DREAMSEEK: IMAGINATION-GUIDED EVIDENCE ACQUISITION FOR LONG-VIDEO QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Finding decisive evidence in a long video can require processing many irrelevant frames. Frame selection reduces the answerer's input, while relevance-guided search uses sparse observations to locate promising regions. However, these observations also provide temporal context for predicting unread visual features, offering guidance beyond measured relevance. We introduce DREAMSEEK , an imagination-guided approach whose question-agnostic temporal world model predicts unread visual features, enabling relevance assessment before real acquisition. New observations refine the predicted timeline and provide relevance corrections that persist across search rounds. This process enriches search guidance through low-cost latent computation, reducing expensive decoding and encoding. With a shared answerer, DREAMSEEK achieves the highest mean accuracy on three of five long-video QA benchmarks and up to 5.1× evidence-acquisition speedups over strong frame-selection baselines.

1 INTRODUCTION

A longstanding challenge in long-video QA is to acquire sufficient evidence for reliable answers without processing large amounts of irrelevant visual content Ye et al. (2025). Such evidence may lie in a brief action, span an activity, or connect events separated in time Wu et al. (2024); Zhou et al. (2025); Chandrasegaran et al. (2024). Uniform sampling does not adapt to these requirements and can miss relevant observations Tang et al. (2025). Increasing the frame budget can improve coverage but raises decoding and encoding costs Schneider et al. (2026); Liu et al. (2025). Irrelevant frames also consume the answerer's limited visual context and can reduce accuracy Liu et al. (2025).

Recent frame selection methods retain a question-relevant subset for the answerer, but still process candidates sampled across the video Liu et al. (2025); Tang et al. (2025). Adaptive evidence search reduces this upfront cost by using sparse observations to guide further inspection Zhu et al. (2026); Qiu et al. (2026). However, methods such as FOCUS treat observed frames mainly as relevance measurements Zhu et al. (2026), overlooking their value as context for predicting unread content. How can a few observed frames help us locate evidence that none of them directly shows?

Sparse observations provide temporal context for predicting unread visual content Bardes et al. (2024), even when they offer little direct answer evidence. This creates an opportunity to assess relevance before real acquisition. However, realizing it poses three challenges. *(i) Predictive ambiguity.* Partial observations can support multiple plausible events with uncertain temporal locations Vondrick et al. (2016), so predicted relevance does not establish that the event occurred. *(ii) Evidence sufficiency.* Further observations can reduce ambiguity, but where to acquire them and when to stop depend on the question, existing evidence, and the need for local detail and broader temporal context. *(iii) Computational cost.* Prediction and verification overhead can offset the savings from selective acquisition.

To address these challenges, we propose DREAMSEEK for imagination-guided evidence acquisition in long-video QA (Table 1). A question-agnostic temporal world model predicts unread visual features from sparse observations. The question queries these predictions to guide acquisition, while real observations refine them and provide persistent relevance corrections. The search stops adaptively and supplies only real frames to the answerer, combining targeted evidence with temporal coverage. Across five long-video QA benchmarks Ye et al. (2025); Fu et al. (2025); Wu et al.

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.

Table 1: Qualitative comparison of evidence-acquisition methods. Full-video encoding refers to processing the complete candidate pool before selection. Visual prediction estimates unread content in pixel space or feature space.

Method	No upfront full-video encoding	Predicts unread visual content	Explicit temporal priority map	No VLM fine-tuning
BOLT (Liu et al., 2025)	×	×	✓	✓
AKS (Tang et al., 2025)	×	×	✓	✓
FOCUS (Zhu et al., 2026)	✓	×	✓	✓
TimeSearch-R (Pan et al., 2025)	✓	×	×	×
VAP (Ma et al., 2026)	×	✓	✓	✓
DREAMSEEK (Ours)	✓	✓	✓	✓

(2024); Zhou et al. (2025); Wang et al. (2025a), DREAMSEEK achieves the highest mean accuracy on three with a shared answerer, exceeding the strongest baseline by up to 1.26 percentage points. At a delivery budget of $K = 64$ frames, it also delivers 1.8–5.1× evidence-acquisition speedups over the evaluated frame-selection baselines. Our contributions are:

- We propose imagination-guided evidence search that makes unread video searchable before real acquisition.
- We propose a question-agnostic temporal world model that combines feature reuse and learned prediction in a shared visual-text space without question-answer supervision.
- We propose an observation-corrected search loop that retains real-frame relevance feedback as predictions evolve.

2 RELATED WORK

Efficient long-video representations and frame sampling. Existing methods construct compact answering inputs by compressing or hierarchically organizing representations of observed video Shen et al. (2025); Wang et al. (2025b), or selecting question-relevant frames Yu et al. (2025); Liu et al. (2025); Tang et al. (2025). BOLT shapes a relevance-based sampling distribution, while AKS balances relevance and coverage across temporal bins. Both still decode and encode all sampled candidates before selection. DREAMSEEK reduces this upfront cost by estimating relevance from predicted visual features before real acquisition.

Adaptive and agentic evidence search. Adaptive search uses partial observations to guide further inspection. T* interpolates observed relevance scores Ye et al. (2025). FOCUS combines clip relevance and uncertainty with nearest-neighbor propagation of observed scalar scores within selected clips Zhu et al. (2026). Agentic reasoning guides temporal-window retrieval or refinement Pan et al.; Yuan et al. (2025) and further visual access Pan et al. (2025); Zeng et al. (2026); Qiu et al. (2026); He et al. (2025). However, TimeSearch-R and Video-o3 require VLM fine-tuning through supervised and reinforcement learning Pan et al. (2025); Zeng et al. (2026). Reported search failures also include premature stopping and commitment to visually similar but irrelevant regions Pan et al. (2025); Qiu et al. (2026). DREAMSEEK learns acquisition guidance in a lightweight temporal world model, avoiding task-specific VLM fine-tuning. It preserves real-observation corrections as predictions evolve, so earlier verification continues to guide search.

Predictive visual representations for evidence acquisition. Temporal structure in latent visual features supports predicting future or missing representations from observed context Vondrick et al. (2016); Tang et al. (2026); Bardes et al. (2024). VAP uses generated expectations for video QA, selecting encoded real candidates in one round by their discrepancy from predictions conditioned on initial frames, the question, and possible answers Ma et al. (2026). DREAMSEEK instead uses prediction to estimate question relevance before acquisition. Repeated observations then refine these predictions and guide further search.

3 PROBLEM FORMULATION

Evidence localization under partial observation. Given a long video V of duration L and a question q, we define the evidence $\mathcal{E}(q, V) \subseteq [0, L]$ as one or more temporal regions whose visual

content supports the correct answer. We consider a from-pixels setting with unknown evidence locations and no dense visual index. Let x_t be the frame at time t and $z_t = E_{\text{img}}(x_t)$ its visual feature. For observed timestamps Ω, the available observations are $\mathcal{O} = \{(t, z_t) : t \in \Omega\}$. Features outside Ω remain unobserved. Acquiring these observations requires decoding and encoding the selected frames. The goal is to control this cost while acquiring sufficient real evidence for accurate answering, rather than recovering the exact boundary of $\mathcal{E}(q, V)$.

Predictive evidence acquisition. If the complete feature sequence were available, we could rank potential evidence locations by similarity to the question representation $u = E_{\text{text}}(q)$ in a shared image-text embedding space. Under partial observation, we posit that temporal regularities allow $\mathcal{O}$ to support a proposal feature $\widetilde{z}_t$ at an unobserved timestamp $t \notin \Omega$. The proposal is inferred without the question as input, but its similarity to u gives a question-specific score for prioritizing visual access. It need not reconstruct z_t exactly, but should preserve semantic cues useful for ranking. Predicted features guide acquisition and do not constitute observed evidence. The answerer uses the question and real frames acquired from the video.

4 DREAMSEEK

DREAMSEEK uses *latent imagination* to guide real-video acquisition, alternating temporal feature completion with question-guided search and verification (Figure 1). New observations refresh completion while their relevance corrections persist across rounds. After search, the final belief guides real-frame delivery to the answerer.

4.1 TEMPORAL WORLD MODEL

Observed actions or states can constrain what precedes or follows them, even when related events differ in appearance. Our temporal world model uses this context for *temporal completion* in the feature space of E_{img}. We partition a video V into T temporal segments with representative timestamps $\{t_i\}_{i=1}^{T}$. For each completion, real observations at timestamps $\mathcal{A} \subseteq \Omega$ serve as anchors. Given their features, the model with parameters θ predicts

$$\widetilde{Z} = F_\theta\left(\{(t, z_t) : t \in \mathcal{A}\}, T\right), \qquad \widetilde{Z} \in \mathbb{R}^{T \times d}, \tag{1}$$

where $z_t = E_{\text{img}}(x_t)$ is the visual feature from the observed anchor frame x_t. The ith row of $\widetilde{Z}$ is the proposal feature $\widetilde{z}_{t_i}$, and d is the visual feature dimension.

