SETGAIN: SET-CONDITIONED ADAPTIVE EVIDENCE SELECTION FOR LONG VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-form video question answering requires selecting a compact yet sufficient evidence set from thousands of frames. Existing methods follow predefined search procedures or optimize selection through trajectory-level outcomes, without modeling how much new information a candidate contributes beyond the evidence already collected. We propose **SetGain**, a set-conditioned evidence-selection framework that jointly learns which frame to select next and when to stop. A gain model conditioned on the candidate pool and the evolving selected set estimates each remaining frame's marginal contribution, trained by supervised initialization with marginal temporal-coverage targets followed by reinforcement learning with answer-quality and evidence-coverage rewards. On Haystack-LVBench, SetGain improves temporal F1 by 13.1% relative over a pointwise top-k selector using the same candidate pool and evidence scores, with 16% fewer frames; end to end it reaches 2.23× the temporal F1 and 8.2 points higher QA accuracy (15.7% relative) than the strongest comparable prior method, TimeSearch-R. It further improves average QA accuracy by 2.9 points (4.7% relative) over TimeSearch-R on VideoMME, MLVU, and LongVideoBench, and reduces end-to-end latency by 53.5% (2.15× speedup) relative to T*, the fastest prior search method. These results demonstrate that explicitly modeling marginal evidence gain improves temporal localization, downstream answering, and inference efficiency. Code is available at `https://anonymous.4open.science/r/SetGain`.

1 INTRODUCTION

Long video question answering hinges on evidence selection. A video may contain tens of thousands of frames, while the evidence needed to answer a question is sparse and scattered across distant segments (Fu et al., 2025; Wu et al., 2024; Zhou et al., 2025; Ye et al., 2025). The common approach ranks frames independently by embedding similarity to the question (Radford et al., 2021; Zhai et al., 2023; Liang et al., 2024) and keeps the top-k. As Figure 1 shows, this fails through redundancy. Frames that repeat the same relevant information can all receive high scores, wasting the frame budget on information the set already contains. This occurs because pointwise scoring evaluates each frame's relevance without considering what it adds to the frames already selected. The goal is therefore to select a compact yet sufficient evidence set needed for a correct answer.

Meeting this goal requires a selector that jointly accounts for evidence compactness and sufficiency according to each question context, but existing approaches fall short on at least one of these requirements. Diversity-based methods (Carbonell & Goldstein, 1998; Kulesza & Taskar, 2012; Sun et al., 2025) use visual difference as a proxy for complementary evidence, but may discard necessary frames because they look similar and retain irrelevant ones because they look different. Agentic search methods such as VideoAgent (Wang et al., 2024) and T* (Ye et al., 2025) target missing evidence, but they depend on pretrained VLM judgments and predefined search rules, and their retrieval prioritizes relevance without explicitly assessing what each new frame adds to the selected set.

Recent reinforcement learning (RL) methods learn frame selection from task feedback, but their supervision does not jointly account for evidence compactness and sufficiency. Methods such as TimeSearch-R (Pan et al., 2026a) rely on question answering (QA) rewards, which are unreliable indicators of sufficiency because language priors can yield correct answers from incomplete evidence (Niu et al., 2021). Temporal annotations add grounding signal, but selecting frames from

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

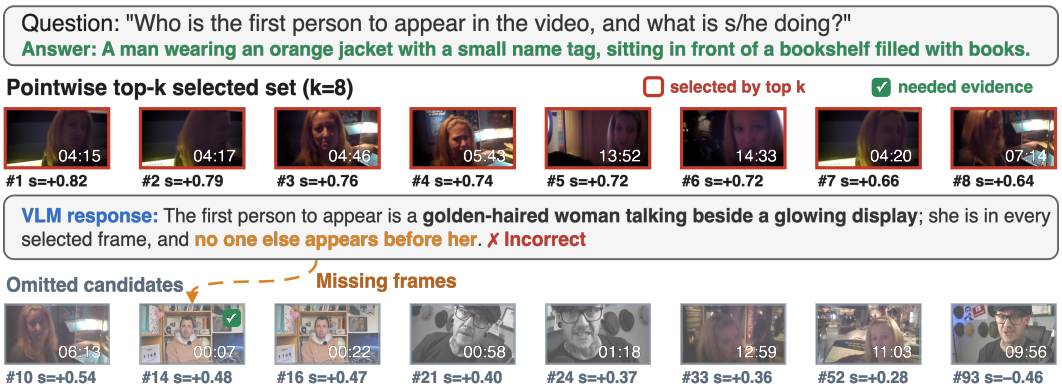


Figure 1: **Redundancy crowds out evidence in pointwise top-k retrieval.** Frames are scored independently, so the top-8 (red) repeat the same subject: most of the budget adds no new evidence.

the correct segment can still repeat the same information while missing details needed for the answer. Furthermore, the number of frames is rarely tied to what the question requires: diversity-based methods and T* use a preset budget, and the methods that do vary it decide by querying the model itself, VideoAgent on its own confidence and TimeSearch-R by re-answering the question, rather than on whether the collected evidence is complete. These limitations motivate supervision that jointly promotes compact and sufficient evidence according to each question's needs.

