radius under the fixed interval weight. Therefore, selecting $I_n^\star$ and sampling its midpoint attains the one-step minimum of the worst-case query-aware covering error. □

Implication for candidate construction. Proposition 3.1 yields a simple one-step sampling rule: Under fixed interval relevance estimates, the worst-case query-weighted covering error is minimized by selecting the interval with the largest $w_n(I)h_I$ and sampling its midpoint. This result specifies where to sample once the interval relevance weights are given. In the next section, we describe how QUEST estimates these weights from the observed frames and applies this rule sequentially until the candidate budget is reached.

4 QUEST: Adaptive Candidate Frame Construction

Section 3 derives the one-step sampling rule assuming that interval relevance weights are given. We now introduce QUEST, which estimates these weights from observed frame-level relevance scores and applies the rule sequentially until the candidate budget is reached.

4.1 Initialization and Interval Relevance

Initial candidate set. Given a candidate budget N, QUEST starts from $W \leq N$ approximately uniformly spaced frames spanning the video timeline, including the first and last frames. These initial frames provide a coarse initial temporal representation and remain in the final candidate set. The complete procedure, including initialization on a 512-point reference index grid, is given in Algorithm 1.

Frame and interval relevance. For each sampled frame v_i, we compute its query-relevance score $r_i := \mathrm{Rel}(v_i, q)$, where Rel is the frame-level scorer used by the downstream selector. At step n, the

observed relevance scores are normalized using $r_n^{\min} := \min_{v_i \in S_n} r_i, r_n^{\max} := \max_{v_i \in S_n} r_i$. For an interval $I = [a, b]$, we form the mean score of its observed endpoints,

$$u_n(I) := \frac{(r_a + r_b)/2 - r_n^{\min}}{r_n^{\max} - r_n^{\min}},$$

and define the interval weight as $w_n(I) := \exp(u_n(I))$. Since $u_n(I) \in [0, 1]$, $w_n(I) \in [1, e]$, which limits the influence of relevance while preserving the temporal-gap term in the sampling priority.

Sequential candidate construction. At each step, QUEST applies the sampling rule derived in Section 3 by selecting

$$I_n^\star \in \arg\max_I w_n(I) h_I,$$

and samples the frame nearest to its midpoint. The new frame is added to S_n, its relevance score is computed, and the selected interval is split with the interval weights recomputed from the updated observations. This process is repeated until $|S_N| = N$. After construction, the candidate frames are ordered temporally and passed to the downstream frame selector together with their cached raw relevance scores. The normalization and interval weights above are used only for candidate construction; the downstream scoring and selection rule remain unchanged.

5 Experiments

We evaluate whether adaptive candidate construction improves video understanding under fixed observation and selection budgets. We compare QUEST with uniform candidate construction across five VLMs and four frame selectors, and further analyze comparisons with adaptive and 1-FPS baselines, computational cost, and initialization sensitivity.

5.1 Experimental Setup

Datasets. We evaluate on five video understanding benchmarks: VideoMME Fu et al. (2025a) with subtitles, LongVideoBench Wu et al. (2024), MLVU Zhou et al. (2025), LVBench Wang et al. (2025a), and a deterministic subset of CG-Bench-mini Chen et al. (2024a). Details of the CG-Bench-mini subset are provided in Appendix A.3.

Models. We evaluate five VLMs in the 7B–8B parameter range: Qwen2-VL Wang et al. (2024), Qwen2.5-VL Bai et al. (2025b), Qwen3-VL Bai et al. (2025a), LLaVA-OneVision-2 An et al. (2026), and LLaVA-Video Zhang et al. (2024b).

Implementation details. We integrate QUEST as a plug-in for four frame selectors: Q-Frame Zhang et al. (2025b), BOLT Liu et al. (2025a), MDP^3 Sun et al. (2025), and Top-K. For each selector, QUEST uses the same frame-level scorer Rel and reuses the scores computed during candidate construction. The scoring encoders are CLIP-L/14 Radford et al. (2021), SigLIP-SO400M Zhai et al. (2023), and LongCLIP-L/14 Zhang et al. (2024a), depending on the selector. We set the observation budget to $N = 128$ and the selection budget to $K = 32$, except for Qwen2-VL, which uses $K = 8$. Both the uniform baseline and QUEST provide N candidate frames to the same downstream selector; BOLT therefore operates on this fixed-size candidate set rather than its original 1-FPS candidate pool. Additional implementation and selector-specific settings are provided in Appendix A.4.

5.2 Main Results

Comparison with uniform candidate construction. As shown in Table 1, QUEST improves the average performance of all five VLMs under matched observation and selection budgets. Across all models, selectors, and benchmarks, QUEST provides an average gain of $+0.87$ percentage points over uniform candidate construction. Notable improvements include $+1.57$ points with LLaVA-Video and BOLT and $+1.52$ points with Qwen2.5-VL and Top-K, with individual benchmark gains reaching up to $+3.68$ points on LVBench. These results demonstrate that improving candidate placement can yield substantial downstream gains without changing the selection rule or increasing either frame budget.

Table 1: Main results across models, selectors, and benchmarks. Each pair of rows compares the base frame selection method with the uniform pool (✗) and the same method combined with QUEST (✓). All entries report absolute accuracies (%), where bold indicates the best within each pair.

