CUEKFS: AGENTIC CUE-DRIVEN KEYFRAME SELECTION FOR LONG VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Keyframe selection (KFS) has long produced compact video summaries for browsing and retrieval, and representative frames for thumbnails. More recently, when conditioned on a question, KFS provides an alternative to uniform sampling for long-video question answering by selecting frames that are more relevant to the question. Most methods rank frames by similarity to the question. Yet a relevant frame may score poorly when the question combines subjects or moments that no single frame shows, or requires implicit information absent from its wording. Other methods try to break down the question into subqueries, but suffer from inaccurate decomposition due to their static initial context. Therefore, we propose CUEKFS, a training-free method that reformulates question–frame matching as comparing frames against a set of dynamically generated visual cues. From an initial set of salient frames, we decompose the question into cues. Each cue concurrently probes the video to navigate to its own evidence. A reasoning VLM then agentically revises the cue set against its evidence to re-explore the video. CUEKFS then allocates the budget across the surviving cues. Across three benchmarks, CUEKFS establishes state-of-the-art results in all 27 evaluated settings with available prior results, achieving budget-averaged gains of up to +4.54% over the previous baseline and a median of only two VLM calls. We further provide a detailed behavioral analysis of CUEKFS, showing that agentic cue refinement drives active re-exploration of the video, yielding relative similarity gains of up to 92% over the initial context.

1 INTRODUCTION

Keyframe selection (KFS), previously studied as video summarization, picks a small set of representative frames from a video and has long been valuable in summarization, retrieval, and thumbnail selection (Gygli et al., 2014; Narasimhan et al., 2021; Zhang, 2025). More recently, KFS has been adopted as an upstream alternative to uniform sampling in video question answering. By tailoring the input frames to the question, it increases their relevance with the same number of frames or reduces this number when possible (Tang et al., 2025; Liu et al., 2025; Chen et al., 2026).

Most methods treat KFS as question–frame matching: they score frames by embedding similarity to the question and select from that signal (Radford et al., 2021; Tschannen et al., 2025; Tang et al., 2025; Sun et al., 2025; Liu et al., 2025; Zhang et al., 2025; Chen et al., 2026; Liang et al., 2024). This primarily rewards surface similarity, yet such similarity alone is insufficient to determine frame relevance: a question may span several subjects or moments that no single frame captures in full, concern causal or temporal relations that no single frame can show, or rely on implicit information, such as genre or intent, that the question's wording does not name. Other methods decompose the question into parts (Tan et al., 2026; Ben-Ami et al., 2026; Shlapentokh-Rothman et al., 2026; Guo et al., 2025; Ye et al., 2025) from the question alone or a coarse video preview, leaving much of the video unseen before retrieval targets are defined. Therefore, subsequent retrieval may refine the regions to examine but remains constrained by a potentially biased initial decomposition.

An embedding model can score each frame efficiently but has limited reasoning capacity. Instead, a VLM offers stronger reasoning but is costly to apply across an entire video, as in dense captioning. We therefore propose CUEKFS, a training-free KFS method that uses the embedding model for video-wide retrieval and the reasoning VLM to dynamically revise what should be retrieved

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.

next. Specifically, a VLM first decomposes the question into visual cues based on a compact video overview, each cue a short self-contained phrase describing one concrete visual element. The embedding model retrieves evidence for each cue across the video, and the VLM agentically revises the cues guided by that evidence. The revised cues then perform another video-wide retrieval, after which CUEKFS allocates the k-frame budget across the surviving cues. We design *peaks* and *plateaus* to surface that evidence, and *Batched Agentic Refinement* to revise the cues efficiently.

Our contributions are threefold. (i) We propose CUEKFS, a method that turns a reasoning VLM with an embedding model into an agentic explorer of long video, matching dynamic cues against all frames to retrieve evidence. (ii) Across three benchmarks, three downstream VLMs, and three frame budgets, CUEKFS establishes state-of-the-art results in all 27 settings with available prior results, achieving budget-averaged gains of up to +4.54% over the previous baseline and a median of only 2 reasoning-VLM calls per selection. (iii) We further provide behavioral analysis of our method, showing that CUEKFS actively explores the video beyond its initial context: on long Video-MME videos, final cues achieve up to a 92% higher match score on frames outside the initial context.

2 RELATED WORK

2.1 SELECTION BY QUESTION RELEVANCE

Most methods score each candidate frame by its similarity to the whole question and add structure to that single signal: AKS balances relevance and coverage (Tang et al., 2025), MDP^3 adds diversity and sequentiality (Sun et al., 2025), BOLT applies inverse transform sampling over the similarity distribution (Liu et al., 2025), Q-FRAME adds multi-resolution adaptation (Zhang et al., 2025), and WFS-SB (Chen et al., 2026) denoises the signal to detect semantic boundaries. Others keep the single question-level signal but condition it, routing by a coarse query type (DIG (Li et al., 2026b)) or weighting fixed modality streams by query intent (Q-GATE (Wang et al., 2026a)). A.I.R. (Zou et al., 2026) keeps the same signal but adds a VLM that verifies its top-ranked frames and densifies the sampling around the ones it confirms. Most of these rank a frame by a single question-level score, an imperfect proxy for its relevance. Some methods also use question–frame similarity in data construction, training, or scoring (Yu et al., 2025; Li et al., 2026a; Yao et al., 2025; Wang et al., 2026b), and are thus contaminated by this proxy. CUEKFS instead decouples the question into cues and scores each frame against every cue, so a frame relevant to any one cue still surfaces.

