VideoSeeker: Coarse-to-Fine Adaptive Evidence Selection for Long Video Understanding

Anonymous ACL submission

Abstract

Long-video question answering (QA) forces video-language models (VLMs) to find sparse answer-supporting evidence from lengthy temporal contexts under a tight frame budget. Existing methods either sample uniformly and miss temporally sparse evidence, or select by query-frame similarity and retrieve *visually similar but answer-irrelevant* distractors that crowd out the decisive frames, but query-relevant segments are not always useful for answering: they may share scenes, entities, or actions with the question yet miss the decisive evidence. We present **VideoSeeker**, a coarse-to-fine adaptive evidence-selection framework organized as a **Retrieve–Verify–Answer** (RVA) pipeline. VideoSeeker first trains a time-aware intra-video contrastive retriever that returns a high-recall pool of candidate segments. It then uses Qwen2.5-Omni-3B as a lightweight segment verifier: each retrieved segment is paired with the question and the Yes/No token logits are converted into an answer-utility probability. Retrieval and verification scores are fused to rerank segments, after which a fixed frame budget is adaptively allocated across the top verified segments before frozen VLM inference. On four long-video QA benchmarks, VideoSeeker consistently improves accuracy by +5.1 to +8.1 points across four frozen VLM backbones under a 64-frame budget, with the largest gains at tighter frame constraints.

1 Introduction

Long-video understanding has become a central problem for modern video-language models (VLMs) (Zhang et al., 2024a; Wang et al., 2024a; Zhu et al., 2025a). Unlike short-clip QA, it requires locating sparse evidence inside an extended temporal context and reasoning over it. VLMs face fundamental architectural constraints: each frame costs hundreds of visual tokens, self-attention scales quadratically with sequence length, and the context window is hard-capped. Even when long inputs fit,

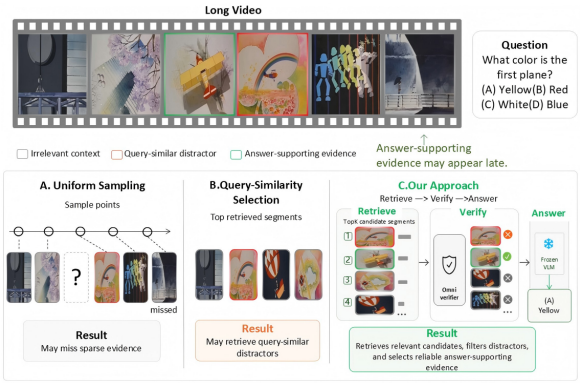

Figure 1: Motivation of answer-supporting evidence selection in long-video QA. Query-similarity is not sufficient; an additional verification stage is needed to identify truly useful evidence.

well-documented attention pathologies—the *lost-in-the-middle* effect (Liu et al., 2024) and attention-sink behaviour (Xiao et al., 2024)—cause models to underutilize information buried in long token sequences. The practical bottleneck is therefore not just *how strong* the VLM is, but *which frames* it actually sees.

One line of research attacks this bottleneck by *compressing* visual tokens before they reach the LLM (Shu et al., 2025; Liu et al., 2025), raising the ceiling on how many tokens can be processed. However, compression does not decide *which* temporal regions deserve those tokens. The dominant deployment pattern remains *uniform frame sampling*—simple and model-agnostic, yet treating every temporal region as equally informative, an assumption badly violated in long videos where answer-bearing events are short and concentrated. Query-aware selection methods address this by scoring frames by visual–semantic similarity to the question (Tang et al., 2025; Zhang et al., 2025b; Zhu et al., 2025b), but *query relevance is not equivalent to answer utility*. Long videos contain far

more same-scene distractors than short clips—a 30-minute cooking video may show the same kitchen counter in dozens of segments while the answer hinges on one specific action—and since the frame budget is fixed, every frame spent on a distractor directly displaces an evidence frame. As videos grow longer, distractors grow linearly while the budget stays constant, making *precision-improving filtering* increasingly critical.

As illustrated in Figure 1, a question about what a person finally takes from a table can retrieve many visually similar table-related clips, while only one short segment actually reveals the answer. Long-video QA therefore demands a pipeline that first *retrieves* potentially relevant segments and then explicitly *verifies* whether each segment is useful for answering—a function fundamentally orthogonal to token compression or attention sparsification. We formulate this as a coarse-to-fine **Retrieve–Verify–Answer** (RVA) process and propose **VideoSeeker** to instantiate it.
Our main contributions are:

- We identify the *relevance–utility gap* as a key bottleneck in long-video understanding and propose the RVA framework that explicitly separates retrieval (recall) from verification (precision).
- We introduce a *time-aware intra-video contrastive* training objective for the retriever, which handles same-video distractors by treating temporally adjacent segments as soft neighbours.
- We design a lightweight VLM-based verifier that converts `Yes`/`No` token logits into calibrated verification probabilities, enabling score-level fusion to suppress query-similar but answer-irrelevant candidates.

2 Related Work

Frame selection under limited budgets. Frame-selection methods reduce long-video inputs before VLM inference. Uniform sampling is robust but query-agnostic. Query-aware methods such as AKS (Tang et al., 2025), Q-Frame (Zhang et al., 2025b), and FOCUS (Zhu et al., 2025b) use text–frame similarity or learned policies to prioritize question-related frames. Training-based selectors such as TSPO (Tang et al., 2026), ViaRL (Xu et al., 2025), and GenS (Yao et al., 2025) learn a selection policy from QA reward. These methods are effective when question-relevant content is visually distinctive, but frame-level selection is unstable on long videos: isolated frames lack temporal context, and query-similar frames may still be insufficient for answering. VideoSeeker addresses this by operating at the segment level, preserving local temporal context before allocating frames inside selected segments.

Intra-video segment retrieval. Video-text retrieval (Luo et al., 2021; Wang et al., 2024b) aligns videos and text in a shared embedding space to retrieve videos from a large corpus. Long-video QA requires a different setting: given one long video and one question, the model must distinguish answer-supporting segments from many same-video distractors. Such intra-video negatives share scenes, characters, and events with the positive evidence and are inherently harder. We address this by training a time-aware intra-video contrastive retriever, where the temporal distance from the annotated evidence modulates negative reliability.

Verification and reranking. Retrieval-augmented reasoning systems often benefit from a second-stage verifier or reranker, because top retrieved items can be relevant but not necessarily useful (Asai et al., 2024; Yu et al., 2024). In long-video QA, this gap is especially important: a segment may match question keywords but lack the evidence required for the answer. Existing video QA methods rely on the downstream VLM to implicitly filter noise, which is unreliable when distractors are visually near-identical to true evidence. VideoSeeker externalizes verification as a dedicated lightweight stage using Qwen2.5-Omni-3B, using Yes/No logits as a calibrated utility signal for reranking and frame allocation.

3 Method

3.1 Overview

Given a long video V partitioned into N fixed-length 15-second segments $\{s_1, \ldots, s_N\}$ and a natural-language question q, VideoSeeker selects a small set of evidence segments and a fixed frame budget that a *frozen* vision–language model (VLM) consumes to produce the answer $\hat{a}$. Because the answer model is never updated, every improvement must come from better evidence selection rather than from stronger generation.

To this end, VideoSeeker organises evidence selection as a **Retrieve–Verify–Answer** pipeline

(Figure 2). A time-aware intra-video retriever first returns a top-K_r candidate pool optimised for recall; Qwen2.5-Omni-3B then verifies each candidate's *answer utility* through its Yes/No logits; finally, retrieval and verification signals are fused to rerank the candidates into a compact top-K_s set ($K_s < K_r$), across which a fixed frame budget B is adaptively distributed. The three stages operate on progressively smaller candidate sets, which keeps the overall cost tractable for hour-long videos despite the increasing per-segment cost.

3.2 Stage 1: Time-Aware Intra-Video Retrieval

Stage 1 nominates a recall-oriented candidate pool. We encode the question and every segment with VideoSeeker and rank segments by cosine similarity, denoted r_i for s_i.VideoSeeker is fine-tuned with a contrastive objective on segment-level data built from TimeLens-100K (Appendix A): segments overlapping the annotated evidence span $[a, b]$ form positives, while negatives are drawn *exclusively from the same video* as the query, mixing hard negatives adjacent to the evidence with medium-difficulty negatives sampled from more distant parts of the video. This intra-video-only construction reflects our target application—retrieving evidence within a single long video rather than across a corpus—and ensures that the training distribution matches the inference-time candidate pool.