Model	Frame Selector	QUEST	LongVB		MLVU		VideoMME		LVBench		CG-Bench		Avg.	
			Acc.	Δ	Acc.	Δ	Acc.	Δ	Acc.	Δ	Acc.	Δ	Acc.	Δ
Qwen2-VL	Q-Frame	✗	56.32	–	62.81	–	58.30	–	40.80	–	32.46	–	50.14	–
		✓	**56.77**	**+0.45**	**62.96**	**+0.15**	**59.04**	**+0.74**	**41.77**	**+0.97**	**33.21**	**+0.75**	**50.75**	**+0.61**
	BOLT	✗	53.18	–	60.30	–	55.89	–	36.22	–	31.03	–	47.32	–
		✓	**55.65**	**+2.47**	**60.71**	**+0.41**	**57.22**	**+1.33**	**36.41**	**+0.19**	**31.93**	**+0.90**	**48.38**	**+1.06**
	Top-K	✗	55.57	–	**63.33**	–	58.48	–	38.22	–	33.28	–	49.78	–
		✓	**55.65**	**+0.08**	63.05	**-0.28**	**58.78**	**+0.30**	**41.90**	**+3.68**	**36.36**	**+3.08**	**51.15**	**+1.37**
	MDP^3	✗	**56.10**	–	61.06	–	**59.04**	–	40.15	–	33.43	–	49.96	–
		✓	55.50	**-0.60**	**62.52**	**+1.46**	58.19	**-0.85**	**40.28**	**+0.13**	**33.58**	**+0.15**	**50.01**	**+0.06**
	Mean	✗	55.29	–	61.88	–	57.93	–	38.85	–	32.55	–	49.30	–
		✓	**55.89**	**+0.60**	**62.31**	**+0.44**	**58.31**	**+0.38**	**40.09**	**+1.24**	**33.77**	**+1.22**	**50.07**	**+0.78**
Qwen2.5-VL	Q-Frame	✗	61.78	–	68.34	–	65.48	–	43.51	–	41.55	–	56.13	–
		✓	61.78	0.00	**69.05**	**+0.71**	**65.70**	**+0.22**	**43.64**	**+0.13**	**41.85**	**+0.30**	**56.40**	**+0.27**
	BOLT	✗	60.73	–	68.96	–	**66.55**	–	40.35	–	39.97	–	55.31	–
		✓	**62.00**	**+1.27**	**70.16**	**+1.20**	66.07	**-0.48**	**42.29**	**+1.94**	**41.77**	**+1.80**	**56.46**	**+1.15**
	Top-K	✗	60.13	–	67.80	–	66.11	–	41.31	–	40.35	–	55.14	–
		✓	**61.63**	**+1.50**	**70.37**	**+2.57**	**66.26**	**+0.15**	**43.19**	**+1.88**	**41.85**	**+1.50**	**56.66**	**+1.52**
	MDP^3	✗	59.91	–	66.78	–	**67.19**	–	41.96	–	40.57	–	55.28	–
		✓	**60.81**	**+0.90**	**69.08**	**+2.30**	67.04	**-0.15**	**42.48**	**+0.52**	**41.62**	**+1.05**	**56.21**	**+0.92**
	Mean	✗	60.64	–	67.97	–	**66.33**	–	41.78	–	40.61	–	55.47	–
		✓	**61.56**	**+0.92**	**69.67**	**+1.70**	66.27	**-0.07**	**42.90**	**+1.12**	**41.77**	**+1.16**	**56.43**	**+0.97**
Qwen3-VL	Q-Frame	✗	64.92	–	72.08	–	68.70	–	45.51	–	44.55	–	59.15	–
		✓	**65.07**	**+0.15**	**73.00**	**+0.92**	**69.22**	**+0.52**	**46.80**	**+1.29**	**45.23**	**+0.68**	**59.86**	**+0.71**
	BOLT	✗	63.50	–	71.86	–	**71.37**	–	44.87	–	44.63	–	59.25	–
		✓	**64.92**	**+1.42**	**73.96**	**+2.10**	70.59	**-0.78**	**46.03**	**+1.16**	**44.85**	**+0.22**	**60.07**	**+0.82**
	Top-K	✗	62.90	–	72.27	–	69.19	–	46.09	–	44.40	–	58.97	–
		✓	**63.80**	**+0.90**	**73.40**	**+1.13**	**70.26**	**+1.07**	**46.55**	**+0.46**	**45.30**	**+0.90**	**59.86**	**+0.89**
	MDP^3	✗	**62.90**	–	71.58	–	69.33	–	45.19	–	42.52	–	58.30	–
		✓	62.45	**-0.45**	**73.97**	**+2.39**	**70.00**	**+0.67**	**47.45**	**+2.26**	**44.93**	**+2.41**	**59.76**	**+1.46**
	Mean	✗	63.56	–	71.95	–	69.65	–	45.42	–	44.03	–	58.92	–
		✓	**64.06**	**+0.51**	**73.58**	**+1.64**	**70.02**	**+0.37**	**46.71**	**+1.29**	**45.08**	**+1.05**	**59.89**	**+0.97**
LLaVA-Video	Q-Frame	✗	60.21	–	69.67	–	65.48	–	44.87	–	35.76	–	55.20	–
		✓	**60.43**	**+0.22**	**69.89**	**+0.22**	**66.15**	**+0.67**	**45.19**	**+0.32**	**37.11**	**+1.35**	**55.75**	**+0.56**
	BOLT	✗	58.86	–	69.84	–	67.30	–	43.38	–	36.36	–	55.15	–
		✓	**62.08**	**+3.22**	**70.69**	**+0.85**	**67.78**	**+0.48**	**44.74**	**+1.36**	**38.32**	**+1.96**	**56.72**	**+1.57**
	Top-K	✗	59.39	–	**70.24**	–	**67.89**	–	45.06	–	37.87	–	56.09	–
		✓	**61.63**	**+2.24**	70.19	**-0.05**	67.85	**-0.04**	**45.58**	**+0.52**	**39.22**	**+1.35**	**56.89**	**+0.80**
	MDP^3	✗	59.16	–	70.43	–	67.44	–	45.13	–	37.27	–	55.89	–
		✓	**60.88**	**+1.72**	**71.40**	**+0.97**	**67.48**	**+0.04**	**46.48**	**+1.35**	**38.62**	**+1.35**	**56.97**	**+1.09**
	Mean	✗	59.41	–	70.05	–	67.03	–	44.61	–	36.82	–	55.58	–
		✓	**61.26**	**+1.85**	**70.54**	**+0.50**	**67.32**	**+0.29**	**45.50**	**+0.89**	**38.32**	**+1.50**	**56.59**	**+1.01**
LLaVA-OV2	Q-Frame	✗	62.45	–	72.27	–	67.44	–	43.64	–	40.57	–	57.27	–
		✓	**63.43**	**+0.98**	**72.92**	**+0.65**	**67.59**	**+0.15**	**44.67**	**+1.03**	**40.87**	**+0.30**	**57.90**	**+0.62**
	BOLT	✗	62.83	–	73.29	–	68.07	–	43.96	–	39.52	–	57.53	–
		✓	**63.35**	**+0.52**	**73.35**	**+0.06**	**68.63**	**+0.56**	**44.29**	**+0.33**	**41.40**	**+1.88**	**58.20**	**+0.67**
	Top-K	✗	60.58	–	72.11	–	**69.15**	–	**45.00**	–	40.65	–	57.50	–
		✓	**62.15**	**+1.57**	**73.51**	**+1.40**	68.30	**-0.85**	44.67	**-0.33**	**41.92**	**+1.27**	**58.11**	**+0.61**
	MDP^3	✗	**62.53**	–	73.06	–	68.11	–	43.38	–	40.72	–	57.56	–
		✓	62.38	**-0.15**	**74.53**	**+1.47**	**68.30**	**+0.19**	**43.83**	**+0.45**	**42.37**	**+1.65**	**58.28**	**+0.72**
	Mean	✗	62.10	–	72.68	–	68.19	–	44.00	–	40.37	–	57.47	–
		✓	**62.83**	**+0.73**	**73.58**	**+0.90**	**68.21**	**+0.01**	**44.37**	**+0.37**	**41.64**	**+1.28**	**58.12**	**+0.66**