To perform completion, we place anchor features at their temporal positions and learned mask tokens elsewhere. The temporal network processes the sequence jointly, using anchor features and their temporal arrangement to represent each position. When frames at different times are visually similar, observed features can help complete unread positions. Our *pointer-mixture predictor* therefore combines direct prediction with anchor reuse. A regression head maps each representation to a predicted visual feature $\widehat{z}_{t_i}$. A pointer head assigns nonnegative weights $p_{i,t}$ to the anchors and $p_{i,\text{NONE}}$ to the regression output, with NONE denoting direct prediction. These weights sum to one, giving the final proposal

$$\widetilde{z}_{t_i} = \sum_{t \in \mathcal{A}} p_{i,t} z_t + p_{i,\text{NONE}} \widehat{z}_{t_i}. \tag{2}$$

Completion produces proposals at every temporal position without additional visual access, while $\mathcal{O}$ retains the real observations.

Training. We train with masked self-supervision, predicting hidden video features from the remaining anchors. Varying anchor density and placement exposes the model to both scattered and clustered observations. Supervision applies only to hidden positions. Regression fits the direct prediction $\widehat{z}_{t_i}$ to the true feature z_{t_i}. The pointer target is the most similar anchor when its cosine similarity to z_{t_i} meets a threshold, and NONE otherwise. To discourage indiscriminate anchor copying, contrastive learning aligns the mixture $\widetilde{z}_{t_i}$ with its target and separates it from features in other training sequences. All training targets are derived from video features. Appendix A provides the architecture, objective definitions and implementation settings.

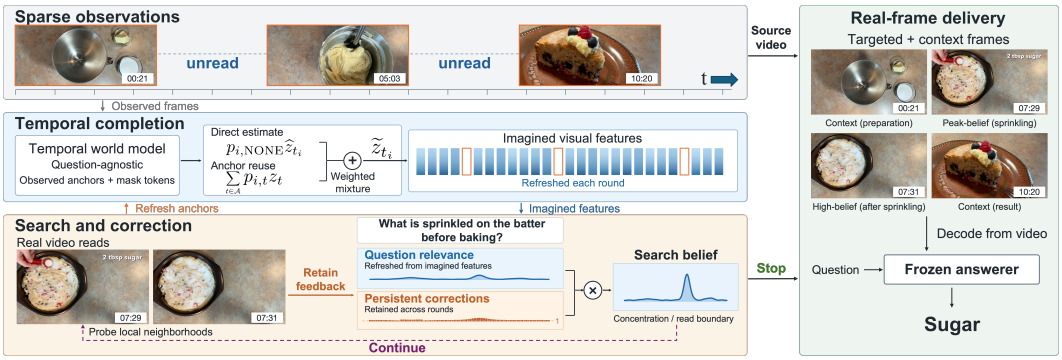

Figure 1: **Overview of DreamSeek.** From sparse observations, a question-agnostic temporal world model combines direct feature prediction with anchor reuse. Predicted relevance is multiplied by persistent observational corrections to form a search belief that guides new reads and anchor updates. After stopping, targeted and context frames from the source video accompany the question to a frozen answerer.

4.2 Iterative Evidence Search and Verification

Search begins by decoding representative frames from sparse, evenly spaced segments and encoding them with E_{img}. These observations form the initial anchors, $\mathcal{A}^{(0)} = \Omega^{(0)}$. At the start of round r, the temporal world model completes the timeline from $\mathcal{A}^{(r)} \subseteq \Omega^{(r)}$ using Eq. equation 1. Comparing the predicted features with the question representation u gives their predicted relevance:

$$\rho_i^{(r)} = \max\left\{0, \cos\left(\widetilde{z}_{t_i}^{(r)}, u\right)\right\}. \tag{3}$$

Search belief. Sparse anchors can leave long intervals weakly constrained. Predicted features may then match the question even when the real frames at those timestamps do not. We correct search priorities using relevance feedback from real observations, retaining it across refreshed completions. Let $\ell^{(r)}$ be the accumulated log-correction field, initialized by $\ell_i^{(0)} = 0$. Combining it with the current predicted relevance gives the unnormalized search belief:

$$b_i^{(r)} = \rho_i^{(r)} \exp\left(\ell_i^{(r)}\right). \tag{4}$$

A positive log correction scales the proposal up, while a negative one scales it down.

Verification and correction. To select regions for verification, we smooth the search belief across neighboring segments and identify separated peaks (local maxima). For a peak at segment index c, let $\mathcal{S}_c$ contain the indices of its surrounding region. The region is unchecked if no timestamp t_i with $i \in \mathcal{S}_c$ has been observed. We verify every unchecked region by decoding frames at its peak and nearby segment timestamps, then encoding them with E_{img}. Reading this neighborhood accommodates imprecision in the predicted evidence location. The maximum cosine similarity between these real features and u gives the region's observed relevance score.

Local reads may miss evidence or produce incidental matches. We limit the influence of each verification by mapping its observed relevance to a bounded positive factor $\gamma_c^{(r)}$. This factor equals one at a fixed reference similarity, exceeds one for stronger scores, and falls below one for weaker scores. Let $\mathcal{C}^{(r)}$ contain the indices of peaks verified in round r. We update the accumulated correction as

$$\ell_i^{(r+1)} = \ell_i^{(r)} + \sum_{c \in \mathcal{C}^{(r)}} \mathbf{1}\{i \in \mathcal{S}_c\} \log \gamma_c^{(r)}. \tag{5}$$

Each correction applies throughout $\mathcal{S}_c$, extending beyond the narrower neighborhood actually read.

Refreshing completion. We retain all acquired observations in $\mathcal{O}^{(r+1)}$. After updating $\ell^{(r+1)}$, we reweight the current proposal and use real features at observed peaks as the next anchors $\mathcal{A}^{(r+1)}$. Peak frames represent candidate regions, while neighboring reads support local verification and accumulated relevance feedback. These anchors condition a fresh completion, with $\ell^{(r+1)}$ retained

across rounds. Each new proposal replaces its predecessor, so repeated predictions are not accumulated as additional evidence.

4.3 PLAYOFF AND ADAPTIVE SEARCH STOPPING

The *playoff* refines relevance estimates within already checked regions by probing moments that remain unread. It runs when search continues but no current candidate region is unchecked. For each current peak, we look for an unread segment within a fixed neighborhood narrower than the correction region. If one is available, we read that segment and its immediate temporal neighbors. As in verification, the maximum cosine similarity between these real features and the question representation u determines the correction factor. We include the probed peaks in the verification update of Eq. equation 5 to reinforce or suppress their surrounding regions. The next completion can reveal new unchecked regions, which take priority over further playoff reads.

Adaptive stopping. Our stopping heuristic measures the share of total belief near the strongest peak. Let $c_\star^{(r)}$ be the segment index with the highest smoothed belief. For positive total belief mass, the search concentration is

$$\kappa^{(r)} = \frac{\sum_{i \in \mathcal{S}_{c_\star^{(r)}}} b_i^{(r)}}{\sum_{i=1}^{T} b_i^{(r)}}. \tag{6}$$

The sums use unsmoothed segment beliefs.

After each completion, including the initial one, we form the current belief and check stopping before verification or playoff reads. Search ends if $\kappa^{(r)}$ reaches a threshold τ, if the number of acquired search observations $|\Omega^{(r)}|$ reaches a boundary B, or if neither acquisition rule has an eligible unread location. The read boundary is checked between rounds, so a batch can take the total beyond B. The final belief then guides real-frame delivery.

4.4 FINAL FRAME DELIVERY

DREAMSEEK combines belief-guided targeting with temporal coverage to supply informative frames and broader context to the answerer. We group consecutive segments into delivery windows, each scored by its maximum segment belief. For a fixed frame budget K, we select $\lfloor K/3 \rfloor$ evenly distributed context windows and fill the remaining budget with the highest-scoring unselected windows.

Optionally, DREAMSEEK$_{\text{ad}}$ keeps the context windows and treats all other windows as candidates for targeted delivery. It adds these candidates in descending score order until their cumulative score reaches a fraction $\eta \in (0, 1]$ of the candidate pool's total score. Its targeted count m varies across questions, so the total $\lfloor K/3 \rfloor + m$ frames can exceed K. If the candidate pool has zero total score, we use uniform delivery with budget K. For either policy, we decode one real frame at each selected window's midpoint for the answerer, even if that timestamp was not read during search. Adaptive delivery leaves the search unchanged.

5 EXPERIMENTS

Training and implementation. We use frozen SigLIP (Zhai et al., 2023) for visual and question encoding. The deployed TWM has approximately 26.9M parameters and is trained self-supervised on pre-extracted video features. All benchmark videos are excluded from TWM training, which uses no QA supervision. Training runs for 32,000 steps on a single NVIDIA GH200 120GB GPU. This provides a lightweight training alternative to TimeSearch-R, which reports VLM fine-tuning through supervised and reinforcement learning on 32 A100 GPUs (Pan et al., 2025). Appendix A gives the training corpora, optimization, and inference settings.