2.2 SELECTION BY QUESTION DECOMPOSITION

Other methods decompose the question into multiple parts before matching. MSJOE reasons out several queries from a coarse preview but merges them into one similarity matrix consumed by a trained sampler (Tan et al., 2026), HIMU parses the question text alone into a hierarchical logic tree of atomic predicates composed into one satisfaction curve (Ben-Ami et al., 2026), and TOOLMERGE decomposes it into tool calls merged into one ranking (Shlapentokh-Rothman et al., 2026). LIF (Guo et al., 2025) extracts objects and typed relations for search with a frozen detector. T* (Ye et al., 2025) decomposes it into target and cue objects from a coarse glance, then iteratively refines where a frozen detector searches for them, though found targets can be removed. In all of these, target descriptions are formed from the initial decomposition. Subsequent search may update where to look or remove found targets, but does not update the targets. CUEKFS instead lets each cue evolve against its own retrieved evidence, so the cue set explores the video dynamically rather than remaining bounded by the initial decomposition.

2.3 EVALUATING KEYFRAME SELECTION

Keyframe selection is evaluated indirectly: the selected frames are supplied to a downstream VLM, and its QA accuracy on video benchmarks (Video-MME (Fu et al., 2025), MLVU (Zhou et al., 2025), LongVideoBench (Wu et al., 2024)) is reported as the selection's score. Studies adopt common downstream VLMs with modest capacity for comparability, but the score then conflates selection quality with the VLM's own capability: a weak VLM under-rewards a correct selection when it cannot answer even from sufficient frames. We retain this protocol but add a strong downstream VLM (GPT-5.5) to reduce the influence of the downstream VLM's limitations on our evaluation, and

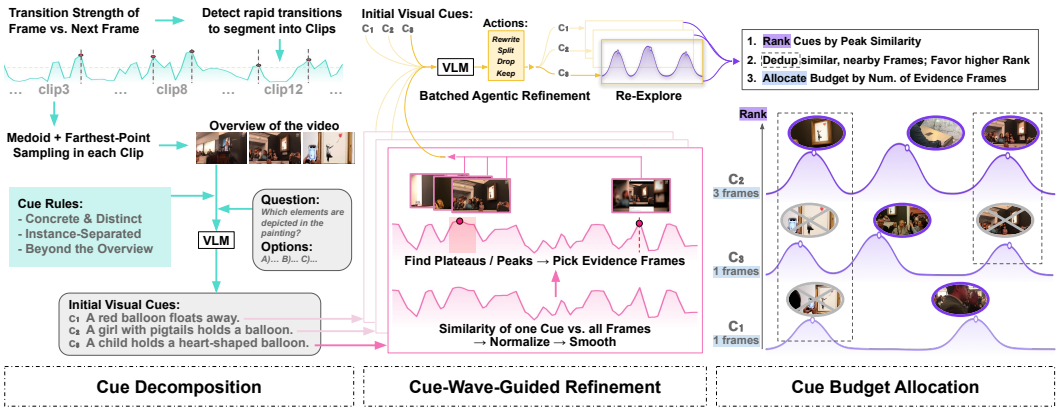

Figure 1: Overview of CUEKFS. Cue Decomposition builds a clip-wise video overview and applies three rules to produce initial visual cues. Cue-Wave-Guided Refinement computes one wave per cue, samples evidence from plateaus and peaks, refines cues in batches with a reasoning VLM, and re-explores the video. Cue Budget Allocation ranks cues by peak similarity, removes similar nearby frames in favor of higher-ranked cues, and allocates the budget by remaining evidence-frame counts. The displayed cue set and frame counts are illustrative.

report accuracy averaged across frame budgets: small budgets keep even a strong VLM sensitive to missing or redundant frames, while averaging reduces dependence on any single budget.

3 METHOD

Given a video V, a question (or more generally, a task context) q, and a frame budget k, CUEKFS returns at most k frames for a downstream VLM. We sample T candidate frames, $\mathcal{F} = \{f_1, \ldots, f_T\}$, indexed in temporal order, and use a frozen embedding model. Its image encoder gives the frame embeddings $\mathbf{e}_t = \phi(f_t)$, while its text encoder embeds a textual cue c as $\psi(c)$. As illustrated in Figure 1, the method comprises Cue Decomposition (§3.1), Cue-Wave-Guided Refinement (§3.2), and Cue Budget Allocation (§3.3). Section 4.1 and Table 3 specify the experimental configurations.

3.1 CUE DECOMPOSITION

As shown in the left of Figure 1, rapid inter-frame transitions divide the video into clips, clip representatives form a compact overview, and a VLM decomposes the question into initial visual cues.

Overview of the video. We first form the question-agnostic inter-frame signal

$$a_t = \cos(\mathbf{e}_t, \mathbf{e}_{t+1}), \qquad t = 1, \ldots, T-1. \tag{1}$$

Similar to Chen et al. (2026), our segmentation uses a wavelet transform, but applies it to adjacent-frame similarity a rather than question-conditioned frame relevance. Peaks in the resulting transition-strength signal define clip boundaries. This avoids propagating errors from matching the whole question directly to individual frames into the overview. Within each clip, we first select the medoid and then use it to start farthest-point sampling, yielding complementary representatives for the overview R under a frame budget N_{repr}. Appendix A gives the transition and sampling details.

Initial visual cues. A reasoning VLM receives R, q, and candidate answers when available, together with cue-construction rules. As shown in Figure 1 (left), the rules cover three principles: *Concrete and Distinct*, *Instance-Separated*, and *Beyond the Overview*. First, each cue should be a short, self-contained description of a concrete visual element that a single frame could show, be matchable by appearance, and remain distinct from the other cues. Second, distinct instances, events, or stages in counting, ordering, before–after, and change-over-time questions are written as separate cues, making each target independently retrievable. Third, the VLM may formulate question-relevant cues even when the related frames are absent from the overview, so R conditions

```
For EACH cue, choose one action:
- drop:    the cue cannot help with the context.
- keep:    the cue is useful for the context; keep it unchanged.
- rewrite: the cue is useful but imprecise or retrieving the wrong thing; give
           one better cue, still faithful to the context.
- split:   the cue bundles two unrelated visual things; give two or more
           specific cues.
```

Figure 2: Evidence-grounded cue-refinement actions. For each cue, the reasoning VLM chooses one of four actions based on that cue's wave-retrieved evidence.

but does not bound what the cues may seek in the initial overview. The VLM then returns the initial visual cue set $\mathcal{C}^0 = \{c_1, \ldots, c_m\}$. Appendix Figure 4 gives the cue-construction prompt.