Why naive contrastive learning is insufficient. A standard InfoNCE objective treats every negative as equally wrong:

$$\mathcal{L}_{\text{con}} = -\log \frac{\exp(r^+/\tau)}{\exp(r^+/\tau) + \sum_{s_i^- \in \mathcal{N}} \exp(r_i^-/\tau)}. \quad (1)$$

In an intra-video pool, however, segments adjacent to the ground-truth interval typically share scenes, speakers, or partial evidence with the positives, and penalising them as strongly as truly unrelated segments injects label noise that destabilises training and pushes the encoder toward over-confident—but temporally brittle—decision boundaries.

Time-aware reweighting. We instead modulate each negative's penalty by its temporal distance to the ground-truth evidence span. For a negative segment $[s_i^-, e_i^-]$,

$$d_i = \max\big(0,\ \max(a - e_i^-,\ s_i^- - b)\big), \quad (2)$$

which is zero whenever the segment overlaps the evidence and grows linearly as it moves away. This distance is mapped into a soft weight

$$w_i = 1 - \exp(-d_i/\lambda)\,, \quad (3)$$

with λ controlling how quickly the penalty recovers (default $\lambda = 30\,\text{s}$). The weight smoothly interpolates between *fully suppressing* boundary-adjacent negatives ($d_i \to 0,\ w_i \to 0$) and *fully penalising* distant or cross-video negatives ($d_i \to \infty,\ w_i \to 1$); the latter limit exactly recovers Eq. 1.

For a query with K positives and M reweighted negatives, the per-positive loss becomes

$$\mathcal{L}_k = -\log \frac{\exp(r_k^+/\tau)}{\sum_{k'=1}^{K} \exp(r_{k'}^+/\tau) + \sum_{j=1}^{M} w_j \exp(r_j^-/\tau)} \quad (4)$$

and the final time-aware contrastive loss $\mathcal{L}_{\text{ta}}$ averages $\mathcal{L}_k$ over the K positives. At inference we keep the top-K_r segments by r_i as the candidate pool $\mathcal{C}_{K_r}$.

3.3 Stage 2: Omni Verification with Yes/No Logits

Retrieval relevance and *answer utility* are related but distinct: a segment can be topically on-topic yet contain no decisive evidence, while a segment with modest cosine similarity may still carry the crucial cue. Stage 2 closes this gap with a lightweight verifier built from Qwen2.5-Omni-3B. For each $s_i \in \mathcal{C}_{K_r}$ we prompt the model with the question and the segment and instruct it to answer with a single word, *yes* or *no*.

Rather than parsing the generated string—which is brittle and discards the model's calibration—we read the next-token logits directly and aggregate over all single-token surface forms of each class via LogSumExp, yielding two class logits L_i^{yes} and L_i^{no} . These collapse into both an unbounded fusion score and a calibrated verification probability:

$$s_i^{\text{Qwen}} = L_i^{\text{yes}} - L_i^{\text{no}}, \qquad v_i = \sigma\big(s_i^{\text{Qwen}}\big). \quad (5)$$

We adopt the log-odds s_i^{Qwen} for downstream fusion because the bounded probability v_i saturates on confident segments and loses discriminative power precisely where fine-grained ranking matters most; v_i is retained only as a calibrated threshold for filtering in Stage 3.

3.4 Stage 3: Score Fusion and Adaptive Frame Allocation

The fused utility combines retrieval and verification after z-score normalisation over the candidate pool,

$$\tilde{u}_i = \alpha\, \text{zNorm}\big(s_i^{\text{Qwen}}\big) + (1-\alpha)\, \text{zNorm}(r_i), \quad (6)$$

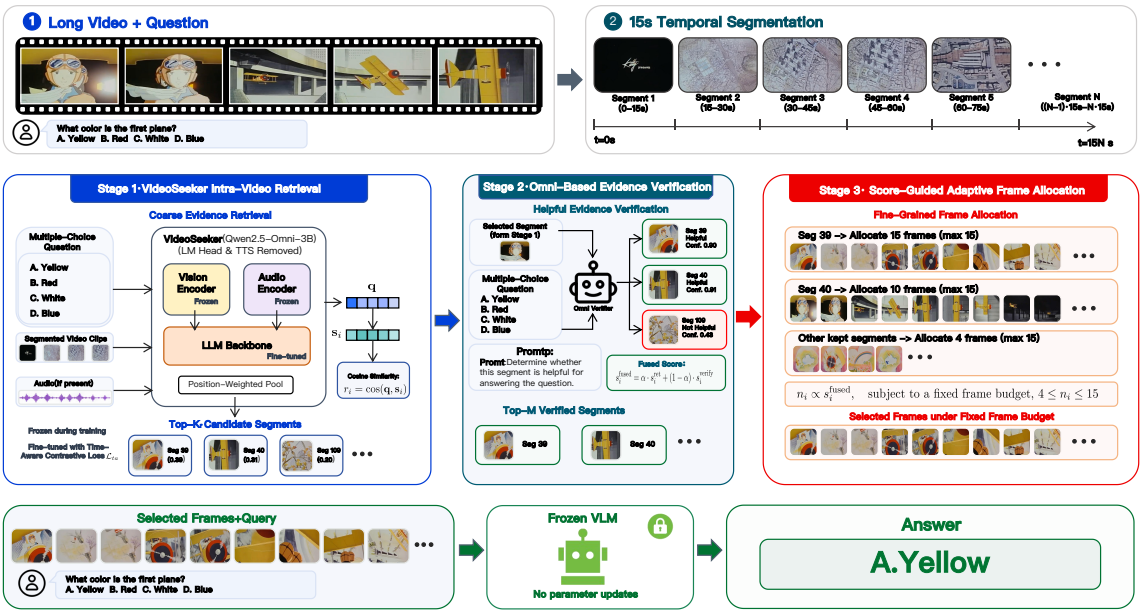

Figure 2: Overview of VideoSeeker. The pipeline progressively refines candidate evidence from a large pool of segments to a small set of high-utility frames consumed by a frozen VLM.

where α controls the relative weight of the verifier. Normalisation is essential here because s_i^{Qwen} is an unbounded log-odds whose scale varies across questions, whereas r_i lies in $[-1, 1]$; without it, the verifier signal would dominate fusion in a question-dependent and largely uncontrollable way. The final evidence set $\mathcal{S}_{K_s}$ consists of the K_s candidates in $\mathcal{C}_{K_r}$ with the highest $\tilde{u}_i$ ($K_s < K_r$); no hard threshold is applied, since the subsequent frame allocator already down-weights low-utility segments through Eq. 7.

From utilities to a frame budget. Distributing B frames uniformly across $\mathcal{S}_{K_s}$ wastes capacity on segments with weak evidence, whereas a purely proportional split starves lower-ranked but still informative segments. We therefore combine a *base allocation* with a *utility-driven residual*: each segment first receives $f_{\min}$ frames, and the remaining $R = B - K f_{\min}$ frames are distributed according to a temperature-controlled softmax over the fused utilities,

$$p_i = \frac{\exp(\tilde{u}_i/\tau_f)}{\sum_j \exp(\tilde{u}_j/\tau_f)}. \tag{7}$$

To prevent any single segment from monopolising the budget, we further cap its allocation at $f_{\max}$ frames, yielding

$$m_i = f_{\min} + \min\big(\lfloor p_i R \rfloor,\ f_{\max} - f_{\min}\big) + \delta_i, \tag{8}$$

where $\delta_i \in \{0, 1\}$ absorbs the floor residual to ensure $\sum_i m_i = B$, and is assigned one frame at a time, in descending order of fractional remainder, to segments not yet saturated at $f_{\max}$. Frames are then sampled uniformly within each selected segment and concatenated in temporal order before being fed to the frozen VLM. Algorithm 1 summarises the complete pipeline.