Benchmarks and baselines. We evaluate DREAMSEEK on five long-video QA benchmarks: LV-Haystack (LVH) (Ye et al., 2025), Video-MME long (Fu et al., 2025), Haystack-LVBench (HLVB) (Wu et al., 2024; Pan et al., 2025), MLVU (Zhou et al., 2025), and LVBench (Wang et al., 2025a). The evaluated videos span approximately 6–60 minutes in LVH, 30–60 minutes in Video-MME long, and 15–60 minutes in HLVB, with MLVU and LVBench extending beyond two hours.

Table 2: Long-video QA accuracy (%). Best shared-answerer means per budget are bold. ‡ Released-system evaluations retain native protocols. TimeSearch-R is in-domain on LVH. ‖ Different evaluation split (Appendix B).

Method	LVH	Video-MME long	HLVB	MLVU	LVBench
(a) Shared-answerer methods at $K = 64$					
Uniform	52.80 ± 0.41	54.37 ± 1.70	60.62 ± 1.99	71.33 ± 0.31	44.70 ± 0.21
BOLT ($\alpha = 3$)	56.21 ± 0.83	56.07 ± 1.05	$\mathbf{64.72 \pm 1.99}$	$\mathbf{75.28 \pm 0.26}$	49.90 ± 0.62
AKS (L5, $t = 0.8$)	56.94 ± 0.58	55.48 ± 0.83	62.87 ± 1.55	73.84 ± 0.33	51.29 ± 0.25
FOCUS	56.93 ± 0.46	56.30 ± 0.95	63.45 ± 0.51	73.24 ± 0.41	52.36 ± 0.21
DREAMSEEK	$\mathbf{57.40 \pm 0.38}$	$\mathbf{57.56 \pm 0.71}$	64.60 ± 1.10	74.96 ± 0.36	$\mathbf{53.58 \pm 0.46}$
(b) Native-system and published results (different protocols)					
TimeSearch-R	$59.39^{‡}$	$55.33^{‡}$	$50.88^{‡}$	$71.57^{‡}$	$44.81^{‡}$
Video-R1	$46.15^{‡}$	$50.22^{‡}$	$50.88^{‡}$	$65.73^{‡}$	$40.56^{‡}$
Video-o3	$48.53^{‡}$	$51.33^{‡}$	$48.54^{‡}$	$65.87^{‡}$	$43.99^{‡}$
LongVideo-R1 (updated, Qwen3-8B agent)	–	58.0	–	$71.3^{‖}$	$60.7^{‖}$
ZoomV (InternVL2.5-8B)	–	53.9	$63.3^{‖}$	$70.0^{‖}$	$51.5^{‖}$
VideoExplorer (7B + 32B)	–	–	–	$58.6^{‖}$	$51.4^{‖}$
VAP (GPT-4o)	–	55.7	–	$60.3^{‖}$	–
FrameThinker (Qwen2.5-VL-7B)	–	47.6	$52.9^{‖}$	$59.1^{‖}$	$36.6^{‖}$
(c) Shared-answerer methods at $K = 32$					
Uniform	51.00 ± 0.35	52.74 ± 1.48	56.92 ± 1.21	65.69 ± 0.85	41.07 ± 0.17
BOLT ($\alpha = 3$)	54.07 ± 0.80	53.52 ± 0.97	60.23 ± 2.05	72.37 ± 0.25	45.41 ± 0.64
AKS (L5, $t = 0.8$)	55.92 ± 0.35	54.44 ± 0.12	$\mathbf{62.28 \pm 0.59}$	71.10 ± 0.07	$\mathbf{50.53 \pm 0.99}$
FOCUS	55.27 ± 0.60	54.22 ± 2.60	61.89 ± 0.68	70.15 ± 0.05	49.52 ± 0.53
DREAMSEEK	$\mathbf{56.69 \pm 0.38}$	$\mathbf{54.89 \pm 0.11}$	62.03 ± 0.62	$\mathbf{73.05 \pm 0.63}$	49.71 ± 0.40

We replicate BOLT (Liu et al., 2025), AKS (Tang et al., 2025), and FOCUS (Zhu et al., 2026), alongside a uniform-sampling baseline. For TimeSearch-R (Pan et al., 2025), Video-R1 (Feng et al., 2025), and Video-o3 (Zeng et al., 2026), we evaluate their released models using method-specific inference pipelines. Table 2 also includes published results and marks evaluation-split differences.

Evaluation protocol. DREAMSEEK and the frame-selection baselines use the same frozen Qwen3-VL-8B-Instruct answerer (Team, 2025), answering prompts, frame preprocessing, and greedy decoding. Evaluation uses the same questions and matched nominal delivery budgets of $K = 64$ and $K = 32$. Here, K counts real frames allocated to the answerer, distinct from frames read during search. Accuracy is reported as mean $\pm$ sample standard deviation across evaluation seeds. Appendix B details comparison protocols, ablation configurations, localization analyses, and acquisition accounting.

5.1 MAIN LONG-VIDEO QA RESULTS

Among shared-answerer methods, DREAMSEEK achieves the highest mean accuracy on three of five benchmarks at both $K = 64$ and $K = 32$, outperforming uniform sampling and FOCUS throughout (Table 2). It remains competitive with BOLT and AKS, with lower seed variability than the question-guided baselines in most comparisons. With a fixed frame budget, irrelevant frames displace useful evidence. The gains may reflect the combination of targeted acquisition and evenly spaced delivery, which retains context that relevance scores may overlook.

At $K = 64$, DREAMSEEK exceeds released Video-R1 and Video-o3 in QA accuracy on all five benchmarks and TimeSearch-R on four, except in-domain LVH. DREAMSEEK uses no task-specific VLM fine-tuning. Among published results, the updated LongVideo-R1 reports higher accuracy on Video-MME long (58.0% versus 57.56%) and LVBench (60.7% versus 53.58%), although its LVBench result uses a different split. These comparisons involve complete systems with their respective answerers and inference protocols.

5.2 EVIDENCE ACQUISITION AND DELIVERY EFFICIENCY

At nominal $K = 64$, DREAMSEEK achieves 1.8–5.1$\times$ evidence-acquisition speedups over the shared-answerer selectors in Table 3. Predicted relevance guides selective decoding and encoding.

Table 3: Evidence-acquisition latency (second/question). Shared-answerer selectors use $K = 64$. Native-system timings retain their original protocols and instrumentation. MLVU BOLT uses $\alpha = 2.5$ for timing and $\alpha = 3$ for accuracy.

	Shared answerer		Native released system		
Benchmark	DREAMSEEK	Competing selector	TimeSearch-R	Video-o3	Video-R1
LVH	41.4	183.3 AKS (L5, $t = 0.8$)	28.6	21.9	566.4
Video-MME long	77.2	139.5 FOCUS	47.7	26.9	684.7
HLVB	65.3	180.0 BOLT ($\alpha = 3$)	26.8	22.2	681.4
MLVU	18.4	93.7 BOLT ($\alpha = 2.5$)	16.9	21.5	690.4
LVBench	66.8	205.8 FOCUS	65.2	30.8	677.2

Table 4: Fixed-count DREAMSEEK and DREAMSEEK$_{\text{ad}}$ at $K = 64$. Adaptive delivery uses the cumulative-score rule in Section 4.4 with $\eta = 0.15$. Frames are average realized delivery counts, with savings relative to nominal 64.

	QA accuracy (%)		Adaptive delivery	
Benchmark	DREAMSEEK	DREAMSEEK$_{\text{ad}}$	Frames	Frames saved
LVH	57.40 ± 0.38	56.20 ± 0.08	40.2	37.2%
Video-MME long	57.56 ± 0.71	55.34 ± 1.02	55.1	13.9%
HLVB	64.60 ± 1.10	64.49 ± 0.21	35.0	45.3%
MLVU	74.96 ± 0.36	72.15 ± 0.26	28.6	55.3%
LVBench	53.58 ± 0.46	52.47 ± 0.18	57.4	10.3%

Completion and correction add overhead, so read savings need not yield proportional time savings. These timings include decoding, encoding, and search but exclude final VLM answering. Against AKS on LVH and FOCUS on Video-MME long and LVBench, DREAMSEEK is also more accurate.

Compared with DREAMSEEK at $K = 64$, TimeSearch-R reports lower acquisition latency on all five benchmarks, and Video-o3 on all except MLVU. Their accuracy is lower except for TimeSearch-R on in-domain LVH. Video-R1 is slower and less accurate throughout. Native hardware paths and timing instrumentation differ.

Beyond acquisition, DREAMSEEK$_{\text{ad}}$ adapts the delivered frame count while leaving search unchanged. Average counts are 10.3–55.3% below nominal 64, with mean QA accuracy 0.11–2.81 percentage points lower than fixed-count DREAMSEEK (Table 4). Fewer frames may reduce answering latency, although this benefit is not quantified here. HLVB combines substantial frame savings with the smallest accuracy decrease, whereas MLVU has the largest saving and decrease.