3.2 Cue-Wave-Guided Refinement

As shown in the middle of Figure 1, the initial cues probe all the frames in parallel, evidence frames sampled from each cue's plateaus and peaks guide batched agentic refinement, and the refined cues re-explore the full video.

Per-cue similarity waves. For every initial cue $c \in \mathcal{C}^0$, we compare it with every candidate frame to obtain a raw similarity signal, then robustly normalize and smooth the signal into a cue wave:

$$s_t^c = \cos\big(\mathbf{e}_t, \psi(c)\big), \qquad w_t^c = \mathrm{smooth}_h\left(\frac{s_t^c - \mathrm{med}(s^c)}{\mathrm{p}_{95}(s^c) - \mathrm{p}_5(s^c) + \varepsilon}\right). \tag{2}$$

Here med and p_r denote the median and the rth percentile, and smooth_h is a moving average of width h. $\varepsilon > 0$ prevents division by zero. Each w^c is normalized separately to locate evidence within its cue; raw s^c values are retained for cross-cue comparison in Cue Budget Allocation.

Plateaus, peaks, and evidence frames. Each cue wave provides two complementary forms of evidence. Plateaus capture sustained matches, such as an extended scene, whereas peaks capture brief, isolated matches, such as a momentary action. We first identify maximal above-threshold intervals $[a, b]$ satisfying

$$w_u^c \geq Q_\rho(w^c) \quad \forall u \in [a, b], \tag{3}$$

where $Q_\rho(w^c)$ is the ρth percentile of that cue's wave. After merging nearby intervals and discarding those shorter than ℓ_0 frames, we select each plateau's medoid and use it to start farthest-point sampling within the interval, yielding up to n_{pl} plateau samples. A peak candidate is a local maximum t satisfying $w_{t-1}^c < w_t^c \geq w_{t+1}^c$. Its prominence measures its rise above its local surroundings:

$$\pi_t^c = w_t^c - \max\left(\min_{u \in [t-\delta, t]} w_u^c,\ \min_{u \in [t, t+\delta]} w_u^c\right). \tag{4}$$

We first suppress candidates separated by fewer than d_p frames in descending peak height, discard the remaining candidates below the prominence threshold π_0, and retain at most n_p peaks with the highest wave values. Each retained peak contributes its center frame. The retained peak centers and plateau samples together form the cue-specific evidence set $\mathcal{E}_c$. We assign each evidence frame a score: w_t^c for a peak center and the plateau's mean wave value for every plateau sample. These evidence scores prioritize frames for refinement and later rank frames within each cue; the selected refinement images are presented in temporal order within each cue. Table 3 and Appendix A give the thresholds and sampling settings.

Batched agentic refinement. Since initial cues are formed before they explore the video, they may be imprecise, bundle distinct targets, or contribute no useful evidence. Therefore, we send each cue and its evidence $\mathcal{E}_c$ obtained from the previous exploration to a reasoning VLM to produce a more accurate cue for further search. Reviewing cues individually would multiply expensive VLM calls, so we process them in batches of up to B cues sharing an evidence-image budget of M, pairing each cue with its own evidence frames. For each cue within a batch, the VLM independently

keeps it unchanged when useful, rewrites it when imprecise or retrieving the wrong thing, splits it when it bundles distinct targets, or drops it when it provides no useful evidence. Figure 2 shows the corresponding prompt. Kept cues and the outputs of rewrite and split form the refined cue set $\mathcal{C}^1$, while dropped cues are excluded. Appendix Figure 5 gives the exact batched refinement prompt.

Re-explore. Every rewritten or split cue in $\mathcal{C}^1$ re-explores all T candidate frames and follows the same wave construction and evidence extraction. Kept cues retain their existing waves and evidence. The resulting evidence frames are candidates for Cue Budget Allocation, retrieved by dynamically refined cues rather than a static context.

3.3 Cue Budget Allocation

As shown in the right panel of Figure 1, Cue Budget Allocation ranks the refined cues by peak similarity, removes frames only when they are visually similar and temporally nearby, and allocates the frame budget according to each cue's remaining evidence count.

Rank cues by peak similarity. For each refined cue $c \in \mathcal{C}^1$, we define its peak similarity as

$$\iota(c) = \max_{1 \le t \le T} s_t^c. \tag{5}$$

Unlike the separately normalized waves w^c, the raw values s^c are comparable across cues. Sorting by $\iota(c)$ resolves shared evidence, prioritizes cues when the budget cannot cover them all, and breaks integer-allocation ties; proportional targets depend on evidence counts.

Deduplicate similar, nearby frames. To avoid spending multiple slots on the same evidence, we process all retrieved evidence frames from higher to lower cue rank, ordering frames within each cue by evidence score. Two evidence frames f_i and f_j, occurring at times t_i and t_j, are considered redundant only when

$$\cos(\mathbf{e}_i, \mathbf{e}_j) \ge \tau_{\cos} \quad \text{and} \quad |t_i - t_j| \le \tau_{\Delta}. \tag{6}$$

Requiring both conditions preserves repeated-looking events at distant times and distinct events that occur close together. A frame is retained only if it is not redundant with any previously retained frame. When redundant frames come from different cues, the frame retrieved by the higher-ranked cue is retained. This leaves no redundant pairs among the evidence frames passed to allocation.

Allocate budget by evidence count. After de-duplication, we divide the frame budget k among cues in proportion to the number of evidence frames each retains. When the budget permits, we first assign one slot to each cue to preserve coverage, then assign each remaining slot to the cue with the largest shortfall from its proportional target for the total budget, subject to its available evidence count. If cue c receives k_c slots, we select its k_c highest-scoring retained evidence frames. Appendix A gives more details. The final frames are returned in temporal order.