4 Experiments

4.1 Setup

Datasets. The retriever is trained on 40K query–evidence pairs constructed from TimeLens (Zhang et al., 2025a), an open-source long-video QA dataset. Each pair comprises one query, 1–3 positive audio–video segments overlapping the annotated evidence interval, and 6 in-video hard negatives drawn from the same video (see Appendix A for the full construction protocol and negative composition). We evaluate on four long-video QA benchmarks: **LVBench** (Wang et al., 2025), **Video-MME** (long split) (Fu et al., 2024), **MLVU** (Zhou et al., 2024), and **LongVideoBench** (Wu et al., 2024), covering action, temporal reasoning, object perception, OCR, speech, and multi-hop questions.

Baselines. We compare with four groups. (i) Reference closed-source / large open-source VLMs (GPT-4o (Hurst et al., 2024), Gemini-1.5-Pro (Team et al., 2024), Qwen2.5VL-72B (Team, 2025a), LLaVA-Video-72B (Zhang et al., 2024b)). (ii) Training-free frame selectors: AKS (Tang

Table 1: Segment retrieval quality on the LVBench evaluation split. *Hit@K*: fraction of queries with ≥ 1 relevant segment in the top-K results (overlap ≥ 1 s). *Coverage@K*: fraction of ground-truth evidence duration covered by the temporal union of the top-K segments. *MRR@K*: Mean Reciprocal Rank at cut-off K. Qwen3Embedding-8B is a text-only model applied to LLM-generated segment captions. **VideoSeeker(R)**: retriever only; **VideoSeeker(R+V)**: retriever with verifier re-ranking. For each column, **bold** indicates the best result and underline indicates the second-best result..

Method	Hit@1	Hit@3	Hit@5	Hit@10	Coverage@5	Coverage@10	MRR@10
VideoCLIP	0.1092	0.1713	0.2049	0.2657	0.0447	0.0723	0.1506
Clip4Clip	0.2023	0.3200	0.3872	0.5048	0.1469	0.2156	0.2845
X-Pool	0.2424	0.3588	0.4357	0.5430	0.1705	0.2477	0.3234
Qwen3Embeding-8B	0.3143	0.4706	0.5318	0.6350	0.2365	0.3227	0.4098
VideoSeeker(R)	0.3199	0.4609	0.5351	0.6342	0.2364	0.3203	0.4094
VideoSeeker(R+V)	**0.3771**	**0.5310**	**0.5995**	**0.6882**	**0.2918**	**0.3698**	**0.4733**

et al., 2025), Q-Frame (Zhang et al., 2025b), FOCUS (Zhu et al., 2025b). (iii) Training-based frame selectors: TSPO (Tang et al., 2026), ViaRL (Xu et al., 2025), GenS (Yao et al., 2025). (iv) Embedding-based segment retrievers: VideoCLIP (Xu et al., 2021), Clip4Clip (Luo et al., 2021), and X-Pool (Gorti et al., 2022) are video–text dual encoders with progressively stronger cross-modal fusion; Qwen3Embedding-8B (Zhang et al., 2025c) is a text-only embedding model applied to LLM-generated captions of each 15 s segment. All group–(iv) methods retrieve via cosine similarity with no task-specific training. Groups (ii)–(iv) share the identical frozen answer VLM and frame budget as VideoSeeker, so any accuracy difference is attributable *purely* to evidence selection.

Implementation. The retriever is built on **Qwen2.5-Omni-3B**: we discard the LM head and TTS module and reuse the vision encoder, audio encoder, and LLM backbone to produce query and segment embeddings. The vision and audio encoders are *frozen*; the LLM backbone is fully fine-tuned in **bf16** with AdamW (lr$=1\times10^{-5}$, gradient accumulation 4), completing 1 epoch on 40K queries in $\approx$24 hours on 8 Ascend AI cores. The verifier $\mathcal{V}_{\text{Omni}}$ reuses the same Qwen2.5-Omni-3B in inference-only mode with no additional fine-tuning. Default hyperparameters: 15 s segments, K_r=30, K_s=10, B=64, α=0.7, τ=0.07, τ_f=1.0, $f_{\min}$=4, $f_{\max}$=15; full details are in Appendix B and C.

Metrics. Our primary metric is multiple-choice QA accuracy. For the dedicated retrieval evaluation (Table 1) we additionally report three metrics: **Hit@K**—fraction of queries with ≥ 1 relevant segment in the top-K results (overlap threshold ≥ 1 s); **Coverage@K**—fraction of ground-truth evidence duration covered by the temporal union of the top-K segments; and **MRR@K**:

$$\text{MRR@}K = \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}}\frac{\mathbf{1}[r_q \leq K]}{r_q} \tag{9}$$

where r_q is the rank of the first relevant segment for query q, and the contribution is zero if no relevant segment appears in the top-K.

4.2 Main Results

Segment Retrieval Quality (Table 1). On the LVBench evaluation split, classical video–text encoders (VideoCLIP, Clip4Clip, X-Pool) lag considerably across all metrics, confirming that generic pre-training objectives transfer poorly to fine-grained temporal evidence localisation. Qwen3Embedding-8B narrows this gap through scale, yet VideoSeeker(R)—a 3B model fine-tuned on only 40K audio–video pairs—matches it on nearly every metric (Hit@5: 0.5351 vs. 0.5318; MRR@10: 0.4094 vs. 0.4098), demonstrating that domain-specific audio–video contrastive fine-tuning is a more efficient lever than raw model scale. Adding the verifier, VideoSeeker(R+V) improves consistently over VideoSeeker(R): Hit@1 +5.7 pts, MRR@10 +6.4 pts, Coverage@10 +5.0 pts, confirming that answer-utility re-ranking provides a complementary signal inaccessible to pure embedding models.

QA Accuracy (Tables 2–3). VideoSeeker yields consistent improvements over uniform sampling across all four benchmarks and all four frozen answer backbones (average gains +5.1–+8.1 pts); because only the selected frames change, these improvements are directly attributable to evidence

selection, confirming backbone-agnosticism. The magnitude varies systematically with *evidence sparsity*: LVBench (67-min videos, 17 s median clue interval, $< 1\%$ of runtime) yields the largest gains ($\geq +11$ on every backbone); MLVU and LongVideoBench (moderate lengths, no annotated evidence intervals; see Table 11) yield $+3$ to $+5$; and Video-MME's dense evidence ($\approx$967 s average clue interval) leaves the smallest headroom ($+2$ to $+3$). This pattern supports our central claim: explicit retrieval and answer-utility verification matter most when evidence is sparse, and are harmless when it is dense.

The magnitude of the gain varies systematically with the *evidence sparsity* of each benchmark. LVBench, whose videos average 67 min while the median clue interval is only 17 s ($< 1\%$ of the runtime), produces by far the largest gains ($\geq +11$ on every backbone); MLVU and LongVideoBench, with moderate video lengths (15.6 min and 7.9 min on average; no annotated evidence intervals are available, see Table 11), yield $+3$ to $+5$; Video-MME has the densest evidence (clue intervals averaging ≈ 967 s on the long split) and accordingly the smallest headroom ($+2$ to $+3$). This pattern supports our central claim: explicit retrieval and answer-utility verification are most valuable when relevant evidence is hard to reach by chance, and harmless when evidence is denser. Per-benchmark statistics are summarised in Table 11.

4.3 Comparison with SOTA Frame Selectors

The lower block of Table 2 compares VideoSeeker with recent training-free and training-based frame selectors under the same Qwen3VL-8B / 64-frame setting. VideoSeeker outperforms all baselines on average, exceeding the strongest training-free competitor (FOCUS) by $+2.77$ points and the strongest training-based one (TSPO) by $+5.11$. The retrieval-only variant (*R+A*) already surpasses every prior selector on average, and inserting the verifier (*R+V+A*) provides a further $+1.09$, validating the RVA decomposition.

4.4 Ablation: Retrieval, Verification, and Adaptive Allocation

We isolate the contribution of the three components of VideoSeeker by progressively enabling them while keeping the answerer (Qwen3VL-8B), the total frame budget (B=64), and the uniform in-segment sampling rule fixed (Table 4). Retrieval contributes by far the largest gain ($+6.34$), confirming that at B=64 *which* segments are passed to the answerer matters more than how the budget is split among them. Adaptive allocation, which redistributes frames in proportion to retrieval scores, adds a small but consistent $+0.64$. The verifier provides a complementary $+1.09$ on top, filtering segments that look semantically related to the query but do not actually support an answer—a failure mode retrieval alone cannot address. The three contributions are additive ($+6.34/+0.64/+1.09 \Rightarrow +8.07$ total) and target distinct failure modes, validating the RVA decomposition.