Comparison with 1-FPS candidate sampling. As shown in Table 2, QUEST remains competitive with frame selectors operating on dense 1-FPS candidate pools. Compared with AKS, the selector-average QUEST improves average accuracy by up to $+1.58$ percentage points across the evaluated VLMs. It also achieves comparable or higher accuracy than the original 1-FPS BOLT setting, while using a fixed candidate budget of $N = 128$. These results show that effective candidate placement can match or outperform selection from substantially denser temporal sampling.

Table 2: Comparison with frame selection from 1-FPS candidates. QUEST reports the average and best performance over four selectors.

Model	Method	LongVB	MLVU	VideoMME	Avg.
Qwen3-VL	AKS	**65.07**	69.64	68.22	67.64
	BOLT	62.00	70.49	67.85	66.78
	QUEST (Avg.)	64.06	73.58	70.02	69.22
	QUEST (Best)	**65.07**	**73.97**	**70.59**	**69.88**
LLaVA-Video	AKS	61.18	67.26	**68.00**	65.48
	BOLT	60.88	69.15	66.22	65.42
	QUEST (Avg.)	61.26	70.54	67.32	66.37
	QUEST (Best)	**62.08**	**71.40**	67.85	**67.11**
LLaVA-OV2	AKS	63.05	70.66	67.59	67.10
	BOLT	61.03	72.65	66.78	66.82
	QUEST (Avg.)	62.83	73.58	68.21	68.20
	QUEST (Best)	**63.43**	**74.53**	**68.63**	**68.86**

Table 3: Comparison with adaptive frame acquisition methods. QUEST reports the average and best performance over four selectors.

Model	Method	LongVB	MLVU	VideoMME	Avg.
Qwen3-VL	CSES	63.43	69.95	68.11	67.16
	FOCUS	63.95	69.55	66.22	66.57
	QUEST (Avg.)	64.06	73.58	70.02	69.22
	QUEST (Best)	**65.07**	**73.97**	**70.59**	**69.88**
LLaVA-Video	CSES	60.36	68.14	66.59	65.03
	FOCUS	60.43	66.82	64.11	63.79
	QUEST (Avg.)	61.26	70.54	67.32	66.37
	QUEST (Best)	**62.08**	**71.40**	**67.85**	**67.11**
LLaVA-OV2	CSES	63.05	69.51	66.93	66.50
	FOCUS	61.41	67.71	63.89	64.34
	QUEST (Avg.)	62.83	73.58	68.21	68.20
	QUEST (Best)	**63.43**	**74.53**	**68.63**	**68.86**

Comparison with adaptive frame acquisition. As shown in Table 3, QUEST also outperforms recent adaptive acquisition methods. The selector-average QUEST improves over CSES by up to $+2.06$ percentage points and over FOCUS by up to $+3.87$ points in average accuracy across benchmarks. This shows that the gains of relevance-weighted temporal quantization persist even when compared with methods that adaptively determine where to acquire additional frames.