4 Experiments

4.1 Experimental setup

Evaluation configuration. Following the standard evaluation protocol, a KFS method receives a video and a multiple-choice question, and returns k frames. A frozen downstream VLM then answers the same question using only the selected frames. We report accuracy on Video-MME (Fu et al., 2025), MLVU (Zhou et al., 2025), and LongVideoBench (Wu et al., 2024), without subtitles. Candidate frames are sampled at 1 fps with budgets $k \in \{8, 16, 32\}$. To reduce sensitivity to the choice of frame budget, we also report the mean across the three budgets, denoted by "Avg." Since VideoQA accuracy depends on both the selected frames and the downstream VLM (§2.3), we use GPT-5.5 (OpenAI, 2026) as a downstream VLM. Table 1 also reports results with Qwen2.5-VL-7B (Bai et al., 2025) and InternVL3-8B (Zhu et al., 2025) for comparison with prior KFS work.

CueKFS configuration. We use the frozen SigLIP2-giant embedding model (Tschannen et al., 2025). For reproducibility, we use the open source Qwen3.6-27B (Qwen Team, 2026) as the reasoning VLM for the ablation studies in Table 2 and the analysis of cue behavior in Section 4.4. The full

Table 1: Accuracy (%) on three long-video QA benchmarks at three frame budgets with three downstream VLMs. Bold denotes the best result in each block. For Qwen2.5-VL and InternVL3, baseline results at k=32 are from Chen et al. (2026).

Downstream VLM	KFS method	Video-MME				MLVU				LongVideoBench			
		k=8	k=16	k=32	Avg	k=8	k=16	k=32	Avg	k=8	k=16	k=32	Avg
GPT-5.5	Uniform	73.70	77.11	81.41	77.41	61.73	67.74	71.90	67.12	63.95	67.46	70.38	67.26
	WFS-SB (Chen et al., 2026)	76.22	78.74	80.96	78.64	76.13	80.36	82.11	79.53	69.78	73.90	76.66	73.45
	CUEKFS	**80.67**	**83.30**	**84.15**	**82.70**	**80.08**	**82.29**	**83.49**	**81.95**	**75.39**	**77.26**	**79.88**	**77.51**
Qwen2.5-VL	Uniform	53.04	57.56	60.00	56.87	54.14	58.37	61.13	57.88	52.80	56.32	58.71	55.94
	AKS (Tang et al., 2025)	–	–	64.0	–	–	–	67.2	–	–	–	63.2	–
	MDP^3 (Sun et al., 2025)	–	–	63.8	–	–	–	66.2	–	–	–	60.0	–
	A.I.R. (Zou et al., 2026)	–	–	65.0	–	–	–	67.5	–	–	–	61.4	–
	WFS-SB (Chen et al., 2026)	58.37	61.30	64.40	61.36	66.05	68.35	70.40	68.27	59.91	63.35	64.40	62.55
	CUEKFS	**64.85**	**66.26**	**66.59**	**65.90**	**69.55**	**71.57**	**71.48**	**70.87**	**62.30**	**64.70**	**65.00**	**64.00**
InternVL3	Uniform	59.41	62.56	64.44	62.14	64.17	67.07	69.23	66.82	52.95	56.25	59.01	56.07
	AKS (Tang et al., 2025)	–	–	66.3	–	–	–	74.2	–	–	–	61.5	–
	MDP^3 (Sun et al., 2025)	–	–	66.8	–	–	–	74.0	–	–	–	60.9	–
	A.I.R. (Zou et al., 2026)	–	–	68.2	–	–	–	74.5	–	–	–	62.8	–
	WFS-SB (Chen et al., 2026)	64.37	65.48	67.40	65.75	73.64	75.62	74.80	74.69	59.76	60.81	62.90	61.16
	CUEKFS	**68.11**	**69.26**	**69.19**	**68.85**	**76.54**	**77.14**	**77.64**	**77.11**	**62.83**	**64.17**	**64.62**	**63.87**

Table 2: Step-by-step ablations on Video-MME with GPT-5.5 as the downstream VLM.

Configuration	k=8	k=16	k=32	Avg	VLM calls (median)
Uniform sampling	73.70	77.11	81.41	77.41	0
Add Overview segmentation with per-clip question matching	75.59	77.26	80.63	77.83	0
Add Cue Decomposition & Cue Budget Allocation	76.33	79.56	81.89	79.26	1
↳ Remove Cue Budget Allocation	71.44	74.81	78.44	74.90	1
Add Per-cue refinement using cue-wave evidence	78.63	81.19	83.04	80.95	11
↳ Change to Batched refinement (Cue-Wave-Guided Refinement)	**79.30**	**80.63**	**83.19**	**81.04**	↳**2**
↳ Remove Cue Budget Allocation	74.85	79.50	81.63	78.66	2
↳ Remove cue-wave evidence	75.81	77.85	80.63	78.10	2

CUEKFS configuration for these experiments has 24 overview frames, batches of 5 cues sharing 24 evidence frames, and $(\tau_{\cos}, \tau_{\Delta}) = (0.95, 2\text{s})$. For the results in Table 1, we use GPT-5.5 as the reasoning VLM. Since CUEKFS has several configurable stages, Table 3 examines how the reasoning VLM and hyperparameter settings affect performance and reports the configuration used in Table 1. Appendix A provides additional implementation details.

4.2 MAIN RESULTS

As shown in Table 1, CUEKFS achieves state-of-the-art accuracy in all 27 settings. With GPT-5.5 as the downstream VLM, its budget-average gains over WFS-SB (Chen et al., 2026) on Video-MME, MLVU, and LongVideoBench are +4.06, +2.42, and +4.06. The corresponding gains over WFS-SB are +4.54, +2.60, and +1.45 with Qwen2.5-VL, and +3.10, +2.42, and +2.71 with InternVL3. With GPT-5.5, CUEKFS also beats uniform sampling by +5.29, +14.83, and +10.25. At k=32, CUEKFS outperforms AKS (Tang et al., 2025), MDP^3 (Sun et al., 2025), and A.I.R. (Zou et al., 2026) on every benchmark.