To address this, we introduce **SetGain**, built around a single quantity: the *marginal evidence gain* of a frame, i.e., how much evidence it adds beyond what the selected set already covers. This quantity governs both decisions in sequential selection: which frame to add next, and when to stop, which is once no remaining frame offers meaningful gain. At inference, **Set-Conditioned Adaptive Evidence Selection** uses a lightweight set-conditioned selector to estimate this gain for every remaining frame in a small candidate pool proposed by a frozen vision–language encoder, adds the highest, and re-estimates after each addition, so frames that repeat collected evidence are suppressed; a stopping head ends the search when the gain is exhausted, letting the number of frames adapt to each question. The resulting sequence of additions forms a trajectory. Suppressing repeats and stopping once the gain is exhausted keep the selected set precise, spending frames only on new evidence. Since frame-level evidence scores are computed once by a frozen cross-encoder (Nogueira & Cho, 2019; Li et al., 2026) outside the loop, each step costs a single lightweight forward pass rather than a Video-LLM call.

Because the selector is defined by the quantity it estimates, training amounts to teaching it that quantity, yet neither available signal expresses it directly. Temporal annotations grade each frame in isolation and cannot distinguish a redundant addition from a new one; answer rewards reflect what the answering model needs but judge only the final trajectory and may reward a lucky guess. **Marginal Gain Learning (MGL)** bridges the two. It first converts the annotations into set-dependent marginal coverage targets, together with a stop target that fires once no remaining frame adds coverage, teaching the selector the gain directly. **Coverage-Guided Policy Optimization (CGPO)** then refines complete trajectories through group-relative comparisons (Shao et al., 2024; Guo et al., 2025a), rewarding both evidence coverage and answer quality, with a confidence margin that gives lucky guesses little credit. By retaining the stop supervision, CGPO ensures that the learned gain reflects what the answerer actually needs and that the evidence budget is set by when the evidence is complete.

Experimental results show that SetGain consistently yields higher-quality evidence. On Haystack-LVBench (Ye et al., 2025), it improves temporal F1 by 13.1% relative over a pointwise top-k selector using the same candidate pool and evidence scores, with 16% fewer frames. End to end, SetGain reaches 2.23× the temporal F1 of TimeSearch-R (Pan et al., 2026a) and 8.2 points higher QA accuracy (15.7% relative). Across long video benchmarks (Fu et al., 2025; Zhou et al., 2025; Wu et al., 2024), it improves average QA accuracy over TimeSearch-R by 4.7% relative with 9.0% fewer frames, and completes the full search-and-answer pipeline 2.15× faster than T* (Ye et al., 2025), the fastest prior search method.

Our contributions are threefold:

1. We propose **SetGain**, which selects evidence by each frame's marginal gain given the frames already chosen: its **Set-Conditioned Adaptive Evidence Selection** adds the highest-gain frame and stops once the gain is exhausted, so the number of frames adapts to each question.
2. We develop **Marginal Gain Learning (MGL)**, which teaches this gain from set-dependent coverage and stop targets, then refines it with **Coverage-Guided Policy Optimization (CGPO)** so that it reflects what the answerer needs.
3. We conduct extensive experiments. On Haystack-LVBench and three long-video QA benchmarks, SetGain yields more precise evidence sets than pointwise selection at comparable recall with fewer frames, improves downstream QA accuracy, and runs $2.15\times$ faster end to end than the fastest prior search method.

2 METHOD

SetGain combines a candidate proposer (Sec. 2.2) with set-conditioned adaptive evidence selection (Sec. 2.3), trained by Marginal Gain Learning (Sec. 2.4); Figure 2 shows the complete pipeline. The proposer, evidence scorer, and answerer remain frozen throughout; only the gain predictor and stopping head are optimized.