4.5 Ablation: Retriever–Verifier Fusion Weight

We next study how to combine the retriever score r_i and the Omni log-odds s_i^{Qwen}. Table 5 sweeps the fusion weight α of Eq. 6, where α=0 corresponds to using the retriever alone and α=1 to using the verifier alone. Two trends emerge. First, neither extreme is optimal: retrieval-only (α=0) and verifier-only (α=1) are both clearly worse than balanced fusion, confirming that retrieval and verification capture *complementary* aspects of evidence quality. Second, accuracy is broadly insensitive to α inside the $[0.3, 0.7]$ range, which is exactly the regime in which both signals are non-trivially weighted; we adopt α=0.7 as the default because it gives the best downstream accuracy.

4.6 Ablation: Retrieval Objective and Frame Budget

We finally verify two design decisions: the *time-aware* contrastive objective for the retriever, and the robustness of the full pipeline across *frame budgets*.

Time-aware vs. standard contrastive. Table 6 compares standard InfoNCE with our time-aware objective. The time-aware loss improves downstream accuracy by $+0.6$ on MLVU and remains competitive on Video-MME, confirming that adjusting negative reliability by temporal distance helps in the intra-video setting. Removing the audio stream further drops accuracy on both benchmarks, justifying the use of the Omni audio encoder.

Robustness across frame budgets. Figure 3 report LVBench accuracy under $B \in \{16, 32, 64, 128\}$. VideoSeeker outperforms uniform sampling at every budget, with the gap shrinking from $+16.48$ at B=16 to $+11.26$

Model / Method	Venue	#Frames	LVBench	Video-MME	MLVU	LongVideoBench	Avg.
Reference VLMs with reported configurations							
GPT-4o	–	–	30.8	71.9	54.9	66.7	56.07
Gemini-1.5-Pro	–	–	33.1	**75.0**	–	64.0	–
Qwen2.5VL-72B	–	–	47.3	73.3	–	60.7	–
LLaVA-Video-72B	–	–	–	70.5	–	61.9	–
Frame-selection methods under controlled Qwen3VL-8B / 64-frame setting							
Uniform Sampling	–	64	40.42	69.22	51.79	62.90	56.08
Training-free frame selection							
AKS	CVPR 2025	64	45.64	70.63	57.57	64.02	59.46
Q-Frame	ICCV 2025	64	43.10	62.75	59.07	52.99	54.47
FOCUS	ICLR 2026	64	53.49	67.04	59.96	65.04	61.38
Training-based frame selection							
TSPO	AAAI 2026	64	48.79	67.07	57.72	62.57	59.04
ViaRL	arXiv 2025	64	42.43	63.74	44.82	62.15	53.29
GenS	ACL-F 2025	64	50.74	66.74	50.40	63.80	57.92
VideoSeeker (R+A)	–	64	51.68	71.44	**61.75**	67.39	63.06
VideoSeeker (R+V+A)	–	64	**55.56**	72.66	60.55	**67.84**	**64.15**

Table 2: Comparison with reference VLMs and state-of-the-art frame-selection methods on four long-video understanding benchmarks. The upper block reports results of representative large-scale VLMs under their original configurations, serving as external references. The lower block presents a controlled comparison in which all frame-selection methods share the same Qwen3VL-8B answer model and an identical budget of 64 input frames, ensuring a fair evaluation of the selection strategy itself. "R+A" denotes retrieval followed by adaptive frame allocation, while "R+V+A" further incorporates an Omni-based verification stage prior to allocation. For each column, **bold** indicates the best result and underline indicates the second-best result.

Backbone	#Frames	LVBench	Video-MME	MLVU	LongVideoBench	Avg.
InternVL3.5-8B	64	43.36	68.22	52.59	63.47	56.91
w/ VideoSeeker	64	54.89↑11.53	71.11↑2.89	56.37↑3.78	68.21↑4.74	62.64↑5.73
MiniCPM-V4.5-9B	64	41.55	63.40	54.98	50.79	52.68
w/ VideoSeeker	64	53.35↑11.80	66.66↑3.26	57.77↑2.88	54.90↑4.11	58.17↑5.49
Kimi-VL-A3B	64	44.84	70.70	60.36	63.50	59.85
w/ VideoSeeker	64	58.85↑14.01	72.55↑1.85	61.35↑0.99	67.09↑3.59	64.95↑5.10
Qwen3VL-8B	64	40.42	69.22	51.79	62.90	56.08
w/ VideoSeeker	64	55.56↑15.14	72.66↑3.44	60.55↑8.76	67.84↑4.94	64.15↑8.07

Table 3: Backbone-agnostic improvements from adding VideoSeeker. For each frozen open-source VLM, the baseline and "w/ VideoSeeker" rows use the same answer backbone and the same 64-frame budget; only the selected input frames are changed.

at B=128 as uniform sampling increasingly intersects the evidence by chance. Notably, VideoSeeker itself plateaus from B=64 to B=128 (55.56 $\rightarrow$ 55.29): once the retriever's top-K_s segments are covered, extra frames mainly add near-duplicate views, and the longer visual-token sequence is known to dilute attention over the informative frames (the "lost-in-the-middle" effect (Liu et al., 2024)). VideoSeeker therefore delivers its main benefit under tight-budget deployment, where principled evidence selection matters most.

4.7 Question-Type Analysis

To localize where the gain comes from, we split LVBench by question category (Table 7). Improvements are largest on *Key Information* (+21.25) and *Entity Recognition* (+19.32), where evidence is highly local and easily confused with visually similar distractors—precisely the regime the Omni verifier is designed to handle. Gains on *Reasoning* (+7.65) and *Summarization* (+3.45) are smaller,

Variant	R	V	A	Avg.
Uniform (whole-video)	✗	✗	✗	56.08
+ Retrieval (uniform alloc.)	✓	✗	✗	62.42
+ Retrieval + Adaptive Alloc.	✓	✗	✓	63.06
+ Retrieval + Verifier + Adaptive Alloc.	✓	✓	✓	**64.15**

Table 4: Ablation on the three components of VideoSeeker: retrieval (**R**), answer-utility verification (**V**), and adaptive frame allocation (**A**). All variants share the same answerer (Qwen3VL-8B), the same total frame budget (B=64), and uniform in-segment sampling. Numbers are accuracy (%) averaged over the four benchmarks in Table 3.

Fusion Weight α	Hit Rate@K_s (%)	Avg. Acc. (%)
0.0 (retrieval only)	53.51	61.38
0.3	56.97	63.52
0.5	59.07	64.06
0.7	59.95	64.15
1.0 (verifier only)	57.86	63.33

Table 5: Effect of the retriever–verifier fusion weight α (Eq. 6). Hit Rate is measured at the top-K_s reranked candidates and counts a hit when a retained segment overlaps the ground-truth evidence interval. Avg. Acc. is the mean QA accuracy over LVBench, Video-MME, MLVU, and LongVideoBench under Qwen3VL-8B / 64-frame budget.

Retriever	LongVideoBench	MLVU
Standard InfoNCE	66.49	60.96
Time-aware (λ=15 s)	**67.69**	61.55
Time-aware (λ=30 s)	67.39	**61.75**
w/o audio stream	67.39	61.55

Table 6: Effect of the time-aware contrastive objective (Eq. 4) on downstream QA accuracy. λ controls the temporal decay rate of negative suppression (Eq. 3). The last row uses only the visual stream.

suggesting room for future cross-segment reasoning on top of the selected evidence.

Question Type	Uniform	TSPO	VideoSeeker
Key Information	42.86	58.89	**64.11**↑21.25
Temporal Grouding	30.81	36.97	**40.28**↑9.47
Entity Recognition	39.72	51.00	**59.04**↑19.32
Reasoning	40.82	43.37	**48.47**↑7.65
Event Understanding	39.84	44.00	**51.04**↑11.2
Summarization	32.76	34.48	**36.21**↑3.45

Table 7: Per-type accuracy (%) on LVBench under Qwen3VL-8B / 64-frame budget. Superscript gains are computed relative to uniform sampling. TSPO per-type numbers will be filled in the camera-ready version.