5.3 SEARCH DYNAMICS

Acquisition effort across rounds. Beyond total acquisition cost, we examine how new frame reads are distributed across search rounds. Some longer searches show a burst–valley–burst (B–V–B) pattern: a large initial batch, few intervening reads, and another large batch (Table 5). Among searches reaching at least four rounds after initialization, this occurs in 93.2–99.4% of cases on four benchmarks and 51.8% on MLVU. Across all questions, including earlier stops, its frequency ranges from 1.3% to 35.1%. The pattern is consistent with the search rules. Initial verification can inspect many unchecked regions. Later completions may introduce fewer unchecked candidates, reducing new reads. When none remain, playoff probing can gather another batch within checked regions.

Among questions entering the first search round after initialization, replacing learned completion with nearest-observed features raises the median number of new reads from 370 to 555 on Video-MME long and from 505 to 800 on LVBench. Each unchecked peak triggers a local verification batch. Therefore, predicted features affect which peaks are proposed and how many frames are read

Table 5: Observed B–V–B allocation. R counts post-initialization rounds. B_1, V, and B_2 denote the initial burst, valley, and later burst. Read-count medians use trajectories exhibiting B–V–B.

Benchmark	Reach $R \geq 4$	B–V–B | $R \geq 4$	B–V–B | all	Median new reads B_1/V/B_2
LVH	146/921 (15.9%)	136/146 (93.2%)	14.8%	210/5/231
Video-MME long	318/900 (35.3%)	316/318 (99.4%)	35.1%	367.5/5/355
HLVB	57/342 (16.7%)	55/57 (96.5%)	16.1%	195/5/231
MLVU	56/2,174 (2.6%)	29/56 (51.8%)	1.3%	160/5/191
LVBench	254/1,223 (20.8%)	251/254 (98.8%)	20.5%	365/5/344

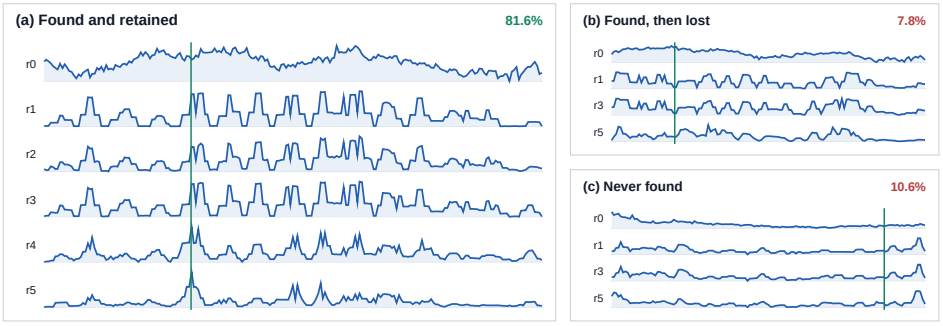

Figure 3: Three outcomes of iterative belief updating. Rows show initialization ($r0$) and subsequent rounds. Blue beliefs are independently min–max normalized per round. Green lines mark reference evidence-window positions reconstructed from stored final ranks. Percentages summarize a separate LVH analysis that excludes the illustrative runs.

under identical search rules. A model trained for temporal prediction can influence acquisition effort without being trained to choose search actions.

Evidence discovery and retention. A search can find relevant evidence and later lower its priority as new observations revise predicted features and relevance corrections. Persistent observational feedback does not guarantee a stable rank as proposals and competing regions change. On LVH, evidence is found and retained in 81.6% of questions, found then lost in 7.8%, and never found in 10.6%. Figure 3 illustrates these three outcomes.

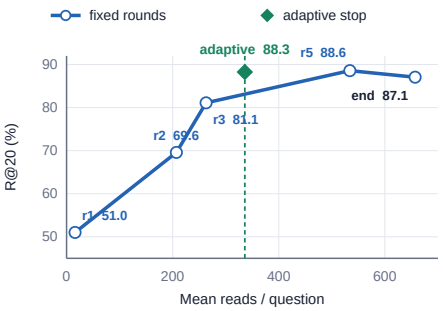

Figure 2: R@20 versus mean search reads on LVH ($n = 921$).

5.4 Ablating the DreamSeek design

Learned temporal completion. Learned completion improves tight evidence capture over nearest copying by 10.62 and 15.27 points at matched budgets of 32 and 64 search reads (Table 6). Nearest copying reuses observed features at unread locations, whereas learned completion uses temporal context to predict different features between anchors. This allows question relevance to vary beyond anchor similarities. For full-search QA on all four benchmarks, learned completion likewise outperforms nearest copying, which outperforms observed-only scoring.

Iterative search and stopping. On LVH, R@20 rises from 51.01% with the initial observations to 88.58% at round five (Figure 2). Adaptive stopping reaches 88.26% with approximately 37% fewer mean reads than the five-round setting. Extended search acquires more frames without improving localization. New observations add information but can also demote previously prioritized regions as the evidence ranking changes. The QA ablation likewise shows limited benefit from continued search. Relative to the evaluated no-stop configuration, adaptive stopping reduces mean search reads

Table 6: Representation ablation: tight capture (%) on the LVH-train analysis split (n = 1,124, excluded from TWM training) and at nominal $K = 64$. V-JEPA's setting is shown in Appendix B.

	Tight evidence capture		Full-search QA accuracy (%)			
Representation	32 reads	64 reads	Video-MME long	HLVB	MLVU	LVBench
Observed only	57.38	61.03	48.73 ± 0.21	53.82 ± 0.55	60.60 ± 0.31	42.74 ± 0.16
Nearest copy	65.57	68.68	50.39 ± 0.24	55.12 ± 0.62	63.27 ± 0.35	45.18 ± 0.18
Learned completion	**76.19**	**83.95**	$\mathbf{57.56 \pm 0.71}$	$\mathbf{64.60 \pm 1.10}$	$\mathbf{74.96 \pm 0.36}$	$\mathbf{53.58 \pm 0.46}$
V-JEPA-based configuration	57.94	60.70	47.56 ± 0.16	55.99 ± 1.45	62.93 ± 0.19	39.00 ± 0.23

Table 7: Stopping ablation at $K = 64$. Δ is no-stop minus adaptive accuracy (percentage points). Read savings are relative to the evaluated no-stop configuration.

	QA accuracy (%)			Mean search reads		
Benchmark	Adaptive stop	No stop	Δ	Adaptive stop	No stop	Reads saved
Video-MME long	57.56 ± 0.71	56.41 ± 0.32	-1.15	680.0	1,137.3	40.2%
HLVB	64.60 ± 1.10	63.89 ± 0.35	-0.71	276.0	633.9	56.5%
MLVU	74.96 ± 0.36	74.01 ± 0.49	-0.95	59.4	297.5	80.0%
LVBench	53.58 ± 0.46	53.64 ± 0.71	$+0.06$	748.9	1,727.8	56.7%

Table 8: Final-frame delivery ablation with identical search outputs and answerer at $K = 64$.

Benchmark	Several nearby frames per selected location	One frame from each top-scoring 8-s window	Top-scoring windows + one-third evenly spaced
Video-MME long	55.74 ± 0.32	54.67 ± 0.32	$\mathbf{57.56 \pm 0.71}$
HLVB	63.71 ± 0.22	63.45 ± 1.24	$\mathbf{64.60 \pm 1.10}$
MLVU	73.70 ± 0.49	72.15 ± 0.35	$\mathbf{74.96 \pm 0.36}$
LVBench	51.58 ± 0.14	47.55 ± 0.43	$\mathbf{53.58 \pm 0.46}$

by 40.2–80.0%, with higher mean QA accuracy on three benchmarks and only 0.06 points lower on LVBench (Table 7).

Final-frame delivery. Combining high-scoring windows with evenly spaced context gives the highest mean QA accuracy on all four benchmarks (Table 8). One frame per top-scoring window performs worse than several nearby frames per location, suggesting that removing neighbors can discard useful local detail. Yet reserving one third of the budget for evenly spaced context improves on both alternatives, consistent with complementary needs for local evidence and broader coverage.

Appendix C examines search-read boundary sensitivity, duration effects on stopping, and question-type differences. DREAMSEEK outperforms uniform sampling on most evaluated types but remains weaker on Video-MME long counting and LVBench summarization. Appendix D compares delivered evidence in successful and unsuccessful cases, highlighting complementary local and temporal context and difficulties with fine visual details.

6 CONCLUSION

DREAMSEEK makes unread video searchable through temporal prediction and persistent corrections from real observations. It improves evidence capture at matched read budgets without search-action supervision and helps existing answerers acquire informative evidence at lower latency than the timed shared-answerer baselines. This gives compact temporal world models a practical role in long-video QA without task-specific VLM fine-tuning.

Limitations. Estimating what evidence remains missing is an open direction for DREAMSEEK . Counting and summarization motivate this next step, since finding relevant moments may still leave repeated occurrences or broader developments uncovered. Learning to represent these gaps could help search decide when to revisit an event, expand temporal coverage, or stop.