2.1 PROBLEM FORMULATION

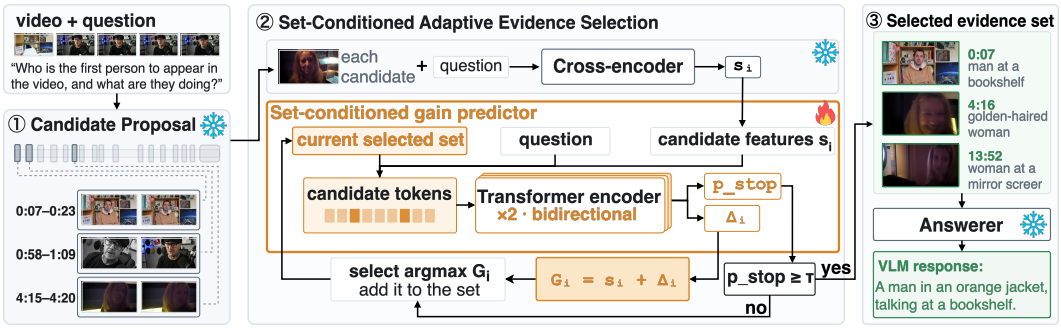


Figure 2: **The SetGain selection pipeline.** The candidate proposer reduces the video to a question-relevant pool, a frozen cross-encoder scores each candidate (s_i), and the set-conditioned gain predictor re-estimates $G_i = s_i + \Delta_i$ after every selection, stopping once the remaining candidates add little. Training is shown in Figure 3; only the gain predictor and stopping head are trained.

Given a video V and a question q, our goal is to select a compact set of frames containing sufficient evidence to answer q. Let $\mathcal{X} = \{x_i = (I_i, t_i)\}_{i=1}^{N}$ denote a temporally ordered candidate pool, where I_i is the image of candidate x_i and t_i its timestamp in seconds. Starting from $\mathcal{S}_0 = \varnothing$, SetGain constructs the evidence set sequentially for $r = 1, 2, \ldots$:

$$x_r^\star = \arg\max_{x_i \in \mathcal{X} \setminus \mathcal{S}_{r-1}} G_i^{(r)}, \qquad \mathcal{S}_r = \mathcal{S}_{r-1} \cup \{x_r^\star\}, \tag{1}$$

until a learned stopping criterion is met, yielding the final evidence set $\mathcal{S}$. Rather than scoring each frame once in isolation, SetGain assigns each remaining candidate a state-dependent gain

$$G_i^{(r)} = G(x_i \mid q, \mathcal{X}, \mathcal{S}_{r-1}), \tag{2}$$

which estimates the additional evidence contributed by x_i given the frames already selected.

2.2 CANDIDATE PROPOSAL

Processing every frame of a long video is expensive, and a question may require evidence from several distinct moments, so a single question-level ranking tends to concentrate the budget on one of

them. Given q, a frozen vision–language model therefore decomposes it into events $\mathcal{Q} = \{q_k\}_{k=1}^{n}$, each describing one observation required to answer q; the number of events n is decided by the model rather than fixed in advance. A frozen vision–language encoder scores every sampled frame v_j at timestamp t_j against each event, $\phi_j^{(k)} = \mathrm{sim}(v_j, q_k)$, and each frame is summarized by its best-matching event,

$$\phi_j = \max_{1 \le k \le n} \phi_j^{(k)}, \qquad \mathcal{P} = \{(v_j, t_j) \mid j \in \mathrm{Top}_B(\phi)\}, \tag{3}$$

where $\mathrm{Top}_B(\phi)$ returns the B highest-scoring frames of the video. A frame is therefore retained on the strength of its best-matching event rather than its similarity to the question as a whole, so evidence for any single required observation can enter the pool. Ordering $\mathcal{P}$ by timestamp gives the candidate pool $\mathcal{X} = \{x_i = (I_i, t_i)\}_{i=1}^{N}$ with $N = B$. The pool has the same size for every question regardless of video length or of how many events the decomposition returns, which bounds evidence-scoring cost and makes the selection budget the only quantity that varies across questions.

2.3 Set-Conditioned Adaptive Evidence Selection

The selector approximates the marginal gain and stopping criterion of Sec. 2.1 from frozen features at inference. Given the candidate pool $\mathcal{X}$, a frozen cross-encoder (Nogueira & Cho, 2019) jointly processes the question q and each candidate frame I_i to produce an evidence score $s_i = f_{\mathrm{ev}}(q, I_i)$, a yes–no readout of whether I_i contains evidence relevant to the question. Each candidate is scored independently; rather than selecting the top-scoring frames directly, SetGain uses s_i as the initial evidence estimate for set-conditioned selection.