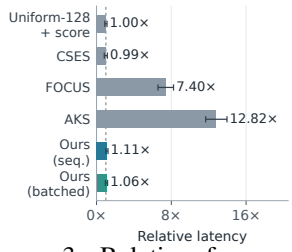

Figure 3: Relative frame selection latency, normalized by Uniform-128 + score (12.84 s). Error bars indicate ± 1 SD.

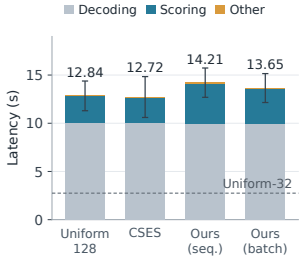

Figure 4: Frame selection latency by stage. Error bars indicate ± 1 SD of total latency; the dashed line denotes Uniform-32.

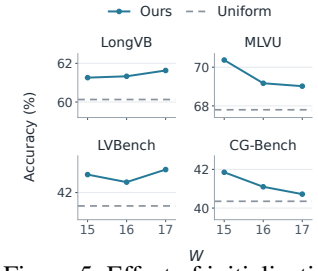

Figure 5: Effect of initialization size W with Qwen2.5-VL and Top-K. Dashed lines denote uniform candidate-pool baselines.

5.3 Analysis

Latency analysis. We measure frame selection latency on three approximately one-hour VideoMME videos using the same decoder and BLIP-based relevance scorer on an NVIDIA H100 GPU. As shown in Figure 3, sequential and batched QUEST require 14.21 s and 13.65 s, respectively, compared with 12.84 s for Uniform-128 + score and 12.72 s for CSES. The additional overhead mainly comes from repeated relevance scoring, while decoding time remains nearly unchanged. In contrast, FOCUS and AKS require 95.08 s and 164.56 s, respectively, due to scoring substantially more frames. Thus, QUEST maintains latency close to fixed-budget sampling while remaining substantially cheaper than methods using much denser candidate sets.

Latency profiling. Figure 4 shows that the additional latency of QUEST mainly comes from repeated relevance scoring, while video decoding time remains nearly unchanged. Batched acquisition reduces this overhead by grouping multiple certified observations into fewer scoring calls, bringing the total latency closer to the uniform baseline. This indicates that the computational overhead of adaptive candidate construction is concentrated in scoring rather than video decoding.

Sensitivity to initialization size. We examine $W \in [15, 16, 17]$ while fixing $N = 128$ and $K = 32$, using Qwen2.5-VL with Top-K selection. Figure 5 shows accuracy variations of $0.37 - 1.35$ percentage points across the evaluated benchmarks, with the best-performing initialization size differing by benchmark. The initialization sizes used in our main experiments are specified in Appendix A.4.

6 Related Work

Video LLMs for long-video understanding. Recent video LLMs extend image–text architectures to video by encoding sampled frames as visual tokens alongside text Li et al. (2024); Wang et al. (2024); Bai et al. (2025b;a); Zhang et al. (2024b); Wang et al. (2025b); Cheng et al. (2024). Representative models such as Qwen2-VL, Qwen2.5-VL, and Qwen3-VL support dynamic-resolution visual processing and increasingly long multimodal contexts Wang et al. (2024); Bai et al. (2025b;a), while LLaVA-OneVision and LLaVA-Video extend multimodal instruction tuning to image and video inputs Li et al. (2024); Zhang et al. (2024b). Processing more frames, however, introduces more visual tokens and higher memory and computation costs, which becomes important for long-video benchmarks Fu et al. (2025a); Wu et al. (2024); Zhou et al. (2025); Mangalam et al. (2023).

Efficient video understanding. Prior work mainly improves efficiency through token reduction or frame selection. Token reduction methods prune or merge visual tokens before or within the LLM Yang et al. (2025a); Tao et al. (2025); Fu et al. (2025b); Huang et al. (2025); Xing et al. (2024); Cheng et al. (2025); Li et al. (2025); Liu et al. (2025b); Shen et al. (2025); Chen et al. (2024b), with recent approaches additionally preserving spatio-temporal structure or combining multiple reduction stages Shao et al. (2025); Hyun et al. (2025); Kim et al. (2025). Frame selection instead reduces the visual input before VLM inference by retaining a compact subset of frames.

Frame selection from candidate frames. Many frame selection methods operate on candidates obtained through predefined temporal sampling. Q-Frame Zhang et al. (2025b), KeyVideoLLM Liang et al. (2024), MDP^3 Sun et al. (2025), and Frame-Voyager Yu et al. (2024) select frames using relevance, diversity, temporal structure, or VLM feedback. BOLT Liu et al. (2025a), AKS Tang et al. (2025), WFS-SB Chen et al. (2026), and CoSeLET Devnani et al. (2026) further exploit denser candidates or temporal structure, often from 1-FPS sampling. DIG Li et al. (2026a), Flexible Frame Selector Buch et al. (2025), and Adaptive Greedy Frame Selection Huang et al. (2026) adapt the selection strategy or output budget to the query. In contrast, our focus is the temporal placement of candidate observations before downstream selection.

Adaptive frame acquisition. Recent methods adapt frame acquisition using the query and partial observations. FOCUS Zhu et al. (2025) allocates observations using estimated relevance and statistical uncertainty over temporal regions, while A.I.R. Zou et al. (2025) combines interval-level relevance, complexity, and temporal length with repeated VLM assessment. VCA Yang et al. (2025b) and VideoBrain Zou et al. (2026) further acquire evidence through iterative or learned exploration. CSES Zhang et al. (2026) explicitly models temporal coverage and acquires frames according to estimated marginal coverage gain. Our work takes a complementary temporal quantization view: under a fixed observation budget, the acquired frames form a finite temporal representation whose resolution is determined by the gaps between observations. `QUEST` weights these gaps by query relevance and prioritizes the largest one, yielding a one-step minimax rule for placing the next observation while preserving the downstream selection rule.