AI USE STATEMENT

We used generative AI primarily to improve grammar, clarity, and organization, and to assist with literature checks and feedback on methodology and result interpretation. The authors reviewed the AI-assisted content for accuracy and take full responsibility for the manuscript.

ETHICS STATEMENT

DREAMSEEK uses existing video datasets and pretrained models and may inherit their biases and privacy concerns. Sparse evidence selection may omit relevant context and contribute to misleading answers. Responsible applications should respect data usage restrictions and individual privacy, prevent unauthorized surveillance, and account for these limitations when interpreting model outputs.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we describe our method in Section 4 and document the datasets, baselines, and evaluation protocol in Section 5. Appendices A and B provide additional training, implementation, and comparison settings, while Appendix C reports sensitivity and task-category analyses. We will also release our implementation as open-source software to support independent verification and further research.

REFERENCES

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Trans. Mach. Learn. Res.*, 2024, 2024.

Keshigeyan Chandrasegaran, Agrim Gupta, Lea M. Hadzic, Taran Kota, Jimming He, Cristóbal Eyzaguirre, Zane Durante, Manling Li, Jiajun Wu, and Li Fei-Fei. Hourvideo: 1-hour video-language understanding. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *Int. J. Comput. Vis.*, 130(1):33–55, 2022. doi: 10.1007/S11263-021-01531-2.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 24108–24118. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.02245.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Vincent Cartillier, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Devansh Kukreja, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,600 hours of egocentric video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(11):9468–9509, 2025. doi: 10.1109/TPAMI.2024.3381075.

Zefeng He, Xiaoye Qu, Yafu Li, Siyuan Huang, Daizong Liu, and Yu Cheng. Framethinker: Learning to think with long videos via multi-turn frame spotlighting. *CoRR*, abs/2509.24304, 2025. doi: 10.48550/ARXIV.2509.24304.

Shuming Liu, Chen Zhao, Tianqi Xu, and Bernard Ghanem. BOLT: boost large vision-language model without training for long-form video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 3318–3327. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00315.

Martin Q. Ma, Willis Guo, Aditya Agrawal, Ankit Gupta, Paul Pu Liang, Ruslan Salakhutdinov, and Louis-Philippe Morency. Video active perception: Effective inference-time long-form video understanding with vision-language models. *CoRR*, abs/2605.01662, 2026. doi: 10.48550/ARXIV.2605.01662.

Junwen Pan, Yuan Zhang, Rui Zhang, Xin Wan, Qizhe Zhang, Ming Lu, Shanghang Zhang, and Qi She. Zoomv: Temporal zoom-in for efficient long video understanding.

Junwen Pan, Qizhe Zhang, Rui Zhang, Ming Lu, Xin Wan, Yuan Zhang, Chang Liu, and Qi She. Timesearch-r: Adaptive temporal search for long-form video understanding via self-verification reinforcement learning. *CoRR*, abs/2511.05489, 2025. doi: 10.48550/ARXIV.2511.05489.

Jihao Qiu, Lingxi Xie, Xinyue Huo, Qi Tian, and Qixiang Ye. Longvideo-r1: Smart navigation for low-cost long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 40505–40515, June 2026.

Benjamin Schneider, Dongfu Jiang, Chao Du, Tianyu Pang, and Wenhu Chen. Quickvideo: Real-time long video understanding with system algorithm co-design. *Trans. Mach. Learn. Res.*, 2026, 2026.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. Longvu: Spatiotemporal adaptive compression for long video-language understanding. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

Chong Tang, Sannara Ek, Dirk Koch, Robert Mullins, Alex Weddell, and Jagmohan Chauhan. Surge: Surprise-guided token reduction for efficient video understanding with vlms. In *International Conference on Learning Representations*, volume 2026, pp. 3849–3868, 2026.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 29118–29128. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.02711.

Qwen Team. Qwen3-vl technical report. *CoRR*, abs/2511.21631, 2025. doi: 10.48550/ARXIV.2511.21631.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 98–106. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.18.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, and Jie Tang. Lvbench: An extreme long video understanding benchmark. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 22958–22967. IEEE, 2025a. doi: 10.1109/ICCV51701.2025.02131.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for LLM reasoning on long videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 3272–3283. Computer Vision Foundation / IEEE, 2025b. doi: 10.1109/CVPR52734.2025.00311.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Antoine Yang, Arsha Nagrani, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vidchapters-7m: Video chapters at scale. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Jinhui Ye, Zihan Wang, Haosen Sun, Keshigeyan Chandrasegaran, Zane Durante, Cristóbal 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 8579–8591. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00802.

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 *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Huaying Yuan, Zheng Liu, Junjie Zhou, Hongjin Qian, Yan Shu, Nicu Sebe, Ji-Rong Wen, and Zhicheng Dou. Videoexplorer: Think with videos for agentic long-video understanding. *arXiv preprint arXiv:2506.10821*, 2025.

Xiangyu Zeng, Zhiqiu Zhang, Yuhan Zhu, Xinhao Li, Zikang Wang, Changlian Ma, Qingyu Zhang, Zizheng Huang, Kun Ouyang, Tianxiang Jiang, Ziang Yan, Yi Wang, Hongjie Zhang, Yali Wang, and Limin Wang. Video-o3: Native interleaved clue seeking for long video multi-hop reasoning. *CoRR*, abs/2601.23224, 2026. doi: 10.48550/ARXIV.2601.23224.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 11941–11952. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01100.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 13691–13701. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.01278.

Zirui Zhu, Hailun Xu, Yang Luo, LIU YONG, Kanchan Sarkar, Zhenheng Yang, and Yang You. Focus: Efficient keyframe selection for long video understanding. In *International Conference on Learning Representations*, volume 2026, pp. 98463–98486, 2026.

A Training and Implementation Details

A.1 Temporal World Model Architecture

Figure 4 shows the temporal world model described in Section 4.1. Its temporal network is a bidirectional Transformer with eight pre-LayerNorm layers, hidden width 512, eight attention heads,

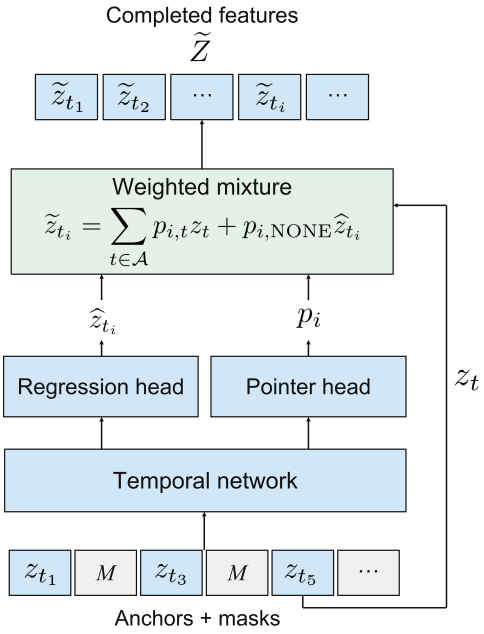

Figure 4: Temporal world model structure. M denotes a learned mask token. The outer path carries observed anchor features to the mixture, and NONE denotes the direct-prediction option. Token counts are schematic.

feed-forward width 2048, and no dropout. Each observed slot receives a linear projection of its 1,152-dimensional SigLIP feature, while each unobserved slot receives a learned mask token. Both receive an observation-status embedding and a sinusoidal embedding of normalized temporal position. A global token encodes $\log(T/1800)$, and attention includes a learned per-head bias that decreases with normalized temporal distance.

After a final LayerNorm, a linear regression head maps each 512-dimensional slot representation to the 1,152-dimensional direct prediction $\widehat{z}_{t_i}$. The pointer head forms a query at each temporal slot and keys at anchor slots through separate learned $512 \to 512$ linear projections. Each anchor logit is the dot product of the query and anchor key divided by $\sqrt{512}$. A separate $512 \to 1$ linear map of the query slot's representation produces the NONE logit. Applying a softmax jointly to the anchor and NONE logits yields $p_{i,t}$ and $p_{i,\mathrm{NONE}}$. These weights combine the original anchor features with the direct prediction according to Eq. equation 2.

A.2 Training Data and Objectives

Training uses features from Ego4D v2 (Grauman et al., 2025), EPIC-KITCHENS-100 (Damen et al., 2022), the Internet Archive Feature Films and Prelinger collections, VidChapters-7M (Yang et al., 2023), and MIT OpenCourseWare. All benchmark videos are excluded.

Episodes contain 512, 1,024, or 1,792 temporal segments, with the length sampled uniformly and respective batch sizes of 32, 16, or nine sequences. Anchor placement mixes scattered and clustered observations. With probability 0.67, approximately one quarter of the anchors are spread evenly with a random phase and the remainder form bursts of one to five segments. Otherwise, anchors cluster around a randomly chosen temporal focus, with a Gaussian spread of $T/16$ segments. Anchor density varies across episodes, exposing completion to different amounts of observed context. Feature pre-extraction applies to temporal-model training, while evaluation follows the from-pixels acquisition protocol.