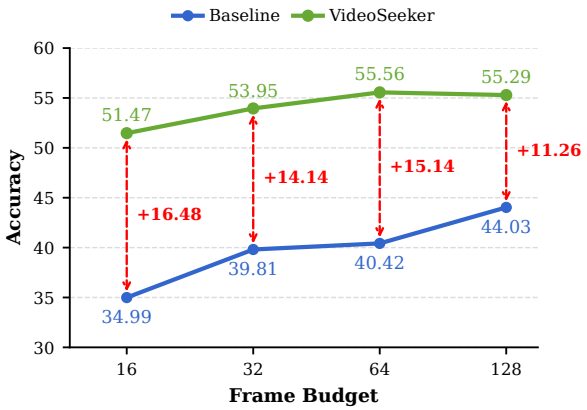

Figure 3: Accuracy vs. frame budget on LVBench. VideoSeeker consistently outperforms uniform sampling under different frame budgets.

4.8 Qualitative Examples

Representative success and failure cases are shown in Appendix F. In successful cases, the retriever proposes a diverse candidate pool, the Omni verifier removes look-alike distractors, and the allocator concentrates more frames on the verified evidence segments. In failure cases, the verifier occasionally rejects useful but indirect context (e.g., a setup shot preceding the evidence event), or the retriever misses the correct evidence entirely.

5 Conclusion

We presented **VideoSeeker**, a coarse-to-fine evidence-selection framework for long-video understanding that follows a Retrieve–Verify–Answer pipeline. A time-aware intra-video contrastive retriever—built on Qwen2.5-Omni-3B with the LM head and TTS module removed—proposes a high-recall pool of segments; the same Qwen2.5-Omni-3B model, used in inference only, verifies answer utility through calibrated Yes/No log-odds; and a score-guided allocator distributes the frame budget toward verified evidence before invoking the frozen answer VLM. This explicit decoupling of *relevance retrieval* and *answer-utility verification* yields consistent gains across four long-video QA benchmarks, four frozen backbones, and a wide range of frame budgets, offering a practical recipe for evidence selection under tight inference constraints.

Limitations

VideoSeeker has several limitations. First, the retriever relies on temporal evidence annotations, and annotation noise may affect positive/negative

construction. Second, the verification stage introduces extra inference cost; although we show in Appendix C that the additional overhead is modest with a 3B verifier, deployments under hard real-time constraints may need an even smaller verifier or batched verification. Third, the verifier is not perfectly reliable and can reject useful contextual segments or accept visually related distractors, especially on multi-hop questions. Finally, fixed-length 15-second segmentation may be suboptimal for questions requiring long-range multi-hop reasoning across distant events; adaptive or hierarchical segmentation is a promising extension.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pages 9112–9141.

Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2024. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*.

Satya Krishna Gorti, Noël Vouitsis, Junwei Ma, Keyvan Golestan, Maksims Volkovs, Animesh Garg, and Guangwei Yu. 2022. X-pool: Cross-modal language-video attention for text-video retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5006–5015.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173.

Xiangrui Liu, Yan Shu, Zheng Liu, Ao Li, Yang Tian, and Bo Zhao. 2025. Video-xl-pro: Reconstructive token compression for extremely long video understanding. *arXiv preprint arXiv:2503.18478*.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2021. Clip4clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. 2025. Video-xl: Extra-long vision language model for hour-scale video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26160–26169.

Canhui Tang, Zifan Han, Hongbo Sun, Sanping Zhou, Xuchong Zhang, Xin Wei, Ye Yuan, Huayu Zhang, Jinglin Xu, and Hao Sun. 2026. Tspo: Temporal sampling policy optimization for long-form video language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 9368–9376.

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

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Qwen Team. 2025a. Qwen2.5-vl.

Qwen Team. 2025b. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024a. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, and 1 others. 2025. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22958–22967.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, and 1 others. 2024b. Internvideo2: Scaling foundation models for multimodal video understanding. In *European conference on computer vision*, pages 396–416. Springer.

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

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pages 21875–21895.

Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. 2021. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 6787–6800.

Ziqiang Xu, Qi Dai, Tian Xie, Yifan Yang, Kai Qiu, DongDong Chen, Zuxuan Wu, and Chong Luo. 2025. Viarl: Adaptive temporal grounding via visual iterated amplification reinforcement learning. *arXiv preprint arXiv:2505.15447*.

Linli Yao, Haoning Wu, Kun Ouyang, Yuanxing Zhang, Caiming Xiong, Bei Chen, Xu Sun, and Junnan Li. 2025. Generative frame sampler for long video understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17900–17917.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *Advances in Neural Information Processing Systems*, 37:121156–121184.

Jun Zhang, Teng Wang, Yuying Ge, Yixiao Ge, Xinhao Li, Ying Shan, and Limin Wang. 2025a. Timelens: Rethinking video temporal grounding with multimodal llms. *arXiv preprint arXiv:2512.14698*.

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

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025c. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

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

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

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

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025a. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

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

A Training Data Construction

TimeLens-100K (Zhang et al., 2025a) is a large-scale audio-visual temporal grounding dataset in which each example consists of a natural-language query q, a long video V (the vast majority of which contain synchronised audio tracks), and an annotated ground-truth evidence interval $[a, b]$ specified in seconds. We construct retrieval training data from this corpus through the five-stage pipeline described below; the caption-generation prompt used in Stage 3 is reproduced in Figure 6.

Stage 1: Video Segmentation. Each video V is partitioned into non-overlapping, contiguous 15-second windows. The handling of the terminal segment follows a duration-based rule: if the remainder is shorter than 5 s, it is merged into the preceding segment, yielding a final window of between 15 s and 20 s; if the remainder is at least 5 s, it is retained as an independent segment. This strategy avoids discarding non-trivial tail content while preventing excessively short segments from entering the candidate pool, producing the final segment sequence $\{s_1, \ldots, s_N\}$ for each video.

Stage 2: ASR Transcription. Whisper Large-V3 (Radford et al., 2023) is applied to the audio channel of each segment s_i, producing a verbatim transcript t_i^{asr} that serves as a faithful text surrogate for the segment's acoustic content.

Stage 3: Visual Caption Synthesis. The video clip, the transcript t_i^{asr}, and a fixed instruction prompt (Figure 6) are jointly provided to Qwen3VL-8B (Team, 2025b), which generates a dense visual-semantic caption c_i. The two text sources are then concatenated into a single document $d_i = t_i^{\text{asr}} \parallel c_i$, encoding both the acoustic and visual content of s_i in plain text.

Stage 4: Text-Based Relevance Scoring. The text pair $(q,\, d_i)$ is submitted to Qwen3-Rerank-8B (Zhang et al., 2025c), which returns a scalar

relevance score $\rho_i \in \mathbb{R}$ for segment s_i. Routing the audio signal through ASR prior to scoring ensures that speech-centric evidence is evaluated on equal footing with visually salient content, without requiring dedicated audio embeddings.

Stage 5: Positive and Negative Sample Construction. Let $s_i = [s_i^{\mathrm{s}},\ s_i^{\mathrm{e}}]$ denote the temporal boundaries of segment s_i, and let

$$\delta_i \;=\; \min\!\big(s_i^{\mathrm{e}},\ b\big) \;-\; \max\!\big(s_i^{\mathrm{s}},\ a\big) \qquad (10)$$

denote its temporal overlap with the ground-truth interval $[a, b]$.

Positive segments. A segment qualifies as a positive candidate if $\delta_i \geq 3\,\mathrm{s}$, meaning the annotated evidence covers at least 20% of the 15-second window and the segment contains a non-trivial portion of the answer-supporting content. Because the duration of $[a, b]$ varies across queries, a single query may yield multiple qualifying segments. We allow up to K=3 positives per training instance: if the number of qualifying segments is at most 3, all of them are retained; if it exceeds 3, the three segments with the largest overlap δ_i are selected. This cap balances label richness against training efficiency while ensuring that the chosen positives are maximally aligned with the ground truth.