4.3 ABLATION STUDIES

Table 2 builds CUEKFS step by step. We replace uniform sampling with overview segmentation, selecting the frame with the highest question–frame similarity within each clip. This segmentation baseline raises the average accuracy by 0.42 points. Adding Cue Decomposition and Cue Budget Allocation brings a further 1.43 point gain. Removing Cue Budget Allocation from this stage instead lowers the average by 4.36 points, demonstrating the importance of combining decomposition with allocation. We then add per-cue refinement grounded in cue-wave evidence, reviewing one cue per call with an average of 35.1 evidence frames. This raises the average from 79.26 to 80.95, a gain

Table 3: Hyperparameter ablations on Video-MME. Values and their corresponding average accuracies are listed in the same order. Bold marks the configuration used in Table 1 and its accuracy.

Hyperparameter	Values evaluated	Avg
Reasoning VLM	Qwen3.6-27B / **GPT-5.5**	81.04 / **81.52**
Overview frames N_{repr}	16 / 24 / 32 / **48** / 64	80.50 / 81.52 / 81.32 / **82.35** / 82.35
Cues per batch B	3 / 5 / 10 / **20** / all	82.06 / 82.35 / 82.30 / **82.47** / 82.42
Review image budget M	16 / 24 / **32** / 48 / 64	82.42 / 82.47 / **82.55** / 82.20 / 82.13
VLM frame edge (px)	**448** / 768 / 1024 / 1536	**82.55** / 81.75 / 81.95 / 82.38
Wave smoothing window h	1 / **3** / 5	82.29 / **82.55** / 81.29
Peak prominence and plateau percentile	$(0.15, 70)$ / $(\mathbf{0.25}, \mathbf{80})$ / $(0.35, 90)$	82.21 / **82.55** / 82.38
Deduplication thresholds $(\tau_{\cos}, \tau_{\Delta}\ \text{s})$	$(0.95, 2)$ / $(0.90, 1)$ / $(0.90, 3)$ $(\mathbf{0.98}, \mathbf{1})$ / $(0.98, 3)$	82.55 / 82.35 / 82.27 **82.70** / 82.67

Table 4: Refinement behavior on Video-MME. Panel (a) reports action frequencies by video length and overall. Counts are averaged over questions where the action occurs. Panel (b) reports refinement gains for overlapping question groups defined by whether each action occurred.

(a) How often is each agentic refinement action used?

Action	Short	Medium	Long	Overall	Count per question
Keep	96%	92%	96%	94%	3.4
Rewrite	60%	74%	85%	73%	2.0
Split	19%	30%	49%	33%	1.7
Drop	63%	69%	80%	71%	3.9

(b) How much does refinement help each action group?

Question group	Δ at k=8	Δ at k=32
Dropped a cue	+4.2	+1.9
Rewrote a cue	+3.7	+2.1
Split a cue	+5.3	+3.6

of 1.69 points, but increases the median number of VLM calls to 11. In the batched-refinement ablation, Cue-Wave-Guided Refinement uses a batch size of B=5 and shows an average of 5.6 evidence frames per cue, keeping the visual context per call roughly comparable. It maintains 81.04 accuracy while reducing median VLM calls from 11 to 2. We use batched refinement in the remaining experiments. Removing Cue Budget Allocation from the batched-refinement configuration in Table 2 reduces 2.38 points, while removing cue-wave evidence reduces it by 2.94 points. These controlled comparisons further validate both Cue Budget Allocation and the plateau-and-peak evidence used in Cue-Wave-Guided Refinement.

Table 3 examines how the reasoning VLM and hyperparameters affect Video-MME performance. The pipeline has multiple stages and hyperparameters, making exhaustive evaluation of their joint configuration space impractical. We therefore first examine the reasoning VLM and then vary one factor at a time across the three pipeline stages, from Cue Decomposition to Cue-Wave-Guided Refinement and finally Cue Budget Allocation. The configuration used in Table 1 uses GPT-5.5 as the reasoning VLM, $N_{\text{repr}} = 48$, $B = 20$, $M = 32$, a 448 px maximum image edge for the reasoning VLM, $h = 3$, $(\pi_0, \rho) = (0.25, 80)$, and $(\tau_{\cos}, \tau_{\Delta}) = (0.98, 1\,\text{s})$. It yields 82.70 average accuracy on Video-MME with GPT-5.5 as the downstream VLM.

4.4 Analysis of cue behavior

Refinement is active and most effective at low budgets. Table 4(a) shows that refinement rewrites at least one cue on 73% of questions, drops one on 71%, and splits one on 33%. All three actions become more frequent as video length increases. This trend suggests that the initial context is less reliable for longer videos, making refinement more important for re-exploration. Panel (b) shows positive refinement gains in all three action groups at both reported budgets. The gains are consistently larger at k=8 than at k=32, showing that refinement is especially important when the frame budget is tight.

Table 5: SigLIP2 score on Video-MME. For each cue set, we average per-cue maximum scores within each question and report the median across questions. The Question row reports the median maximum score for the question text. Candidate frames are 1-fps video frames ("Full video") or the overview frames from Cue Decomposition ("Overview"). Final cues comprise kept initial cues, rewritten cues, and split cues. Since Overview is a subset of Full video, the relative gain Full/Overview-1 is nonnegative. Larger gains indicate stronger matches beyond the overview.

Retrieval text	Comparison	Short	Medium	Long
Question	Full video	0.34	0.79	0.77
Initial cues	Full video	0.89	0.95	0.96
	Overview	0.85	0.81	0.67
	Full/Overview − 1	*0.05*	*0.17*	*0.43*
Final cues	Full video	0.93	0.98	0.98
	Overview	0.87	0.82	0.64
	Full/Overview − 1	*0.07*	*0.20*	*0.53*
Rewritten cues	Full video	0.98	0.99	1.00
	Overview	0.93	0.88	0.69
	Full/Overview − 1	*0.05*	*0.13*	*0.45*
Split cues	Full video	0.92	0.98	0.98
	Overview	0.80	0.71	0.51
	Full/Overview − 1	*0.15*	*0.38*	*0.92*

Table 6: Cue coverage on Video-MME. A frame is covered if any cue has a SigLIP2 score of at least 0.5. Panel (a) reports median Jaccard overlap before and after refinement. Panel (b) reports QA accuracy (%) with GPT-5.5 as the downstream VLM, given frames selected by CUEKFS, sampled outside CUEKFS's final cue coverage, or sampled uniformly. Δ is the overall accuracy difference from CUEKFS.