Set-conditioned gain predictor. Let $\bar{\mathbf{e}}_i$ and $\bar{\mathbf{e}}_q$ denote the normalized frame and question embeddings from the proposal encoder (Sec. 2.2). At selection step r, we describe each candidate x_i by the candidate features

$$\mathbf{u}_i^{(r)} = \left[s_i,\, \tilde{t}_i,\, \psi_i,\, m_i^{(r)},\, \rho_i^{(r)} \right], \tag{4}$$

where s_i is its evidence score, $\tilde{t}_i = t_i/T$ its timestamp normalized by the video duration T, and $\psi_i = \langle \bar{\mathbf{e}}_i, \bar{\mathbf{e}}_q \rangle$ its question similarity, computed with the same encoder used for candidate proposal. These three features are fixed, whereas the remaining two depend on the selected set: $m_i^{(r)} = \mathbb{1}[x_i \in \mathcal{S}_{r-1}]$ indicates whether x_i has already been selected, and $\rho_i^{(r)} = \max_{x_\ell \in \mathcal{S}_{r-1}} \langle \bar{\mathbf{e}}_i, \bar{\mathbf{e}}_\ell \rangle$ measures its similarity to the current selected set, with $\rho_i^{(r)} = 0$ when $\mathcal{S}_{r-1} = \varnothing$. The term $\rho_i^{(r)}$ serves as a cheap inductive bias for reasoning about the candidate relative to previously selected evidence.

Each candidate is represented by one token formed from $[\bar{\mathbf{e}}_i; \mathbf{u}_i^{(r)}]$, and an additional question token formed from $\bar{\mathbf{e}}_q$ provides question context. A two-layer bidirectional Transformer h_θ (Vaswani et al., 2017) jointly processes all tokens, so each prediction depends on the full candidate pool and the evolving selected set. The gain head predicts a residual correction to the independent evidence score:

$$G_i^{(r)} = s_i + \Delta_i^{(r)}, \qquad \Delta_i^{(r)} = h_\theta(q, \mathcal{X}, \mathcal{S}_{r-1})_i. \tag{5}$$

The residual output is initialized to zero, so the initial policy follows the cross-encoder ranking and then learns to suppress candidates whose information the selected frames already carry, leaving those that contribute evidence not yet covered to rise.

Adaptive selection and stopping. The Transformer outputs for the candidate tokens are also pooled into a stopping head that predicts a stopping probability $p_{\mathrm{stop}}^{(r)}$ at state $\mathcal{S}_{r-1}$. Selection terminates when $p_{\mathrm{stop}}^{(r)} \ge \tau$; otherwise, SetGain adds the unselected candidate with the highest $G_i^{(r)}$ to the selected set (Eq. 1) and updates the candidate states. While s_i is computed once, the gain predictor and stopping head are reevaluated as the selected set evolves. The selected frames are finally reordered chronologically and passed to the answerer.

2.4 Marginal Gain Learning

Marginal Gain Learning (MGL) trains the selector in two stages. **Marginal coverage regression** (MCR) regresses the selector toward annotation-derived targets for the marginal gain and stopping

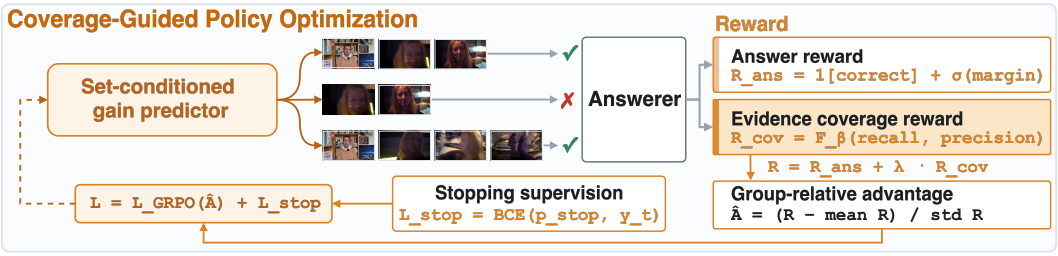


Figure 3: **Coverage-guided policy optimization.** Gumbel perturbations of the initialized policy produce diverse trajectories, which the frozen answerer scores. Group-relative advantages $\hat{A}$ update the selection policy, while stopping supervision (Eq. 8) prevents the stopping head from collapsing. Reward weights and the KL term are omitted.

criterion of Sec. 2.1 directly; **Coverage-Guided Policy Optimization** (CGPO) then aligns them with what the answerer actually needs by optimizing complete selection trajectories using answer quality and evidence coverage.