Negative segments. A segment enters the negative candidate pool if and only if it is temporally disjoint from $[a, b]$:

$$s_i^{\mathrm{e}} \leq a \quad \text{or} \quad s_i^{\mathrm{s}} \geq b. \qquad (11)$$

The M=6 candidates with the highest relevance scores ρ_i are selected from this pool to form the hard-negative set $\{s_1^-, \ldots, s_6^-\}$. Since these segments are semantically close to q while remaining temporally disjoint from the evidence, they directly replicate the intra-video distractors encountered at inference time. Cross-video negatives are deliberately excluded, as the retrieval task is strictly confined to a single video.

Corpus Filtering and Statistics. A video is retained only when it yields at least one positive segment satisfying $\delta_i \geq 3\,\mathrm{s}$ *and* at least M=6 negative candidates satisfying Equation (11); videos that are too short to meet both criteria simultaneously are discarded. Each surviving training instance takes the form $\big(q,\ \{s_1^+, \ldots, s_K^+\},\ \{s_1^-, \ldots, s_6^-\}\big)$, where $K \in \{1, 2, 3\}$. After filtering, the final training corpus comprises **40K** queries, each paired with 1–3 positive segments and exactly 6 hard-negative segments.

B Additional Implementation Details

Retriever. The retriever reuses the Qwen2.5-Omni-3B backbone (vision encoder, audio encoder, and LLM backbone) with the LM head and TTS module discarded. The vision and audio encoders are *frozen* during fine-tuning; only the LLM backbone is updated. Training uses AdamW with an initial learning rate of 1×10^{-5}, BF16 precision, gradient accumulation over 4 steps, and a batch size of 8 per NPU core (effective batch size 32 across 8 Ascend AI accelerator cores). We train for 1 epoch on 40K query–evidence pairs, taking approximately 24 hours. The time-aware decay parameter is λ=30 s (Eq. 3). The segment length is fixed at 15 s throughout.

Verifier. The verifier $\mathcal{V}_{\text{Omni}}$ reuses the *same* Qwen2.5-Omni-3B model as the retriever backbone, operated in inference-only mode with no additional fine-tuning. Each candidate segment is represented by at most 8 uniformly sampled frames plus the corresponding audio track. Decoding is greedy (temperature $= 0$) to ensure deterministic, reproducible log-odds scores.

Answer VLM. Frozen Qwen3VL-8B with 64-frame budget by default. Other backbones use their own preprocessors but the same budget.

Hyperparameters. K_r=30, K_s=10, fusion weight α=0.7, allocation temperature τ_f=1.0, base frames per segment $f_{\min}$=4, frame cap per segment $f_{\max}$=15, segment length 15 s.

C Efficiency Analysis

All measurements are conducted on Ascend AI accelerator under BF16 precision, averaged over all videos in LVBench; warm-up runs are excluded. VideoSeeker's pipeline decomposes into two stages of different temporal granularity: Stage 1 (video encoding) is *per-video* and can be executed offline as a precomputation step, while Stage 2 (Omni verification) must be re-run for each new QA pair.

Stage 1 – Per-video encoding and online retrieval. The video encoder embeds all N 15-second segments of a video in a single offline sweep; the resulting segment embeddings are query-agnostic and can therefore be precomputed and cached prior to serving. On LVBench, this encoding step takes 469 s per video with a peak memory footprint of 16,620 MB. At query time,

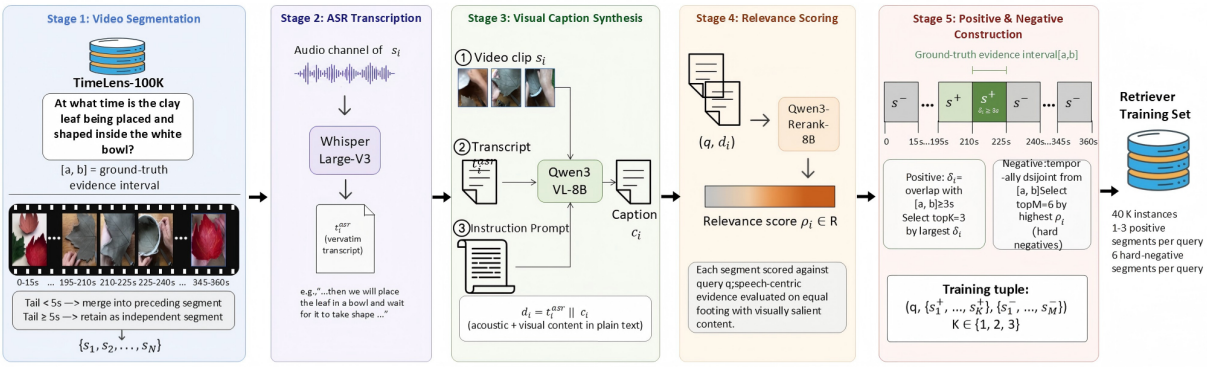

Figure 4: Training data construction for the retriever.

Algorithm 1 VideoSeeker: Retrieve–Verify–Answer

Require: Long video V, question q, frozen VLM, retriever (E_q, E_v), verifier $\mathcal{V}_{\text{Omni}}$
Require: Pool size K_r, top segments K_s, frame budget B
Require: Fusion weight α, softmax temperature τ_f, base frames $f_{\min}$, cap $f_{\max}$
Ensure: Answer $\hat{a}$

Partition V into 15s segments $\{s_1, \ldots, s_N\}$; $\mathbf{h}_q \leftarrow E_q(q)$
// Stage 1: Time-Aware Retrieval
for $i = 1, \ldots, N$ **do**
  $r_i \leftarrow \cos\big(\mathbf{h}_q,\ E_v(s_i)\big)$
end for
$\mathcal{C}_{K_r} \leftarrow \mathrm{TopK}_{K_r}(\{r_i\})$
// Stage 2: Omni Verification
for $s_i \in \mathcal{C}_{K_r}$ **do**
  $(L_i^{\text{yes}}, L_i^{\text{no}}) \leftarrow \mathcal{V}_{\text{Omni}}(q, s_i)$ ▷ LogSumExp over surface forms
  $s_i^{\text{Qwen}} \leftarrow L_i^{\text{yes}} - L_i^{\text{no}}$
end for
// Stage 3: Fusion and Adaptive Allocation
$\tilde{u}_i \leftarrow \alpha\,\mathrm{zNorm}(s_i^{\text{Qwen}}) + (1-\alpha)\,\mathrm{zNorm}(r_i),\ \forall s_i \in \mathcal{C}_{K_r}$
$\mathcal{S}_{K_s} \leftarrow \mathrm{TopK}_{K_s}(\{\tilde{u}_i\})$; $R \leftarrow B - K_s f_{\min}$
$p_i \leftarrow \mathrm{Softmax}(\tilde{u}_i/\tau_f)$ over $\mathcal{S}_{K_s}$
$m_i \leftarrow f_{\min} + \min\big(\lfloor p_i R \rfloor,\ f_{\max} - f_{\min}\big) + \delta_i$ ▷ δ_i distributes residual, $\sum_i m_i = B$
$\mathcal{F} \leftarrow \bigcup_{s_i \in \mathcal{S}_{K_s}} \mathrm{UniformSample}(s_i, m_i)$
$\hat{a} \leftarrow \mathrm{VLM}(q, \mathcal{F})$
return $\hat{a}$

cosine-similarity ranking over the cached embeddings adds only 0.04 s and 9,073 MB per QA pair—effectively negligible. This amortisation property is practically important: in any deployment scenario where multiple questions are posed against the same video (e.g., a lecture archive, a long movie, or a surveillance recording), the per-video encoding cost is paid once and all subsequent queries are served at near-zero retrieval latency.

Stage 2 – Batched Omni verification. Qwen2.5-Omni-3B is the dominant per-QA cost, evaluating each of the K_r candidate segments returned by

Batch	Latency (s/QA)	Peak mem. (MB)	LVBench (%)
1	94.29	13,832	55.56
2	81.91	17,000	55.23
4	**66.23**	23,524	55.43
8	69.00	36,096	55.16

Table 8: Effect of verification batch size on latency, peak memory, and LVBench accuracy (K_r=30, Ascend AI accelerator, BF16).