7 Conclusion

In this work, we introduced `QUEST`, a training-free method for adaptive candidate frame construction under a fixed observation budget. We formulate candidate construction as relevance-weighted temporal quantization, viewing the acquired frames as a finite temporal representation whose resolution depends on both query relevance and unresolved temporal gaps. Under fixed interval relevance estimates, this formulation yields a one-step minimax rule that selects the interval with the largest relevance-weighted gap and samples its midpoint. `QUEST` estimates interval relevance from partial observations and applies this rule sequentially while preserving the downstream frame-selection rule. Across five VLMs, four frame selectors, and five video understanding benchmarks, `QUEST` improves downstream accuracy over uniform candidate construction under matched observation and selection budgets, and remains competitive with dense 1-FPS and recent adaptive acquisition baselines. These results highlight that, beyond how candidate frames are selected, where candidate evidence is acquired along the video timeline is itself an important design choice for efficient video question answering.

AI USE STATEMENT

Generative AI tools were used to provide feedback on experimental design and to assist with literature search, including identifying potentially relevant prior work that may have been missed during manual review. The authors developed and verified the core research idea, methodology, experimental decisions, and final interpretation of the results. All AI-assisted suggestions were independently reviewed by the authors before being incorporated into the paper.

LIMITATION

While QUEST improves candidate frame construction, its performance depends on the quality of the relevance scorer and the initial candidate frames. Inaccurate relevance estimates or uninformative initial observations can lead to suboptimal allocation of the observation budget across temporal intervals. Additionally, although QUEST reuses the downstream selector's relevance scorer, its acquisition rule does not explicitly account for the selector's selection strategy. Future work could incorporate lightweight selector-aware acquisition rules while preserving plug-and-play compatibility. Finally, QUEST is designed for sequential acquisition under a limited observation budget; its integration with offline video preprocessing remains an avenue for future work.

REFERENCES

Xiang An, Yin Xie, Feilong Tang, Yunyao Yan, Huajie Tan, Didi Zhu, Changrui Chen, Xiuwei Zhao, Bin Qin, Kaicheng Yang, et al. Llava-onevision-2: Towards next-generation perceptual intelligence. *arXiv preprint arXiv:2605.25979*, 2026.

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. *CoRR*, abs/2502.13923, 2025b.

Shyamal Buch, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. Flexible frame selection for efficient video reasoning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29071–29082, 2025.

Guo Chen, Yicheng Liu, Yifei Huang, Yuping He, Baoqi Pei, Jilan Xu, Yali Wang, Tong Lu, and Limin Wang. Cg-bench: Clue-grounded question answering benchmark for long video understanding. *arXiv preprint arXiv:2412.12075*, 2024a.

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

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. *arXiv preprint arXiv:2603.00512*, 2026.

Chuanqi Cheng, Jian Guan, Wei Wu, and Rui Yan. Scaling video-language models to 10k frames via hierarchical differential distillation. *arXiv preprint arXiv:2504.02438*, 2025.

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

Bhavika Suresh Devnani, Jitesh Jain, Humphrey Shi, and Judy Hoffman. Coselect: Adaptive frame selection for video-language understanding. 2026.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24108–24118, 2025a.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. Framefusion: Combining similarity and importance for video token reduction on large vision language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22654–22663, 2025b.

Allen Gersho. Asymptotically optimal block quantization. *IEEE Transactions on information theory*, 25(4):373–380, 1979.

Robert M. Gray and David L. Neuhoff. Quantization. *IEEE transactions on information theory*, 44 (6):2325–2383, 1998.

Xiaohu Huang, Hao Zhou, and Kai Han. Prunevid: Visual token pruning for efficient video large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19959–19973, 2025.

Yuning Huang, Xiaoyu Ji, Joseph Huang, Yichi Zhang, and Fengqing Zhu. Adaptive greedy frame selection for long video understanding. *arXiv preprint arXiv:2603.20180*, 2026.

Jeongseok Hyun, Sukjun Hwang, Su Ho Han, Taeoh Kim, Inwoong Lee, Dongyoon Wee, Joon-Young Lee, Seon Joo Kim, and Minho Shim. Multi-granular spatio-temporal token merging for training-free acceleration of video llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23990–24000, 2025.

Marc Kesseböhmer and Sanguo Zhu. Quantization dimension via quantization numbers. 2003.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. Infinipot-v: Memory-constrained kv cache compression for streaming video understanding. *arXiv preprint arXiv:2506.15745*, 2025.

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

Jialuo Li, Bin Li, Jiahao Li, and Yan Lu. Divide, then ground: Adapting frame selection to query types for long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11369–11380, 2026a.

Yulin Li, Haokun Gui, Ziyang Fan, Junjie Wang, Bin Kang, Bin Chen, and Zhuotao Tian. Less is more, but where? dynamic token compression via llm-guided keyframe prior. *arXiv preprint arXiv:2512.06866*, 2025.

Zongyao Li, Kengo Ishida, Satoshi Yamazaki, Xiaotong Ji, and Jianquan Liu. Kfs-bench: Comprehensive evaluation of key frame sampling in long video understanding. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5643–5652. IEEE, 2026b.

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.