Marginal coverage regression. Let $\mathcal{T}^\star = \{g_j\}_{j=1}^{M}$ denote the annotated evidence segments, and let D_{ji} measure the compatibility between segment g_j and candidate x_i. The targets below use temporal compatibility, which decays with the distance from x_i to g_j; the coverage reward additionally incorporates appearance agreement. Given $\mathcal{S}_{r-1}$, define its current coverage of g_j as $c_j^{(r)} = \max_{x_\ell \in \mathcal{S}_{r-1}} D_{j\ell}$. The gain and stopping targets are

$$\widehat{G}_i^{(r)} = \frac{1}{M}\sum_{j=1}^{M}[D_{ji} - c_j^{(r)}]_+, \qquad y_{\text{stop}}^{(r)} = \mathbb{1}\left[\max_{x_i \in \mathcal{X}\setminus\mathcal{S}_{r-1}} \widehat{G}_i^{(r)} < \epsilon\right]. \tag{6}$$

Here, $\widehat{G}_i^{(r)}$ measures the additional coverage contributed by x_i, while $y_{\text{stop}}^{(r)}$ indicates that no remaining candidate provides meaningful evidence. We initialize the policy by regressing the predicted gains $G_i^{(r)}$ toward $\widehat{G}_i^{(r)}$ and training the stopping head with binary cross-entropy. Details of D_{ji} are given in Appendix A.

Coverage-Guided Policy Optimization. As illustrated in Figure 3, the initialized policy generates K trajectories per question by sampling selections with independent Gumbel perturbations of $G_i^{(r)}$ and sampling the stop action from $p_{\text{stop}}^{(r)}$ at each step. The selected frames are reordered chronologically and evaluated by the frozen answerer. Each trajectory A_k receives

$$R(A_k) = R_{\text{ans}} + \lambda R_{\text{cov}}, \qquad R_{\text{ans}} = w_{\text{acc}}\,\mathbb{1}[\hat{a} = a^\star] + w_{\text{mar}}\,\sigma\left(\frac{\delta - b_q}{\gamma}\right). \tag{7}$$

The answer reward R_{ans} combines a correctness term, which provides the primary task signal, with a margin term that provides denser feedback (Kwiatkowski et al., 2026; Wang et al., 2026), where δ is the log-probability gap between the correct and strongest incorrect answers, b_q is a question-specific baseline, and γ is a temperature (Appendix A.1). The evidence coverage reward R_{cov} is a soft F_β score (Rijsbergen, 1979) computed from D: soft recall encourages complete coverage of the annotated evidence, while soft precision penalizes unrelated selections, with β controlling the balance between completeness and selectivity. For each question, the group-relative advantage of trajectory A_k is $\hat{A}_k = (R(A_k) - \text{mean}_k\, R)/\,\text{std}_k\, R$.

Stopping supervision. Small differences in the relative rewards of sampled trajectories can initially favor either stopping or continuing. Policy-gradient updates then increase the probability of the favored decision, so it is sampled more often and the drift compounds until $p_{\text{stop}}^{(r)}$ approaches zero or one. Once the sigmoid saturates, little gradient remains for recovery, and the policy may collapse to a constant frame budget. We therefore retain direct stopping supervision using a separate H-step greedy rollout of the current gain predictor. The rollout does not execute the stop action; at

each state $\mathcal{S}_u$, it applies the stop target from Eq. 6:

$$\mathcal{L}_{\text{stop}} = \frac{1}{H} \sum_{u=0}^{H-1} \text{BCE}\Big(p_{\text{stop}}^{(u)},\, y_{\text{stop}}^{(u)}\Big)\,. \tag{8}$$

The loss is averaged across the batch and provides a direct gradient at every greedy selection state, independently of the stop actions sampled during policy optimization.

Finally, GRPO (Shao et al., 2024; Guo et al., 2025a) optimizes the selection policy using the advantages $\hat{A}_k$. The complete objective is

$$\mathcal{L} = \mathcal{L}_{\text{GRPO}}(\hat{A}) + \beta_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{stop}}\mathcal{L}_{\text{stop}}, \tag{9}$$

where the KL term regularizes the policy toward the reference policy.

3 Experiments

Datasets. We evaluate temporal search together with downstream question answering on Haystack-LVBench and Haystack-Ego4D (Ye et al., 2025), and additionally report question answering on VideoMME (Fu et al., 2025), MLVU (Zhou et al., 2025), and LongVideoBench (Wu et al., 2024). For training, we use Seeker-173K (Zeng et al., 2026), which supplies the annotated temporal evidence segments used as marginal-coverage targets. Seeker-173K is disjoint from all evaluation benchmarks; no LV-HAYSTACK question, video, or temporal annotation is used for training.