Stage 1. Because the K_r segments are independent given the question, their forward passes can be merged into a single batched call to amortise NPU kernel-launch overhead without altering the per-segment output logits in principle; the minor accuracy fluctuations across batch sizes reported in Table 8 span at most 0.40 points and are attributable to floating-point accumulation order under BF16.

Batch size 4 achieves the lowest latency (66.23 s/QA) at a peak memory of 23,524 MB, and maintains accuracy within 0.13 points of the serial baseline (55.56%). Scaling to batch size 8 raises latency by 2.77 s and peak memory to 36,096 MB without yielding any accuracy benefit; batch size 16 triggers an out-of-memory error.

Comparison with AKS. AKS uses a CLIP model to score individual frames against the query and must re-execute this scoring for every new QA pair, incurring 77.88 s and 1,127 MB per query. With precomputed video embeddings, VideoSeeker's per-QA latency at batch size 4 is 66.23 s—a **14.9%** reduction over AKS—while processing richer 15-second multimodal segments rather than isolated frames. Moreover, because VideoSeeker's segment embeddings are query-agnostic, they are trivially cacheable: once a video

has been indexed, any subsequent query is served by Stage 1 in 0.04 s, incurring only the Stage 2 verification cost.

Take-away. VideoSeeker's one-time per-video encoding cost (469 s) is amortised across all queries for the same video, making the framework increasingly cost-efficient as the number of queries per video grows. Under an offline pre-indexing deployment, the dominant per-QA cost is Omni verification, which at batch size 4 amounts to 66.23 s—already 14.9% faster than AKS (77.88 s)—while delivering substantially higher long-video QA accuracy (see Table 3 in the main text). Batch size 4 therefore represents the recommended default, providing the best latency–memory trade-off within the Ascend AI accelerator device budget.

D Additional Ablations on Retrieval and Verification Hyperparameters

Stage 1: Retrieval pool size K_r. Table 9 ablates the size of the candidate pool returned by the time-aware retriever. A small pool (K_r=10) yields high verification efficiency but insufficient recall: the ground-truth evidence segment is absent from the pool for a non-trivial fraction of questions, creating a performance ceiling that no downstream verification can overcome. Increasing K_r raises recall and accuracy, but with diminishing returns beyond K_r=30: the gain from K_r=30 to K_r=50 is marginal while verification latency grows linearly. We therefore adopt K_r=30 as the default, which provides the best accuracy–efficiency trade-off.

K_r	Coverage@K_r (%)	Verif. Lat. (s)
10	32.03	943
20	41.13	1886
30	48.25	2829
40	53.75	3772
50	58.95	4715

Table 9: Ablation of the retrieval pool size K_r (K_s=10, B=64). Recall@K_r measures whether the ground-truth evidence segment appears in the top-K_r pool. Accuracy saturates around K_r=30 while verification latency grows linearly, motivating our default choice.

Stage 3: Number of post-verification segments K_s. After score fusion, VideoSeeker retains the top-K_s segments for adaptive frame allocation. Table 10 sweeps $K_s \in \{5, 10, 15, 20\}$ under a fixed frame budget B=64 and K_r=30. K_s=10 achieves the best result (64.15): each segment receives on average 6.4 frames, which is sufficient to sample the critical visual details within a 15-second window. Beyond K_s=10, the per-segment frame count drops below 4 on average, causing each segment to be represented by too few frames to reliably capture its visual content, and accuracy decreases.

K_s	Frames/Seg. (avg.)	Avg. Acc.
5	12.8	63.91
10	**6.4**	**64.15**
15	4.3	63.68
20	3.2	63.18

Table 10: Ablation of the number of post-verification segments K_s (K_r=30, B=64, Qwen3VL-8B). K_s=10 achieves the best accuracy by balancing evidence coverage against per-segment frame density.

E Verifier Prompt

We use the following verification prompt:

```
You are a strict video retrieval reranking judge. Given a user question and a video clip, decide whether the clip contains information useful for answering the question. Do not answer the question itself. Output "yes" if the clip contains visual, audio, speech, subtitle, on-screen text, or contextual evidence that can help answer the question. Output "no" if the clip does not contain useful evidence for answering the question, or if relevance would require guessing. Your entire response must be exactly one lowercase English word: yes or no.
```

F Additional Qualitative Case Studies

We present two additional qualitative cases comparing VideoSeeker against the TSPO baseline(Figure 5). For each case, we show the frames retrieved by TSPO and VideoSeeker respectively, along with their corresponding answers. In both cases, TSPO selects frames that are query-related but fail to capture the answer-critical moments, leading to incorrect predictions. VideoSeeker, by contrast, retrieves answer-supporting frames through Omni verification, yielding correct responses on both fine-grained visual attribute questions and intention-level reasoning questions.

Benchmark	Split	#QA	Video Length (min)		Evidence Length (s)		Sparsity (%)
			Avg.	*Med.*	*Avg.*	*Med.*	
LVBench	—	1,492	67.0	60.3	381.7	17.0	**0.4**
Video-MME†	Short	900	1.4	1.4	28.8	26.0	31.5
	Medium	900	9.4	8.1	160.0	164.7	29.3
	Long	900	39.8	40.3	967.7	890.7	37.3
MLVU‡	—	502	15.6	8.3	—	—	—
LongVideoBench‡	—	1,337	7.9	3.5	—	—	—

Table 11: Per-benchmark dataset statistics supporting the evidence-sparsity analysis in §4.2. **Avg. V.L.** and **Med. V.L.**: average and median video length, respectively. **Avg. C.L.** and **Med. C.L.**: average and median evidence (clue-interval) length per QA pair. **Sparsity** = Med. C.L. / Avg. V.L., expressing the typical fraction of the video that contains answer-supporting evidence; smaller values indicate sparser evidence. †Evidence intervals are annotated only for a random sample of three QA pairs per video subtype. ‡No publicly annotated ground-truth evidence intervals are available; evidence length and sparsity are therefore not computable.