Shuming Liu, Chen Zhao, Tianqi Xu, and Bernard Ghanem. Bolt: Boost large vision-language model without training for long-form video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3318–3327, 2025a.

Xuyang Liu, Yiyu Wang, Junpeng Ma, and Linfeng Zhang. Video compression commander: Plug-and-play inference acceleration for video large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1910–1924, 2025b.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

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

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

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Holitom: Holistic token merging for fast video large language models. *arXiv preprint arXiv:2505.21334*, 2025.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. Fastvid: Dynamic density pruning for fast video large language models. *arXiv preprint arXiv:2503.11187*, 2025.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29118–29128, 2025.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Dycoke: Dynamic compression of tokens for fast video large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18992–19001, 2025.

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

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967. IEEE, 2025a.

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*, 2025b.

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.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025a.

Zeyuan Yang, Delin Chen, Xueyang Yu, Maohao Shen, and Chuang Gan. Vca: Video curious agent for long video understanding. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20168–20179. IEEE, 2025b.

Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xiaolei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, et al. Frame-voyager: Learning to query frames for video large language models. *arXiv preprint arXiv:2410.03226*, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

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.

Junyang Zhang, Puhan Luo, Chen Tang, Yuxi Shi, and Xiang-Yang Li. Coverage-driven adaptive keyframe selection for video understanding. *arXiv preprint arXiv:2608.00714*, 2026.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 881–916, 2025a.

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, 2025b.

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

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

Zirui Zhu, Hailun Xu, Yang Luo, Yong Liu, Kanchan Sarkar, Zhenheng Yang, and Yang You. Focus: Efficient keyframe selection for long video understanding. *arXiv preprint arXiv:2510.27280*, 2025.

Junbo Zou, Ziheng Huang, Shengjie Zhang, Liwen Zhang, and Weining Shen. Videobrain: Learning adaptive frame sampling for long video understanding. *arXiv preprint arXiv:2602.04094*, 2026.

Yuanhao Zou, Shengji Jin, Andong Deng, Youpeng Zhao, Jun Wang, and Chen Chen. Air: Enabling adaptive, iterative, and reasoning-based frame selection for video question answering. *arXiv preprint arXiv:2510.04428*, 2025.

Table 4: Model configurations for the baseline comparisons. Pixel limits denote configured preprocessing arguments, not measured output resolutions.

Model checkpoint	Visual input and preprocessing
Qwen3-VL-8B-Instruct	Video input, minimum/maximum pixels $200,704/1,605,632$, actual frame timestamps retained.
Qwen2.5-VL-7B-Instruct	Video input, the same pixel bounds, default processor temporal spacing.
LLaVA-Video-7B-Qwen2	Video input, default spatial pooling; no added timestamp instruction
LLaVA-OneVision-2-8B-Instruct	Image input for each frame with timestamp, minimum and maximum pixel arguments both set to $72,900$.
Qwen2-VL-7B-Instruct	Video input, minimum/maximum pixels $3,136/602,112$

A Experimental Details

A.1 Evaluation Protocol

We use the LongVideoBench validation split ($1,337$ questions), the MLVU development benchmark ($2,174$ questions), and the VideoMME test split ($2,700$ questions). LVBench contains $1,549$ evaluation questions. For CG-Bench-mini, we use a subset of 500 videos containing $1,331$ questions. We report question-level accuracy on LongVideoBench, VideoMME, and LVBench. For MLVU, we report the unweighted mean accuracy over its seven multiple-choice tasks.

Selected frames are supplied in chronological order. On VideoMME, we include only subtitle segments whose time intervals contain at least one selected frame. This rule is shared across methods, although the resulting subtitle text depends on the selected frames. We use greedy decoding with a maximum of 32 generated tokens on LongVideoBench and 16 on the other benchmarks. The random seed is 42. Videos are decoded with Decord, and experiments are conducted on NVIDIA H100 GPUs.

A.2 Model Configurations

Table 4 summarizes the model configurations used for the baseline comparisons. We use BF16 inference and FlashAttention-2, and pass the selected frames directly through each model's input processor and conversation template. For Qwen2.5-VL, the pixel bounds follow its LMMs-Eval Zhang et al. (2025a) implementation,[1] and we apply the same bounds to Qwen3-VL.

For LLaVA-OV2, the pixel budget follows the official MLVU evaluation setting of LMMs-Eval Zhang et al. (2025a). We retain this pixel budget for all dataset evaluations.

A.3 CG-Bench-mini Subsampling

We evaluate on a deterministic subset of CG-Bench-mini Chen et al. (2024a). We use this subset to reduce the evaluation cost. The annotation file is `cgbench_mini.json` from the Hugging Face dataset `CG-Bench/CG-Bench` at revision `a3be039f6ba85e022a749eca111b64ffcafadd1d`; it contains 3,000 questions over 1,118 videos. We sort the 1,118 unique `video_uid` values in ascending order of the hexadecimal SHA-256 digest of each UTF-8 encoded `video_uid`, and keep the first 500 videos. We retain every CG-Bench-mini question on these videos, which gives 1,331 questions. The ordering depends only on the video identifiers and was fixed before any model was evaluated, so the subset does not depend on the outcome of any experiment.

A.4 Frame Selection Settings

We construct $N = 128$ candidates and select $K = 32$ input frames, except for Qwen2-VL, which uses $K = 8$. The initialization size is $W = 17$ for LongVideoBench, VideoMME, and LVBench, and $W = 15$ for MLVU and CG-Bench-mini. Each candidate set is constructed using the corresponding

[1] `https://github.com/EvolvingLMMs-Lab/lmms-eval`

Table 5: Scorers and selection parameters. Resolutions refer to scorer inputs. BOLT uses $\alpha = 3.0$ on MLVU and VideoMME and $\alpha = 2.5$ on the remaining benchmarks.