Evaluation Metrics. Following LV-HAYSTACK (Ye et al., 2025), we report temporal and visual precision, recall, and F1 using its standard five-second temporal matching and SSIM-based visual matching protocols. Precision, recall, and F1 are computed per question and then averaged over questions. We additionally report QA accuracy, average selected frames, theoretical TFLOPs, and wall-clock latency per question.

Baselines. We compare SetGain with four baseline groups: uniform sampling with 8 or 32 frames; workflow-based search methods, including VideoAgent (Wang et al., 2024), retrieval-based search, and T* (Ye et al., 2025); the RL-trained video reasoner Video-R1 (Feng et al., 2025) and the learned search policy TimeSearch-R (Pan et al., 2026a); and controlled training-free selectors based on pointwise top-k, MMR (Carbonell & Goldstein, 1998), and DPP (Kulesza & Taskar, 2012). The controlled baselines use the same candidate pool, evidence scores, and answerer as SetGain, differing only in their selection rule. For TimeSearch-R, we use its released checkpoint only for frame search; answers are produced by the same frozen Qwen2.5-VL-7B answerer used for SetGain.

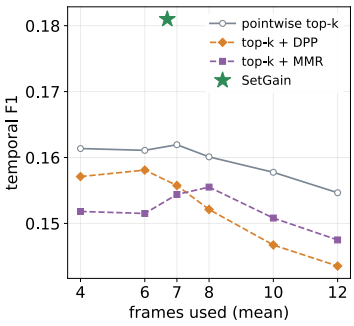
Implementation Details. The gain predictor uses a two-layer bidirectional Transformer with width 256 and four attention heads, and together with the stopping head totals 2.17M trainable parameters. We use Qwen2.5-VL-7B for both question decomposition and answering, SigLIP-400M as the proposal encoder with a candidate budget of $B = 128$ frames, and Qwen3-VL-Reranker-2B (Li et al., 2026) as the cross-encoder evidence scorer; all remain frozen. Both training stages run for a fixed 1,000 steps: no run is early-stopped, no intermediate checkpoint is selected, and no evaluation data is used for validation or model selection. During coverage-guided policy optimization, we sample $K = 8$ trajectories per question, anneal the sampling temperature from 1.5 to 0.8, and optimize with AdamW at a learning rate of 3×10^{-4}. We set $\beta_{\text{KL}} = 0.05$ and $\lambda_{\text{stop}} = 1.0$, and use $\beta = 1.5$ for the evidence coverage reward; Appendix A.1 lists the remaining hyperparameters. At inference, the stopping threshold is $\tau = 0.5$, and the number of selected frames is bounded between 4 and 8. All training and evaluation experiments are conducted on two NVIDIA GeForce RTX 3090 GPUs.

3.1 Main results

Temporal Search Performance. Figure 4a reports temporal search on Haystack-LVBench. A pointwise top-k selector over our candidate pool and cross-encoder scores already nearly doubles the temporal F1 of TimeSearch-R, so much of the gap to prior work reflects fine-grained evidence scoring rather than the selection rule; we include this diagnostic to separate the two and exclude

Method	QA Model	# Frames	Temporal			Visual			QA	
			P	R	F_1	P	R	F_1	LVBench	Ego4D
Static Frame Sampling										
Uniform	Qwen2.5-VL-7B	8	1.4	6.3	2.2	56.0	72.0	62.7	33.7	32.0
Uniform	GPT-4o	8	1.4	6.3	2.2	56.0	72.0	62.7	47.1	41.5
Uniform	GPT-4o	32	1.4	24.9	2.7	58.7	81.6	67.3	50.5	45.5
Adaptive Temporal Search										
VideoAgent	GPT-4	10.1†	1.2	8.5	2.1	58.8	73.2	64.7	–	–
Retrieval-based	GPT-4o	8	1.5	6.3	2.3	63.1	65.5	64.1	–	–
T*	GPT-4o	8	1.6	7.1	2.5	58.4	72.7	64.3	51.9	45.0
Retrieval-based	GPT-4o	32	1.3	21.8	2.4	59.9	80.8	67.8	–	–
T*	GPT-4o	32	1.7	28.2	3.1	58.3	83.2	67.8	53.1	46.5
Text-Video Interleaved Reasoning										
TimeSearch-R	Qwen2.5-VL-7B	8.8†	5.4	22.3	8.1	63.2	**76.4**	69.2	52.3	38.5
TimeSearch-R	TimeSearch-R-7B	8.8†	5.4	22.3	8.1	63.2	76.4	69.2	52.1	53.5
Controlled Evidence Selection										
Cross-encoder + top-k‡	Qwen2.5-VL-7B	8	14.8	22.2	16.0	64.2	76.8	69.9	55.3	49.0
SetGain (Ours)	Qwen2.5-VL-7B	6.7†	**18.0**	**23.0**	**18.1**	**69.2**	70.8	**70.1**	**60.5**	**51.0**

(a) Temporal search and downstream QA.