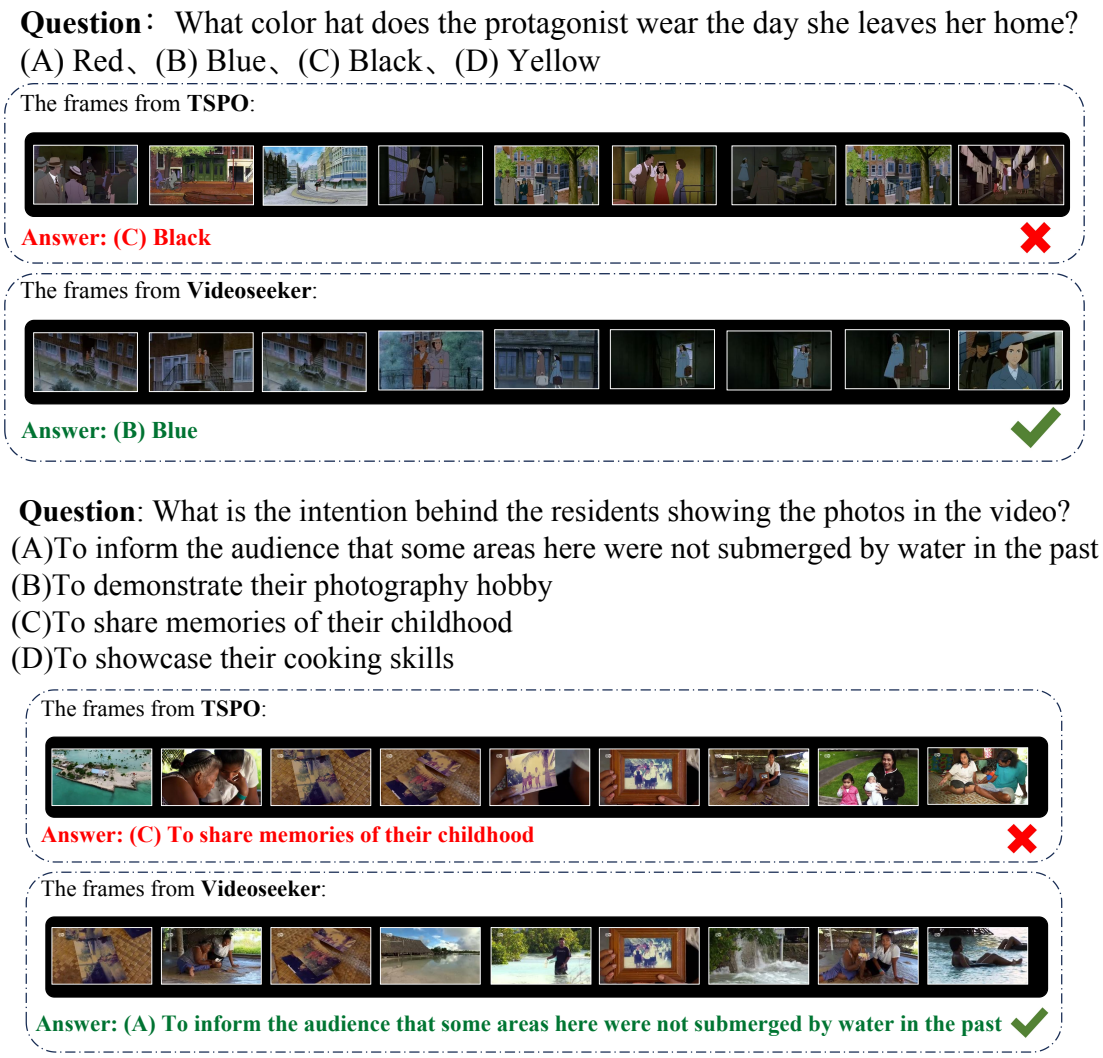

Figure 5: Additional qualitative success cases. VideoSeeker improves over uniform sampling by verifying whether retrieved segments are answer-supporting rather than merely query-related, enabling accurate temporal counting and key-information retrieval.

Caption Generation Prompt (Stage 3, Appendix A)

You are an expert video content analyzer for a search engine database. Your task is to generate a dense, visually grounded, retrieval-oriented caption for a video clip. The caption will be used for downstream search and reranking, so it should maximize distinctive, concrete, and query-relevant details from both the visual content and the audio transcript.
Guidelines: 1. Visual grounding first: Describe what is clearly visible in the clip with high specificity. Focus on people, clothing, colors, patterns, objects, scene type, background structures, on-screen text, camera perspective, and changes across time. Use precise nouns and verbs rather than vague summaries.
2. Retrieval-oriented detail: Prioritize details that would help a search system distinguish this clip from similar clips. Include attributes such as color, size, texture, material, quantity, relative position, and notable visual identifiers when visible.
3. Action and temporal order: Describe actions concretely and in sequence. State who does what, how, with which object, in what direction, and with what result. If the clip contains multiple shots or transitions, reflect that temporal flow clearly.
4. People and appearance: When people are visible, describe their apparent gender presentation only if visually evident, approximate age group if clear, clothing type, clothing color, patterns, accessories, hairstyle, posture, and interaction with objects or other people. Do not guess identity or relationships unless explicitly supported by the video or transcript.
5. Objects and environment: Name objects as specifically as possible. Describe the setting precisely, such as kitchen, classroom, basketball court, supermarket aisle, train platform, office desk, beach walkway, snowy street, or stage. Include prominent background elements and readable text or logos on screen.
6. Audio / transcript usage: If a transcript is provided, use it to infer context, clarify ambiguous actions, and capture the semantic content of speech. For longer transcript segments, do not copy everything verbatim. Instead, compress the speech into its key meaning while preserving search-relevant terms, topics, commands, reactions, or dialogue content. When possible, align spoken content with visible actions or moments in the clip.
7. No-speech case: If the transcript is exactly "[No speech detected]", do not mention audio, silence, or the absence of speech. Rely exclusively on visual evidence and increase the density of visual details.
8. Faithfulness: Do not invent facts that are not visually present or clearly supported by the transcript. Avoid speculation about names, locations, professions, events, or intentions unless explicitly grounded.
9. Style: Write one single coherent paragraph in natural language. Do not use bullet points. Do not use filler phrases such as "The video shows" or "This clip contains." Keep the writing dense, concrete, and information-rich.
Output requirement: Return exactly one paragraph that densely describes the clip, integrating visual content first and transcript-based context when relevant.
Few-Shot Examples: **Example 1:** *Input Transcript:* "The Code of Hammurabi tells us an immense amount about Babylonian society." *Output Description:* A rapid sequence of animated educational icons appears on a white background, featuring a white classical profile in a grey oval, a speech bubble in a blue oval, a columned building in a green oval, and a stack of colorful books in a yellow oval. This resolves into a cyan "No. 86" logo above the cursive text "Continue the Conversation". A subsequent title card displays a large red oval with white text: "The Code of Hammurabi the Rule of Law: Why Written Law Matters," credited to "Paul Rahe, Professor, Hillsdale College." The visual style then shifts to a dark blue background with beige line-art animation. Text overlays read "c. 1754 BC" in the top left and "Hammurabi, King of Babylon" in the top right. On the left, two figures in simple tunics hold and unroll a large vertical scroll covered in cuneiform-style script. On the right, a larger bearded figure, identified as King Hammurabi, sits on an ornate throne wearing a patterned robe and horned headdress, raising his right hand in a declarative gesture. The narrator's statement that the Code "tells us an immense amount about Babylonian society" directly contextualizes this animated depiction of the ancient legal text being presented before the Babylonian king.
Example 2: *Input Transcript:* "Commonwealth record holder of 865. And Michael Masso of Cuba, fourth in the world and youth junior championships. Still a junior now, he's only 18 years of age." *Output Description:* Inside a packed athletics stadium featuring a purple track and blue "TDK" advertising boards, a line of male track and field competitors is introduced to the crowd. The camera first focuses on a bald South African athlete wearing a green and yellow sleeveless top reading "SOUTH AFRICA." He points both index fingers forward before raising his arms straight up in acknowledgment. An on-screen lower-third graphic displays his name, "Luvo MANYONGA," the South African flag, and "8.65 PB," which perfectly aligns with the announcer's voiceover introducing him as the "Commonwealth record holder of 865." The camera then pans right to the next athlete in line, Maykel Massó, who wears a red Puma tank top with "CUBA" printed in white. As the announcer introduces him by name, noting he is "fourth in the world" and "only 18 years of age," Massó raises his arms to clap above his head before lowering them. A corresponding graphic displays the Cuban flag, his name, and "8.33 PB". Finally, the camera shifts to Jarrion Lawson, an American

athlete in a blue Nike "USA" tank top, who stands stoically as his name and "8.58 PB" appear on the overlay. Several other competitors, including one in a red Chinese jersey, wait in a single-file line in the background, while a track official in a white shirt and black pants sits on a chair to the left.
Example 3: *Input Transcript:* "[No speech detected]" *Output Description:* The video captures the start and early progression of a men's 100-meter sprint at the London 2012 Olympic Games, as indicated by the purple track branding and the on-screen graphic displaying the World Record of 9.58 and the Olympic Record of 9.69. Eight athletes are initially seen crouched in their starting blocks on the red track, wearing distinct national uniforms. From the inside lane closest to the bottom of the screen to the outside lane at the top, the athletes are wearing a green top with dark shorts, a white top with red shorts, a white top with dark shorts, a light blue top with dark shorts, a red top with dark shorts, a vibrant yellow, green, and black uniform, another white top with dark shorts, and finally a black top with dark shorts on the far outside. As the race begins, all sprinters explode from the blocks simultaneously, and the camera pans right to track their acceleration down the straightaway. As the runners reach their full stride and the on-screen timer approaches the 8.0-second mark, the athlete in lane seven wearing the yellow, green, and black uniform clearly pulls ahead of the pack, establishing a dominant lead over the rest of the field just before the video concludes.
Current Task: *Input Transcript:* "transcript$_{p}art$" $*OutputDescription:*$

Figure 6: Instruction prompt provided to Qwen3VL-8B in Stage 3 of the data construction pipeline. Prior to inference, the placeholder {asr_transcript} is replaced with the Whisper Large-V3 transcription t_i^{asr} of the corresponding 15-second segment.