Selector	Scorer	Selection parameters
Top-K	CLIP-L/14 (224)	Highest relevance scores.
BOLT	CLIP-L/14 (224)	Inverse-transform sampling with exponent α.
MDP^3	SigLIP-SO400M/14 (384)	$\lambda = 0.2$; segment size 32; condition size 1; kernel scales $\{1/8, 1/4, 1/2, 1, 2\}$.
Q-Frame	LongCLIP-L (336)	$\tau = 0.8$.

Table 6: Answer-format instructions used for evaluation. These are instruction excerpts, not complete prompt templates.

Benchmark	Answer instruction
LongVideoBench	`Answer with the option's letter from the given choices directly.`
MLVU	`Only give the best option.` `Best option: (`
VideoMME	`Respond with only the letter (A, B, C, or D) of the correct option.`
LVBench	`Answer the question with the option letter`
CG-Bench-mini	`Respond with only the letter of the correct option.` `Answer with the option's letter from the given choices directly.`

selector's scorer, as listed in Table 5. The scoring text contains the question and answer choices. The same selector configuration is used with uniform candidates and with `QUEST`.

For CSES Zhang et al. (2026), FOCUS Zhu et al. (2025), and AKS Tang et al. (2025), we compute frame relevance using BLIP-ITM-Large-COCO with the question, excluding answer choices, as text input. CSES uses $K_{\min} = 8$, $K_{\max} = 32$, and an observation limit of 128, retaining its adaptive stopping rule. For FOCUS, we use the official implementation. AKS uses candidates sampled at 1 FPS, a recursion depth of 5, and a threshold of 0.2 on LongVideoBench and 0.8 on the other evaluated benchmarks.

A.5 Evaluation Prompts

Each VLM receives the benchmark question and answer choices. The table lists the answer format instructions. For VideoMME with subtitles, the subtitle text precedes the question and choices, and the prompt ends with "The best answer is:". These evaluation prompts are separate from the text used for relevance scoring. We extract option letters using each benchmark's answer parser of LMMS-Eval Zhang et al. (2025a).

B Batched Acquisition

Proposition B.1 (Batch acquisition)**.** *On the frame-index grid, suppose* $D_n := r_n^{\max} - r_n^{\min} > 0$ *and order* $\mathcal{A}_n = \{I_1, \dots, I_s\}$ *by decreasing* $R_n(I)$*, and use the same deterministic tie rule in both sequential and batched acquisition. For* $I_i = [a_i, b_i]$*, define*

$$c_i := \frac{\lceil h_{I_i}/2 \rceil}{2} \exp\left(\frac{(\max(r_{a_i}, r_{b_i}) + r_n^{\max})/2 - r_n^{\min}}{D_n} \right).$$

If $2 \le k \le \min\{s, N - n\}$ *satisfies*

$$h_{I_j} \ge h_{I_\ell} \qquad (2 \le j \le k,\ j < \ell \le s),$$

$$\max_{i<k} \left\{ \frac{c_i}{R_n(I_k)}, \frac{\sqrt{e}\,\lceil h_{I_i}/2 \rceil}{h_{I_k}} \right\} < 1.$$

then $\mathrm{Mid}(I_1), \ldots, \mathrm{Mid}(I_k)$ *form a valid sequence of the next* k *acquisitions. Batching these acquisitions preserves the resulting sequential state and observation budget.*

Proof. As the observed score range increases, the log-priority ratio of two untouched intervals lies between its initial value and their log-length ratio. The length condition therefore keeps I_j maximal among the remaining original intervals. For any child C_i of an earlier split, its length and endpoint mean give, while I_k remains untouched,

$$\frac{R_{n+j-1}(C_i)}{R_{n+j-1}(I_k)} \le \max\left\{\frac{c_i}{R_n(I_k)}, \frac{\sqrt{e}\,\lceil h_{I_i}/2 \rceil}{h_{I_k}}\right\} < 1 \qquad (i < j \le k).$$

Thus no new child displaces I_j, and induction proves the claim. □

If no batch is certified, we retain the sequential acquisition step.

C PSEUDOCODE

Algorithm 1 summarizes QUEST. For initialization, we map W uniformly spaced target times to a reference grid of $M = 512$ frame indices. Constructing this grid uses only video metadata; only the selected frames are observed and scored. Subsequent acquisitions operate on the native frame indices. Let $t_i = (i-1)/f$ for frame rate f, and let rint denote nearest-integer rounding with ties to even. All selection ties favor the smaller frame index, using the proposed midpoint index to break ties between intervals.

Algorithm 1: QUEST candidate frame construction

Input: Video $V = \{v_i\}_{i=1}^{T}$, query q, frame rate f, scorer Rel, budgets N and W, reference size $M = 512$; $2 \le W \le \min\{N, M\}$ and $N \le T$.
Output: Temporally ordered candidate set S_N and cached raw relevance scores.