(b) Temporal F1 across controlled frame budgets.

Figure 4: **Temporal search and downstream QA.** Temporal and visual P/R/F1 on Haystack-LVBench, QA on both Haystack splits; baselines from Ye et al. (2025); Pan et al. (2026a). Gray: reference only. Bold/underline: best and second-best among published methods and SetGain. ‡Controlled diagnostic sharing our pool, scorer, and answerer; not ranked. †Average adaptively selected frames (Appendix C).

Table 1: **Downstream QA accuracy (%)** on LongVideoBench, MLVU, and VideoMME. Bold and underline mark the best and second-best results. Uniform GPT-4o results are from the TimeSearch-R author response (Pan et al., 2026b). †Average number of adaptively selected frames.

Method	QA Model	# Frames	LongVideoBench val	MLVU m-avg	VideoMME Overall
Uniform	GPT-4o	8	56.7	58.1	63.9
Uniform	GPT-4o	32	61.1	66.6	**70.8**
VideoAgent	GPT-4	87†	–	–	56.0
T*	GPT-4o	8	–	–	56.5
Video-R1-7B	Video-R1-7B	32	56.4	61.6	59.9
TimeSearch-R	Qwen2.5-VL-7B	7.8†	57.1	68.6	59.1
SetGain (Ours)	Qwen2.5-VL-7B	7.1†	**63.7**	**69.3**	60.4

it from the ranking. Relative to it, SetGain improves F1 by 13.1% relative while selecting fewer frames, gaining precision at comparable recall and thus suppressing redundancy rather than retrieving more evidence. The same pattern holds against prior work: SetGain attains 2.23× the temporal F1 of TimeSearch-R and, despite lower visual recall, the highest visual precision and visual F1 with 24% fewer frames. It also gives the highest QA accuracy on both Haystack splits under the controlled setting (Appendix B), and at matched budgets it gains 11.7–17.5% over pointwise top-k, MMR, and DPP (Figure 4b).

Downstream QA Performance. Table 1 reports downstream QA. Using the same frozen Qwen2.5-VL-7B answerer, SetGain improves over TimeSearch-R on all three benchmarks while selecting fewer frames (7.1 on average), with the largest gain on LongVideoBench (+6.6 points). On VideoMME, many short videos and a stronger answerer favor uniform GPT-4o. Compact, set-conditioned selection therefore supports strong answering without large frame budgets.

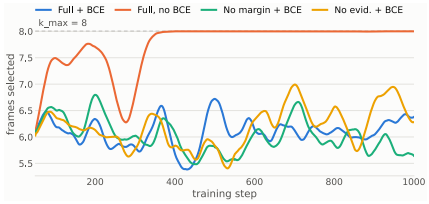
Inference Efficiency. Table 2 compares inference cost. SetGain completes the full search-and-answer pipeline in 11.40 s per question, a 2.53× speedup over TimeSearch-R and 2.15× over T*. SetGain and TimeSearch-R share the same SigLIP frame encoding and frozen answerer, so the difference lies almost entirely in how candidates are assessed: replacing 2.4 turns of autoregressive 7B reasoning with single-pass reranker calls cuts search latency by more than half and total compute by 46.8%. Compute alone does not determine latency, however. T* uses 19% fewer TFLOPs than SetGain yet runs 2.15× slower, since its detector calls are spread across dependent search iterations, whereas SetGain's evidence scores are computed independently and batched in parallel. The gain predictor itself adds only 0.025 s, and answering costs are comparable across both methods, so the end-to-end gap is set by search.

Table 2: **Inference efficiency per question.** Search covers candidate proposal and evidence selection; overall adds answering. †Estimated, as GPT-4 FLOPs are not public.