(a) Coverage overlap	Short	Medium	Long	All
Initial cues vs. Final cues	0.89	0.74	0.65	0.76
Initial cues vs. Rewritten cues	0.49	0.41	0.36	0.39
Initial cues vs. Split cues	0.52	0.39	0.31	0.37

	(b) QA accuracy by frame source	Short	Medium	Long	All	Δ
k=8	CUEKFS	85.56	77.56	74.78	79.30	–
	Outside coverage	72.78	69.00	67.22	69.67	−9.63
	Uniform sampling	79.89	71.67	69.56	73.70	−5.59
k=16	CUEKFS	87.33	79.11	75.44	80.63	–
	Outside coverage	75.00	73.11	67.56	71.89	−8.74
	Uniform sampling	83.67	76.11	71.56	77.11	−3.52
k=32	CUEKFS	89.89	83.11	76.56	83.19	–
	Outside coverage	78.33	76.33	72.56	75.74	−7.44
	Uniform sampling	89.00	80.00	75.22	81.41	−1.78

Cues retrieve evidence beyond the overview. Table 5 shows that initial cues match frames more strongly than the question in every duration split. Refinement further increases the relative gain over the overview, from 0.05 to 0.07 on short videos and from 0.43 to 0.53 on long videos. This widening gain suggests that refinement increasingly benefits exploration beyond the overview as videos grow longer. Split cues have the largest relative gain in every duration split, reaching 0.92 on long videos. These gains support refinement's role in separating bundled visual elements for retrieval.

Refinement redirects cue coverage. In Table 6(a), initial and final cue coverage have an overlap of only 0.76 overall and 0.65 on long videos. Rewritten and split cues have lower overlaps of 0.39 and 0.37 with their original coverage. Together with the stronger matches in Table 5, these results indicate that refinement actively redirects cues to re-explore different parts of the video. This redirection becomes more pronounced as videos grow longer, with split-cue overlap falling from 0.52 on short videos to 0.31 on long videos.

CUEKFS explores useful regions. In Table 6(b), we select frames outside final cue coverage using farthest point sampling initialized by medoids. Accuracy is 7.44 to 9.63 points below CUEKFS and even below uniform sampling at every budget. Thus, CUEKFS explores question-relevant regions, while frames outside its coverage are less informative.

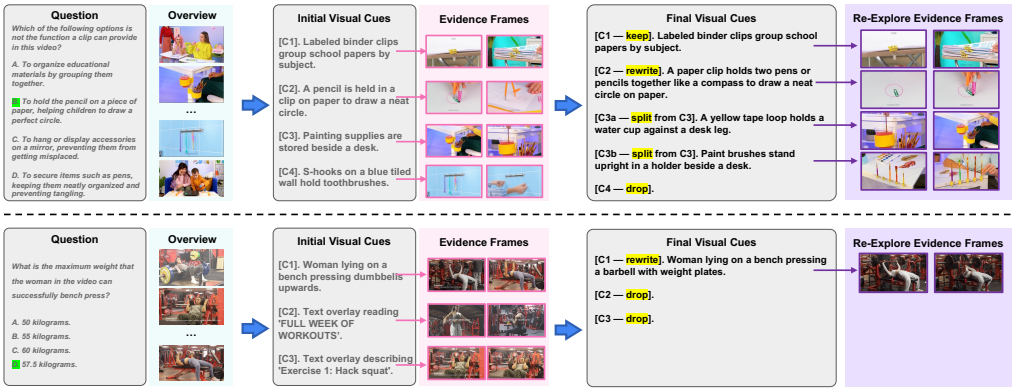

Figure 3: Qualitative illustration of CUEKFS: successful refinement (top) and failure (bottom). Correct options are green. We display selected overview frames and up to two evidence frames per cue per stage. The top panel is a curated mechanism illustration using real video frames; the bottom pairs recorded cues and actions with illustrative evidence frames.

4.5 QUALITATIVE RESULTS

Successful case. Figure 3 (top) illustrates each step of CUEKFS, starting with representative, diverse overview frames. The initial cues turn the visual elements in the answer options into distinct retrieval targets, allowing the first exploration to gather evidence frames for each candidate function. These evidence frames then benefit the agentic refinement to get the final cues. For example, C2 initially mentions "a pencil is held in a clip on a paper" based on option B, whereas its evidence reveals the clip-based construction. It triggers a rewrite as "a paper clip holds two pens or pencils," which further unearths a new frame that shows more details before drawing circles, indicating that the clip is not used to hold the pencil to the paper. Similarly, splitting the broad C3 into a cup target and a paint-brush target prevents two different constructions from competing within one query and retrieves separate evidence for each. In contrast, C4 retrieves S-hooks holding toothbrushes, which cannot verify a function of a clip. Dropping it avoids allocating the final frame budget to this distractor. Keeping the already precise C1 preserves its relevant evidence. Together, these operations produce more specific and less redundant evidence for Cue Budget Allocation's final frame selection.

Failure case and limitations. The lower panel of Figure 3 exposes two limitations. First, open-ended cue discovery can introduce irrelevant targets: the wrong hack-squat cue is eventually dropped, but its retrieval and review still consume computation. Second, event-level information can be lost. Answering *successfully* in this case requires evidence that a lift is completed, yet the VLM treats bench pressing as a single action and rewrites C1 only to identify a barbell, without separating the lowering, lifting, and completion stages. Re-exploration therefore retrieves the 60 kg attempt but not its outcome, causing the downstream VLM to mistake a failed lift for a successful one; the correct maximum is 57.5 kg. This failure exposes a limitation of the frame-wise embedding representation: since each frame is encoded independently, its similarity score does not represent the local temporal transition that determines whether the lift is completed. A temporal embedding model that represents each frame together with its neighboring frames could make such a short-lived transition appear as a coherent plateau rather than an isolated peak in our cue similarity wave, allowing our defined *plateau* to retain evidence of the transition and its outcome.