The three objectives introduced in Section 4.1 are combined as

$$\mathcal{L} = \mathcal{L}_{\mathrm{reg}} + 0.5\,\mathcal{L}_{\mathrm{ptr}} + 2.0\,\mathcal{L}_{\mathrm{nce}}. \tag{7}$$

The regression loss is the mean squared error between the direct prediction $\widehat{z}_{t_i}$ and the true feature z_{t_i}, averaged over feature dimensions and hidden positions. The pointer loss is cross-entropy against the most similar observed anchor when its cosine similarity to the hidden target is at least 0.9, and against NONE otherwise. The contrastive term is InfoNCE with temperature 0.05, evaluated on the mixture outputs at up to 32 sampled hidden positions per sequence. Each positive is the corresponding true feature, and negatives are sampled true features from other sequences in the batch. Other positions in the same sequence are excluded from the negative set to avoid treating nearby, visually similar content as negative evidence. All supervision applies to hidden positions, and the targets are derived from video features without question-answer labels.

Table 9 reports the optimization settings.

A.3 Search and Delivery Implementation

Question independence applies to the completion function given its anchors. During search, question-dependent relevance scores influence which regions are inspected and which observations enter the next anchor set. Different questions can therefore lead to different completed timelines for the same video, although the temporal model receives no question input. Furthermore, the full observation set and the current anchor set serve different roles. The observation set retains acquired real features, while only the anchor subset supplies the context for a particular completion. Retaining an observation does not imply that its feature remains an input to every subsequent prediction.

Table 9 summarizes the principal training and inference settings. Search uses two-second temporal segments, and the text encoder receives the question without answer options or an added prefix. Predicted features are normalized for cosine-similarity scoring. Belief smoothing uses a moving average over two segments on either side, and candidate peaks are retained greedily by height with at least eight segments between them. Verification reads the peak and two neighboring segments on each side, while its correction applies over the wider region of eight segments on each side. For an observed regional cosine score s, the correction factor is

$$\gamma(s) = \mathrm{clip}\left(0.2 + 0.8\frac{\max(s, 0)}{0.0839},\ 0.2,\ 5.0\right). \tag{8}$$

The reference similarity 0.0839 gives a neutral factor of one. These factors update the persistent log-correction field according to Eq. equation 5. The normal search configuration uses a concentration threshold of 0.05 and a read boundary of 512. Both are checked between rounds, so a verification batch can take the cumulative read count beyond the boundary. Final delivery follows Section 4.4, using eight-second windows and one real frame at each selected window's midpoint.

B Evaluation and Analysis Details

B.1 Comparison Protocols

Table 2 separates shared-answerer comparisons from evaluations of complete released systems. Uniform sampling, BOLT, AKS, FOCUS, and DreamSeek use common questions, answering prompts, frame preprocessing, and a frozen answerer at each nominal delivery budget. Matching K controls the frame allocation to the answerer, while search reads and acquisition computation can differ. TimeSearch-R, Video-R1, and Video-o3 retain their native answerers and inference pipelines. TimeSearch-R's released checkpoint is in-domain on LVH. Published results marked $^{\|}$ use different evaluation splits and provide additional context beyond the shared-question comparisons.

B.2 Localization and Search Dynamics

The two localization analyses evaluate different aspects of evidence search. Tight evidence capture in Table 6 evaluates localization at fixed budgets of 32 and 64 search reads on the LVH-train analysis split (n = 1,124), which is excluded from temporal-model training. Figure 2 uses a separate LVH set (n = 921) and reports R@20, evidence recall within the top 20 ranked temporal windows. Its horizontal axis measures cumulative search reads averaged over questions. The capture analysis evaluates localization at fixed read budgets, whereas the R@20 curve follows the evidence ranking as search proceeds. Their absolute values refer to different metrics and evaluation populations.

Table 9: Principal training and inference settings. The parameter count covers the temporal world model. The encoders and answerer remain frozen. Search radii are measured in two-second segments. B is the between-round read boundary, and K is the nominal delivery budget.

Setting	Value
Training	
Temporal-model parameters	Approximately 26.9M
Visual and text encoders	SigLIP
Training inputs	Pre-extracted 1,152-dimensional visual features
Optimizer	AdamW
Learning rate / weight decay	3×10^{-4} / 0.01
Warm-up / schedule	600 steps, followed by cosine decay
Training budget	32,000 steps
Gradient clipping threshold	1.0
Precision / hardware	bfloat16 / one NVIDIA GH200 120GB GPU
Inference	
Shared answerer	Qwen3-VL-8B-Instruct
Initial observations	16 evenly spaced segment timestamps
Smoothing radius / peak separation	2 / 8 segments
Verification radius / correction radius	2 / 8 segments
Playoff eligibility radius	4 segments
Concentration threshold / normal read boundary	$\tau = 0.05$ / $B = 512$
Delivery-window duration	8 seconds
Context allocation	$\lfloor K/3 \rfloor$ evenly spaced windows
Adaptive cumulative-score retention	$\eta = 0.15$ at nominal $K = 64$

Discovery and retention describe changes in the priority of reference evidence over a search trajectory. *Found and retained* denotes evidence prioritized at the final search state, without requiring uninterrupted priority after its first discovery. *Found then lost* denotes evidence prioritized at an earlier state but no longer prioritized at the end. *Never found* denotes evidence that is never prioritized during the trajectory. A loss of priority does not imply deletion of an acquired observation, and these outcomes do not establish whether the final delivered frames contain sufficient evidence for a correct answer. The trajectories in Figure 3 illustrate these outcomes using a separate set of runs from the aggregate proportions. Their belief curves are normalized independently within each round, so curve heights cannot be used to compare absolute belief magnitude across rounds.

The burst–valley–burst analysis in Table 5 concerns new reads per acquisition round: an initial large batch, an intermediate phase with few new reads, and a later large batch. Here, R counts rounds after initialization. The conditional rate uses searches reaching $R \geq 4$ as its denominator, while the full-set rate uses all evaluated questions, including searches that stop earlier. Phase read-count medians are calculated only over trajectories exhibiting the pattern. The figures use separate display conventions: Figure 3 labels initialization as $r0$, while Figure 2 labels the same stage as $r1$. These labels distinguish display indexing from the post-initialization round count R.

B.3 Ablation Controls

The representation controls in Table 6 distinguish scoring from observed features alone, extending observed features to unread locations by nearest copying, and predicting unread features with learned completion. The fixed-read comparison controls acquisition effort. The full-search QA comparison instead uses a common answerer and nominal $K = 64$, allowing search-read counts to vary with the representation. Consequently, its accuracy differences include the effects of the resulting search trajectories.

The V-JEPA configuration uses V-JEPA 2 ViT-g features from 64 frames sampled within each two-second segment at 384-pixel input resolution. Its observation unit contains 64 sampled frames, unlike the single-frame observations used by the SigLIP-based controls. Token features are mean-pooled into a 1,408-dimensional segment representation, and a temporal world model with the same hidden architecture predicts in that feature space. A separate projector maps these fea-

Table 10: Search-read boundary sensitivity at nominal $K = 64$, with learned completion, answerer, and delivery rule fixed. Accuracy (%) is mean ± sample SD across seeds. Reads/q denotes mean search reads. Boundaries are checked between rounds, so a batch may exceed B, while adaptive stopping can terminate earlier.

	Normal adaptive search		$B = 256$		$B = 128$	
Benchmark	Accuracy	Reads/q	Accuracy	Reads/q	Accuracy	Reads/q
Video-MME long	57.56 ± 0.71	680.0	56.39 ± 0.32	252.3	56.00 ± 0.16	126.3
HLVB	64.60 ± 1.10	276.0	64.89 ± 0.83	180.9	63.58 ± 0.21	117.1
MLVU	74.96 ± 0.36	59.4	74.89 ± 0.39	38.8	73.10 ± 0.42	29.8
LVBench	53.58 ± 0.46	748.9	51.85 ± 0.81	250.1	49.13 ± 0.29	125.3

tures into the SigLIP text space using LayerNorm followed by three linear layers with widths $1408 \to 2048 \to 2048 \to 1152$, GELU activations, and dropout 0.1 between layers. The projector is trained on Ego4D narration and temporal-query pairs using a pairwise sigmoid loss, with other segments from the same video supplying temporal negatives. This alignment training is specific to the V-JEPA control. During search, eight completions are sampled from the pointer distribution and their sigmoid relevance scores are averaged. Real observations are scored through the same projector, with a separately calibrated neutral score. This configuration therefore changes the visual access unit, feature space, and relevance readout together. Its comparison does not isolate the temporal predictor alone.