$G \leftarrow \left\{1 + \mathrm{rint}\left(\frac{j(T-1)}{M-1}\right) : j = 0, \ldots, M-1\right\}$
$S_W \leftarrow \varnothing$
for $j = 0, \ldots, W-1$ **do**
  $\tau_j \leftarrow t_1 + \frac{j}{W-1}(t_T - t_1)$
  $i \leftarrow \arg\min_{k \in G:\, v_k \notin S_W} |t_k - \tau_j|$
  $S_W \leftarrow S_W \cup \{v_i\}$
end
Compute and cache $r_i \leftarrow \mathrm{Rel}(v_i, q)$ for each $v_i \in S_W$
for $n = W, \ldots, N-1$ **do**
  $r_n^{\min} \leftarrow \min_{v_i \in S_n} r_i$, $r_n^{\max} \leftarrow \max_{v_i \in S_n} r_i$
  $\mathcal{A}_n \leftarrow \{[a,b] : v_a, v_b \text{ are consecutive in } S_n,\ b - a \ge 2\}$
  foreach $I = [a,b] \in \mathcal{A}_n$ **do**
    $h_I \leftarrow b - a$
    $u_n(I) \leftarrow \dfrac{(r_a + r_b)/2 - r_n^{\min}}{r_n^{\max} - r_n^{\min}}$
    $c(I) \leftarrow \arg\min_{a<i<b} \left|t_i - \frac{t_a + t_b}{2}\right|$
  end
  $I_n^\star \leftarrow \arg\max_{I \in \mathcal{A}_n} \left(\log h_I + u_n(I)\right)$
  $c \leftarrow c(I_n^\star)$
  Compute and cache $r_c \leftarrow \mathrm{Rel}(v_c, q)$
  $S_{n+1} \leftarrow S_n \cup \{v_c\}$
end
return S_N *in chronological order and its cached scores* $\{r_i : v_i \in S_N\}$

The log-domain priority $\log h_I + u_n(I)$ is equivalent to maximizing $R_n(I) = \frac{1}{2} h_I \exp(u_n(I))$. The downstream selector uses the cached raw scores without the normalization used for candidate construction.

D CANDIDATE POOL QUALITY

We evaluate candidate pools directly against annotated visual evidence before downstream frame selection. Each pool contains $N = 128$ frames, with the initialization and scoring settings in Appendix A.4. We compare each scorer-specific QUEST pool with the same uniform pool. Bold denotes improvements over Uniform; parentheses show differences computed before rounding.

KFS-Bench. We use the KFS-Bench Li et al. (2026b) annotations for $1,291$ LongVideoBench questions and 888 VideoMME questions. We report the Unified Keyframe Sampling Score (UKSS) using the official evaluator. UKSS combines the fraction of relevant candidate frames, scene coverage under scene-specific frame-count thresholds, and the balance of frame allocation across annotated relevant scenes. We report $100 \times$ UKSS, with higher values indicating better overall sampling quality. As shown in Table 7, QUEST achieves higher UKSS than uniform sampling with all three scorers on both benchmarks.

Table 7: UKSS ($\times 100$) on KFS-Bench.

Candidate pool	Scorer	LongVideoBench	VideoMME
Uniform	–	19.28	32.48
QUEST	CLIP-L/14	**19.52** (+0.24)	**33.03** (+0.55)
QUEST	SigLIP-SO400M	**19.44** (+0.16)	**33.13** (+0.65)
QUEST	LongCLIP-L (336)	**19.78** (+0.50)	**33.42** (+0.95)

CG-Bench-mini. We use the annotated clue intervals from CG-Bench Chen et al. (2024a) for the $1,331$ questions in our subset (Appendix A.3). We measure the clue-hit rate, defined as the percentage of questions for which at least one candidate frame falls within an annotated clue interval. The rate ranges from 0% to 100%, with higher values indicating that more questions retain at least one annotated clue. Table 8 shows that QUEST increases this rate by 0.68–1.43 percentage points across the three scorers, corresponding to 9–19 additional questions with at least one clue captured.

Table 8: Clue-hit rate (%) on CG-Bench-mini.

Candidate pool	Scorer	Clue-hit
Uniform	–	69.95
QUEST	CLIP-L/14	**70.62** (+0.68)
QUEST	SigLIP-SO400M	**71.37** (+1.43)
QUEST	LongCLIP-L (336)	**70.85** (+0.90)

E ABLATION

We examine the contributions of temporal gaps and query relevance using Qwen2.5-VL with Top-K selection and CLIP-L/14 scoring on MLVU and LVBench with $N = 128$, $K = 32$. All acquisition variants use the same initialization and midpoint refinement as QUEST.

Length only removes relevance weighting, whereas *Relevance only* removes the temporal-gap factor. We also evaluate *Raw max endpoint*, which replaces the normalized endpoint mean $u_n(I)$ with the raw endpoint maximum $\max(r_a, r_b)$. This variant changes both the endpoint estimator and its normalization. Table 9 lists the resulting acquisition priorities and accuracies.

As shown in Table 9, QUEST achieves the highest accuracy on both benchmarks. Removing the temporal-gap factor causes the largest reduction, by 8.15 and 2.07 percentage points on MLVU and LVBench, respectively. Removing relevance weighting also reduces accuracy by 3.13 and 1.16 points, while using the raw endpoint maximum reduces accuracy by 1.00 and 0.71 points. These results support jointly accounting for temporal spacing and query relevance when allocating a fixed observation budget.

Table 9: Acquisition ablation with Qwen2.5-VL and Top-K. Values are accuracies (%); bold denotes the best result.

Variant	Acquisition priority	MLVU	LVBench
Uniform	–	67.80	41.31
Length only	h_I	67.24	42.03
Relevance only	$\exp(u_n(I))$	62.22	41.12
Raw max endpoint	$h_I \exp(\max(r_a, r_b))$	69.37	42.48
QUEST	$h_I \exp(u_n(I))$	**70.37**	**43.19**