5 CONCLUSION

We presented CUEKFS, a training-free method combining a reasoning VLM and an embedding model to explore long videos. Dynamic visual cues retrieve evidence that guides their refinement before frame budget allocation. Across three benchmarks, downstream VLMs, and frame budgets, CUEKFS establishes state-of-the-art results in all 27 evaluated settings for which prior results are available. Ablations support its component designs, while behavioral analyses show useful retrieval beyond the overview and redirected exploration, particularly in longer videos. Overall, the results show that iterative, evidence-guided retrieval can sharpen underspecified cues, separate conflated targets, and suppress distractors before final selection. A promising direction is to incorporate temporal context into the embedding model to better capture event transitions and outcomes.

AI USE STATEMENT

Generative AI tools assisted with language revision and LaTeX typesetting. The authors determined the scientific content, verified all claims and reported values against the underlying experiments, and take responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

Section 3 describes the method, and Section 4.1 specifies the evaluation protocol. Table 3 reports the reasoning VLM and hyperparameter selection. Appendix A provides additional implementation details. Appendix B gives the prompt templates for Cue Decomposition and the batched agentic refinement step of Cue-Wave-Guided Refinement.

REFERENCES

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

Dan Ben-Ami, Gabriele Serussi, Kobi Cohen, and Chaim Baskin. Himu: Hierarchical multimodal frame selection for long video question answering. *arXiv preprint arXiv:2603.18558*, 2026. arXiv:2603.18558.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2603.00512.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. 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 (CVPR)*, pp. 24108–24118, June 2025.

Weiyu Guo, Ziyang Chen, Shaoguang Wang, Jianxiang He, Yijie Xu, Jinhui Ye, Ying Sun, and Hui Xiong. Logic-in-frames: Dynamic keyframe search via visual semantic-logical verification for long video understanding. In *Advances in Neural Information Processing Systems*, volume 38, pp. 124389–124422, 2025.

Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII*, volume 8695 of *Lecture Notes in Computer Science*, pp. 505–520. Springer, 2014. doi: 10.1007/978-3-319-10584-0_33.

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 *International Conference on Machine Learning (ICML)*, 2026a. arXiv:2510.03584.

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 (CVPR)*, 2026b. arXiv:2512.04000.

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 IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3318–3327, 2025. doi: 10.1109/cvpr52734.2025.00315. arXiv:2503.21483.

Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. CLIP-It! language-guided video summarization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. arXiv:2107.00650.

OpenAI. GPT-5.5 system card. https://openai.com/index/gpt-5-5-system-card/, 2026.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026. URL https://qwen.ai/blog?id=qwen3.6-27b.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Michal Shlapentokh-Rothman, Prachi Garg, Yu-Xiong Wang, and Derek Hoiem. Decomposing queries into tool calls for long-video keyframe retrieval. *arXiv preprint arXiv:2605.23826*, 2026. arXiv:2605.23826.

Hui Sun, Shiyin Lu, Huanyu Wang, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Ming Li. MDP^3: A training-free approach for list-wise frame selection in video-LLMs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24090–24101, 2025. arXiv:2501.02885.

Wenhui Tan, Xiaoyi Yu, Jiaze Li, Yijing Chen, Jianzhong Ju, Zhenbo Luo, Ruihua Song, and Jian Luan. MSJoE: Jointly evolving MLLM and sampler for efficient long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2602.22932.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2502.21271.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Shaoguang Wang, Weiyu Guo, Ziyang Chen, Xuming Hu, and Hui Xiong. Where to focus: Query-modulated multimodal keyframe selection for long video understanding. *arXiv preprint arXiv:2604.17422*, 2026a.

Yiheng Wang, Lichen Zhu, Yueqian Lin, Yudong Liu, Jingyang Zhang, Hai "Helen" Li, and Yiran Chen. Query-conditioned evidential keyframe sampling for MLLM-based long-form video understanding. *arXiv preprint arXiv:2604.01002*, 2026b.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 28828–28857, 2024. arXiv:2407.15754.

Linli Yao, Haoning Wu, Kun Ouyang, Yuanxing Zhang, Caiming Xiong, Bei Chen, Xu Sun, and Junnan Li. Generative frame sampler for long video understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 17900–17917, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.921. URL https://aclanthology.org/2025.findings-acl.921/. arXiv:2503.09146.

Jinhui Ye, Zihan Wang, Haosen Sun, Keshigeyan Chandrasegaran, Zane Durante, Cristobal Eyzaguirre, Yonatan Bisk, Juan Carlos Niebles, Ehsan Adeli, Li Fei-Fei, Jiajun Wu, and Manling Li. Re-thinking temporal search for long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8579–8591, 2025. arXiv:2504.02259.

Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xiaolei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, Hao Zhang, and Qianru Sun. Frame-voyager: Learning to query frames for video large language models. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.03226.

Haopeng Zhang. Bridging multimodal and video summarization: A unified survey. In *Proceedings of the 5th New Frontiers in Summarization Workshop (NewSum)*, pp. 157–171, 2025.

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 (ICCV)*, pp. 22056–22065, 2025. arXiv:2506.22139.

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

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

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 *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.04428.

APPENDIX

A ADDITIONAL IMPLEMENTATION DETAILS

Image preprocessing and decoding. Images supplied to SigLIP2 have a maximum edge of 384 pixels. Qwen3.6-27B uses greedy decoding with temperature 0.