The stopping comparison in Table 7 holds learned completion, the answerer, and the fixed delivery rule constant at nominal $K = 64$. Its read savings are measured relative to the evaluated no-stop configuration, using the difference in mean search reads divided by the no-stop mean. Accuracy differences are no-stop minus adaptive accuracy in percentage points. This comparison allows the search endpoint and resulting evidence to differ. In contrast, the delivery comparison in Table 8 applies different frame-delivery rules to identical search outputs, isolating how those outputs are converted into answering context.

B.4 Read Counts and Timing

A search read processes a selected real video observation through E_{img}. Final delivery can decode timestamps outside the search observation set, so search-read counts and delivered-frame counts measure different stages. The latency measurements in Table 3 start from pixels and include decoding, feature construction, and search computation, excluding final VLM answering. Each reported selector comparison uses the configuration listed for that benchmark. On MLVU, BOLT uses $\alpha = 2.5$ for timing and $\alpha = 3$ for accuracy, so these entries describe different operating configurations. Native-system timings retain their respective hardware paths and instrumentation. The acquisition speedups and reductions in delivered frames therefore do not establish proportional reductions in total QA latency.

C Sensitivity Analyses and Boundary Conditions

Reducing the search-read boundary from its normal setting of $B = 512$ produces different accuracy–cost tradeoffs across benchmarks (Table 10). At $B = 256$, HLVB and MLVU retain similar mean accuracy to normal adaptive search while using approximately one third fewer reads. Video-MME long and LVBench show larger read reductions accompanied by lower mean accuracy. Tightening the boundary to $B = 128$ lowers mean accuracy on all four benchmarks relative to normal adaptive search, with the largest decrease of 4.45 points on LVBench. This sensitivity complements the stopping ablation in Table 7: adaptive stopping can avoid unnecessary continuation, whereas a tighter fixed read boundary can also curtail useful acquisition. MLVU illustrates why mean read counts alone cannot determine a suitable boundary. Normal search averages 59.4 reads per question, yet $B = 128$ reduces mean accuracy by 1.86 points. A boundary above the mean can still affect questions requiring more observations, so the average does not capture how acquisition effort is distributed across questions.

Table 11: Descriptive accuracy differences in percentage points between DREAMSEEK and uniform sampling at nominal $K = 64$ with a shared answerer. Positive Δ favors DREAMSEEK . Question-type labels follow each benchmark.

Video-MME long type	Δ	LVBench type	Δ
Attribute	$+8.64$	Entity	$+5.70$
Temporal reasoning	$+4.95$	Event	$+2.94$
Action recognition	$+3.70$	Key information	$+2.66$
Counting	-10.76	Reasoning	$+2.69$
		Temporal	$+3.90$
		Summarization	-2.83

The concentration criterion in Eq. equation 6 also has an intrinsic dependence on timeline length. For a flat positive belief $b_i = b > 0$ over T segments, it reduces to

$$\kappa = \frac{|\mathcal{S}_{c_\star}|b}{Tb} = \frac{|\mathcal{S}_{c_\star}|}{T}.$$

For a selected region of fixed size, this fraction increases as the number of timeline segments decreases. The configured region contains 17 segments when it is not clipped by a timeline boundary, giving $\kappa = 17/T$ under a flat belief. With fixed temporal spacing, shorter videos have a larger concentration score even though the belief provides no stronger localization. Thus, a fixed threshold can be reached whenever $|\mathcal{S}_{c_\star}|/T \geq \tau$, including under a flat belief. This calculation identifies a length-dependent baseline for the stopping score. It does not establish how the effect changes QA accuracy across video durations.

The question-type results reveal a further boundary in the usefulness of relevance-guided acquisition (Table 11). DREAMSEEK improves over uniform sampling on attribute, temporal-reasoning, and action-recognition questions in Video-MME long and on five of the six reported LVBench categories. However, it trails uniform sampling by 10.76 points on Video-MME long counting and 2.83 points on LVBench summarization. These deficits are consistent with a gap between finding relevant moments and acquiring complete evidence for aggregation. Counting may require coverage of distinct occurrences, while summarization may require developments distributed across the timeline. The gains from evenly spaced context in Table 8 provide related support for the value of broader context. However, neither comparison isolates incomplete coverage as the cause of the category deficits. They motivate acquisition and stopping criteria that account for the evidence required across moments as well as relevance at individual locations.

D QUALITATIVE EXAMPLES

We present successful and unsuccessful cases comparing the evidence delivered by DREAMSEEK , Uniform, BOLT, AKS, and FOCUS with a shared answerer at nominal $K = 64$. Each figure shows annotated reference frames above six selected frames per method. Display positions are allocated evenly across annotated intervals where possible, filled by distinct delivered frames selected by temporal proximity, and ordered chronologically. Columns need not share timestamps across methods. These subsets illustrate final evidence content, while answer outcomes use the complete delivered context and the available option-shuffle runs: two for DREAMSEEK and three for each baseline. The LVH panels use source-resolution exports at recorded delivery timestamps, whereas answering uses 256p video.

The successful cases in Figures 5–8 illustrate the value of combining useful local views with temporal context that spans the question's requirements. DREAMSEEK retains the bookshelf, outdoor, and car scenes in the queried order, views of the mixer's appearance, and handling context for the mask's location before dropping. The mask question requires recognizing that it was in the person's hands, so the relation between an object and the preceding action matters alongside its visibility. In the first-person case, the opening frame comes from the evenly spaced coverage allocation and provides temporal context alongside subsequent targeted views. The baseline comparisons reveal how incomplete coverage or less direct views can limit the evidence available for answering. BOLT and FOCUS omit the intermediate outdoor scene required by the sequence question, while FOCUS's

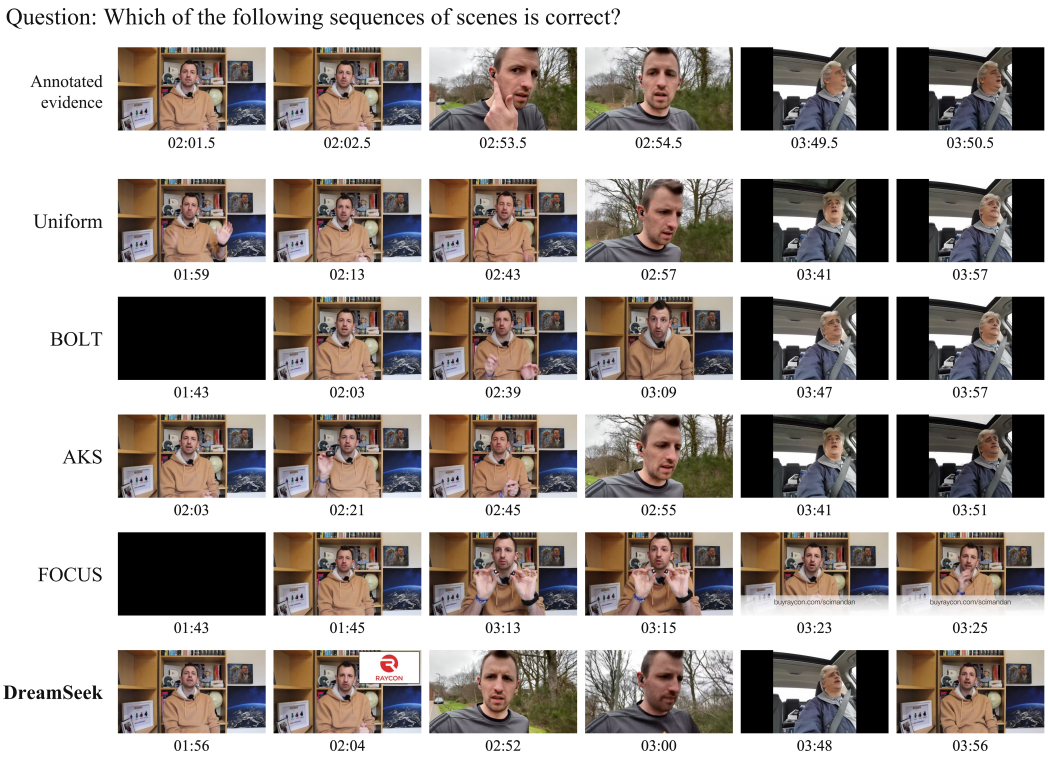

Figure 5: Evidence for a sequence spanning three scenes in HLVB. The reference row shows the bookshelf, outdoor, and car scenes. Method rows contain six selected frames from each final delivery set.

first-person delivery begins later in the opening sequence. AKS supplies less direct views of the mixer and answers its appearance question incorrectly, despite retaining other views of the activity. Several baselines also provide effective evidence: AKS and Uniform preserve the correct scene sequence, BOLT and Uniform capture the mixer's appearance, and FOCUS succeeds on the mask question. Across these selected examples, DREAMSEEK 's combined delivery provides complementary evidence for appearance, preceding state, and temporal order.