Cue Decomposition. Overview segmentation uses a db4 discrete wavelet transform and retains its finest detail band. Boundary peaks are detected in the absolute reconstructed response, with height at least $\mathrm{mean} + 0.5\,\mathrm{std}$, prominence at least 0.05 times the response range, and separation of at least $\max(5, \lfloor 0.02n \rfloor)$ frames, where n is the length of the signal currently being segmented, including recursive splits. Clip durations are constrained to between 4 and 12 s. Overview slots are allocated approximately in proportion to each clip's candidate-frame count, with at least one slot per clip when the budget permits. We floor each clip's share of the total overview budget, enforce the one-slot minimum, and distribute leftover slots by fractional remainder; any excess is trimmed from clips with smaller weights that have more than one slot. Each clip's allocation is capped by its candidate count, and its medoid initializes within-clip farthest-point sampling. If the selected representatives underfill the budget, global farthest-point sampling adds unused frames, initialized with the selected representatives. If the medoids outnumber the overview budget, farthest-point sampling reduces the medoid set.

Cue-Wave-Guided Refinement. The prominence base window extends $\delta{=}10$ frames on each side of a peak. Retained peak centers have a minimum separation of $d_p{=}3$ frames, with at most $n_p{=}5$ peaks per cue. Above-threshold intervals are merged when the next start index minus the previous end index is at most 2, allowing at most one intervening below-threshold frame. A retained plateau spans at least $\ell_0{=}3$ frames and contributes $n_{\text{pl}}{=}3$ samples. Within each refinement batch, each cue with available evidence first receives one image. Each remaining image slot goes to the cue with the largest shortfall from its evidence-count-proportional share, up to its available evidence count. Evidence images shown to the reasoning VLM are arranged in temporal order within each cue and mapped to cue indices in the prompt.

Cue Budget Allocation. Let a_c be cue c's retained evidence count and $r_c = k a_c / \sum_j a_j$ its target share of the total budget. If k cannot cover all cues with evidence, the top-k cues by rank receive one slot each. Otherwise, each nonempty cue starts with $k_c = 1$; while slots remain, we increment the eligible cue with the largest $r_c - k_c$, subject to $k_c < a_c$, breaking ties by cue rank. If the total retained evidence count is at most k, all retained evidence frames are selected. If the retained evidence contains fewer than k frames, farthest-point sampling in embedding space fills the remaining slots from unused candidate frames, initialized with the frames already selected.

Scores for cue analysis. The analyses in Tables 5 and 6 convert raw cosine similarities to $\sigma(\alpha s_t^c + \beta)$, where $\alpha > 0$ and β are SigLIP2's frozen scale and bias, and σ is the sigmoid function.

B FULL PROMPTS

Figures 4, 5, and 6 provide the prompt templates for Cue Decomposition, the batched agentic refinement step of Cue-Wave-Guided Refinement, and downstream QA, respectively.

System

```
You are given some context about a video -- a question, an instruction, a
description target, or a query -- and a few sample frames from it. Write short
English search phrases ("cues"); each cue is later matched against the video's
frames to find where some visual thing appears. You only say what to look for;
you never answer the context.

Each cue:
- one self-contained sentence, < 25 words, describing what a single frame could
  show -- a subject, attribute, action, spatial relation, or on-screen text
  (quote the shortest useful fragment);
- specific enough to find by how a frame looks;
- distinct from the others;

When the context involves counting, ordering, before-after, or change-over-time,
write separate cues for distinct instances, events, or stages.

The attached frames are only a sparse sample of the video; if the context
needs something not visible in them, still write a cue for it.

Return one JSON object matching the schema, nothing else.
```

User

```
CONTEXT
"<question>"

CANDIDATE ANSWERS
A. <option A>   B. <option B>   ...

The attached frames are in time order.
Order the cues roughly by when each is likely to appear in the video.

Return exactly:
{ "visual_cue_list": [ {"cue": "<cue 1>"} ] }
```

Figure 4: Prompt template for Cue Decomposition. The reasoning VLM receives the task context and video overview and returns initial visual cues. The candidate-answer block is included only for multiple-choice inputs.

System

```
You are given SEVERAL search phrases ("cues"), each with a few video frames that
look most like that cue.

For EACH cue, choose one action:
- drop:    the cue cannot help with the context.
- keep:    the cue is useful for the context; keep it unchanged.
- rewrite: the cue is useful but imprecise or retrieving the wrong thing; give
           one better cue, still faithful to the context.
- split:   the cue bundles two unrelated visual things; give two or more
           specific cues.

Return one JSON object with exactly one action PER cue, each tagged with that
cue's index, nothing else.
```

User

```
CONTEXT
"<question>"

CANDIDATE ANSWERS
A. <option A>   B. <option B>   ...

CUES (N total). The attached images are these cues' evidence frames concatenated
in cue order, in time order within each cue:
[0] "<cue 0>"  (images 1-4)
[1] "<cue 1>"  (images 5-7)
...

For EACH cue index return one action. Return exactly:
{ "reviews": [
  {"index": 0, "action": "keep"},
  {"index": 1, "action": "rewrite", "cue": "<rewritten cue>"},
  {"index": 2, "action": "split", "cues": ["<cue 1>", "<cue 2>", ...]},
  {"index": 3, "action": "drop"}
] }
```

Figure 5: Prompt for batched agentic refinement in Cue-Wave-Guided Refinement. Each cue is paired with its own cue-wave evidence through an indexed image range. The reasoning VLM returns one keep, rewrite, split, or drop action per cue.

System

```
You are a helpful assistant.
```

User: Video-MME

```
Select the best answer to the following multiple-choice question based on the
video. Respond with only the letter (A, B, C, or D) of the correct option.
<question>
A. <option A>
B. <option B>
C. <option C>
D. <option D>

Answer with the option's letter from the given choices directly.
```

User: LongVideoBench

```
<question>
A. <option A>
B. <option B>
C. <option C>
D. <option D>
E. <option E, when present>
Answer with the option's letter from the given choices directly.
```

User: MLVU

```
<question and candidate options>
Only give the best option.
Best option: (
```

Figure 6: Downstream QA prompts. All downstream VLMs receive the selected frames in temporal order with the corresponding dataset-specific user prompt.