The unsuccessful cases in Figures 9 and 10 suggest specific opportunities to refine the delivery of fine visual details. In both cases, DREAMSEEK 's closest delivered frame is only two seconds from an annotated interval, closer than the baseline frames, but every method answers incorrectly in all recorded runs. The relevant activity is represented in the delivered context, while the questions require identifying a small manipulated object or resolving its position relative to another object. A plausible contributing factor is the temporal granularity of final delivery: one frame per eight-second window can bracket a brief interaction, as the frames at 02:52 and 03:00 do around the annotated tile-handling interval at 02:54–02:58. Small objects and clutter may also make the required distinctions difficult to resolve at the 256p answering resolution. Red components remain visible in other delivered frames, so a temporal gap alone does not establish the cause of the object-identification error. These observations motivate denser final sampling within selected regions and higher-resolution views of small objects as concrete refinements to investigate. Such changes could preserve the benefits of selective acquisition while giving the answerer clearer evidence of short interactions and spatial relations. Their potential to correct these errors remains to be evaluated.

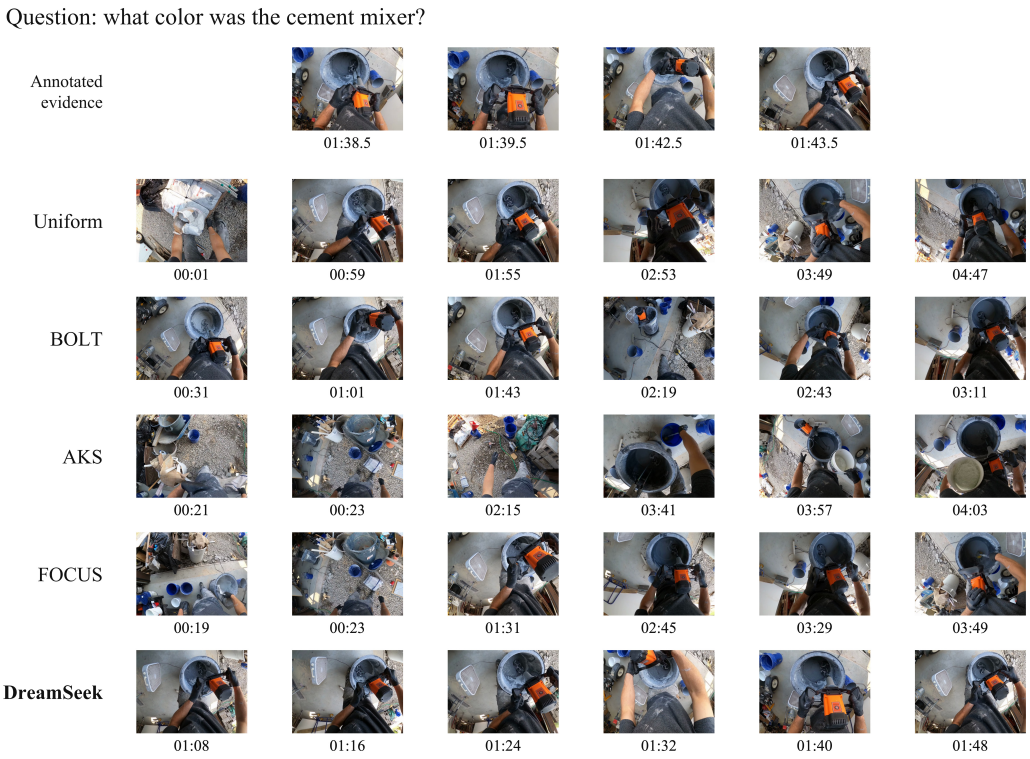

Figure 6: Object-appearance evidence in LVH. The question concerns the handheld mixer's orange, black, and gray appearance. The annotated references and delivered frames show different views of the tool.

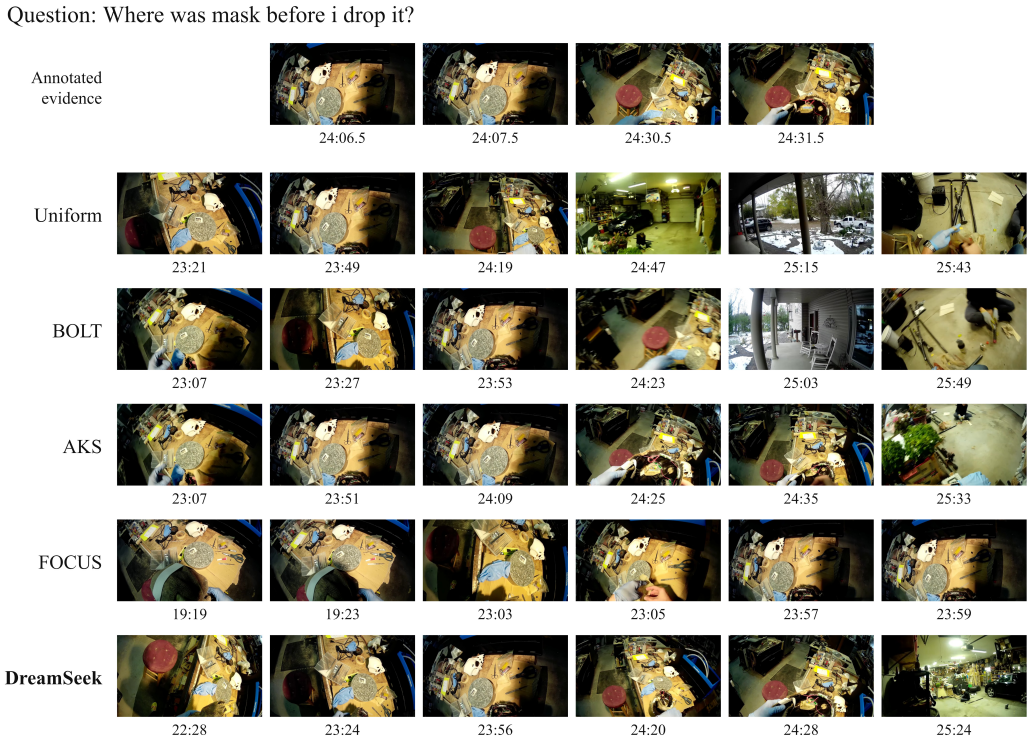

Figure 7: Evidence for the mask's location before a dropping action in LVH. The comparison shows the reference moments and surrounding delivered views of the workshop and object handling.

Question: Who is the first person to appear in the video?

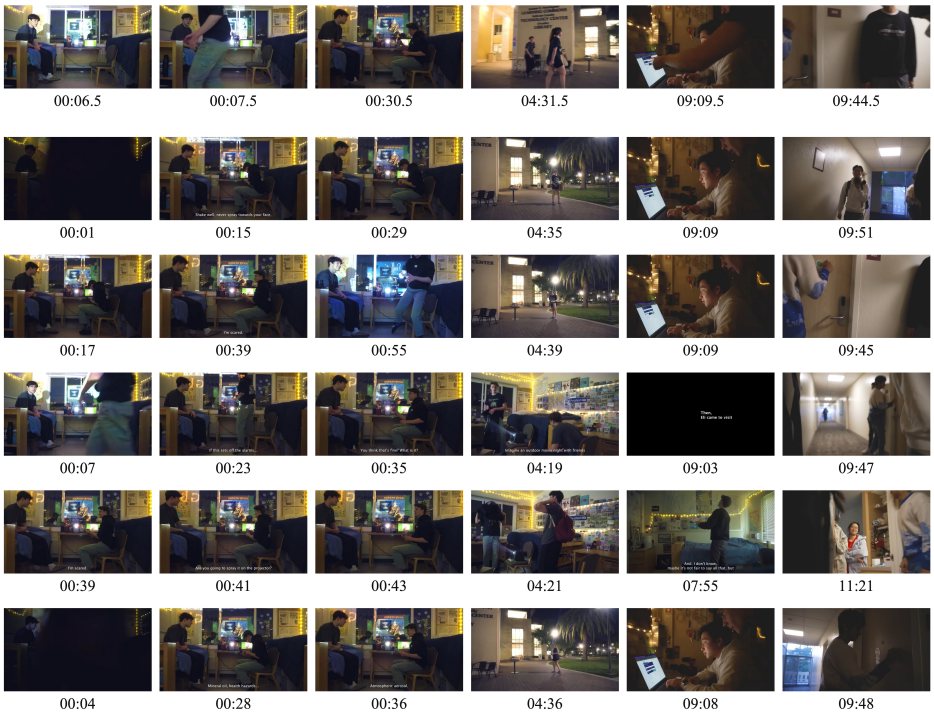

Figure 8: Evidence for identifying the first person to appear in an HLVB video. The opening frames provide temporal context for the identity question. Later reference images depict additional appearances.



Figure 9: An unsuccessful object-identification example in LVH. The annotated interaction involves a small red tile-leveling component, while the closest DREAMSEEK frames bracket the reference interval.

Question: where is the scissors before i pick it?



Figure 10: An unsuccessful spatial-relation example in LVH. The correct answer places the scissors to the left of the blue-and-black storage box before they are picked up. The cluttered workshop views require distinguishing this relation from the alternatives.