Method	Search Models / Calls	Avg. Frames	Search Efficiency		Overall Task Efficiency	
			TFLOPs ↓	Latency (s) ↓	TFLOPs ↓	Latency (s) ↓
VideoAgent	GPT-4 ×4; CLIP-1B ×840	10.1	536.5†	72.76	690.7†	83.35
T*	Qwen2.5-VL-7B ×1; YOLO-World-XL ×216	8.0	**381.5**	23.51	**425.1**	24.50
TimeSearch-R	SigLIP-400M ×465; TimeSearch-R-7B ×2.4 turns	8.8	946.9	27.85	987.9	28.79
Ours	Qwen2.5-VL-7B ×1; SigLIP-400M ×465; Qwen3-VL-Reranker-2B ×128	**6.7**	492.9	**10.63**	525.3	**11.40**

Variant	Stage 1	Acc.	Mar.	Evid.	Temporal Localization			
					T-P	T-R	T-F1	Frames
Cross-encoder + top-k					14.8	22.2	16.0	8.00
Stage 1 only	✓				16.1	23.1	17.2	7.70
Without evidence reward	✓	✓	✓		15.7	**24.1**	16.8	6.51
Without margin	✓	✓		✓	17.7	21.2	17.3	7.52
Full reward	✓	✓	✓	✓	**18.0**	23.0	**18.1**	6.70

(a) Training stages and reward components. Stage 1: marginal coverage regression; Acc.: correctness; Mar.: margin; Evid.: evidence coverage.



(b) Selected frames with and without stop-BCE.

Figure 5: **Reward and stopping analysis.** (a) Temporal localization for each combination of training stage and reward term. (b) Average selected frames during coverage-guided policy optimization, with and without stop-BCE; curves show a 50-step moving average over the 1,000 training steps.

4 Ablations

Reward Components. Figure 5a ablates the training stages and reward terms. Marginal coverage regression alone already improves temporal F1 over pointwise top-k while selecting fewer frames, and coverage-guided policy optimization improves substantially on this initialization only when all reward terms are present. Without the evidence reward, recall rises but precision falls and F1 drops below the initialization: answer-level feedback alone spreads selections across the video without discriminating which frames carry evidence. Without the margin reward the failure is reversed. Correctness alone is binary, so it is constant across any group of trajectories that agree in correctness and contributes no group-relative advantage; the comparison falls to the evidence reward, which raises precision but gives the lowest recall of all variants, so F1 barely moves from the initialization (17.3 vs. 17.2). Only the full reward retains both, achieving the best F1 with the second-smallest budget. Because each reward shapes the learned budget, frame counts are outcomes rather than controls; Figure 4b compares selection quality at matched k.

Adaptive Stopping. Figure 5b tracks the average number of selected frames over the 1,000 steps of coverage-guided policy optimization, with and without the stopping loss (stop-BCE, Eq. 8). Without it, the full-reward policy saturates at the eight-frame limit after roughly 350 steps, reducing SetGain to a fixed-budget selector through the collapse described in Sec. 2.4. With stop-BCE, the budget fluctuates between roughly 5.4 and 7.0 frames for all three reward variants and never approaches the limit. These curves average stochastic rollouts, in which both the selected frame and the stop action are sampled; at inference, where selection is greedy and stopping applies the fixed threshold $\tau = 0.5$, the same policies select 6.51–7.52 frames on average (Figure 5a). Stopping supervision is therefore necessary to prevent budget collapse. Appendix C probes the trained stopping head directly and shows that the budget it chooses varies with the question and with the frames already selected.

Set-conditioned features. Table 3 adds the components of the gain predictor one at a time, retraining both stages for each variant. A pointwise correction head trained with the same pipeline but no set state improves F1 over pointwise top-k, but its input is identical at every step, so its stopping probability is constant and it selects the maximum budget on every question. Adding the selected-set state enables adaptive stopping and raises precision, but recall falls and F1 is nearly unchanged: the policy knows which frames it has selected but cannot judge whether the remaining candidates would add new evidence, and stops too early. Adding ρ recovers recall at unchanged precision, consistent with ρ helping the policy distinguish candidates that add coverage from those already represented.

Table 3: **Cumulative ablation of set-conditioned features** on Haystack-LVBench. Each row adds one component to the row above, starting from a trained pointwise head; both training stages are rerun per variant. Without pool context, each candidate attends only to itself and the selected frames. Frames*: average selected frames.

Variant	T-P	T-R	T-F1	Frames*
Cross-encoder + top-k	14.8	22.2	16.0	8.00
Trained head, no set state	15.5	**23.7**	16.8	8.00
+ selected flag m	16.5	21.9	16.9	7.25
+ ρ	16.5	23.5	17.4	7.56
+ pool context (SetGain, full)	**18.0**	23.0	**18.1**	**6.70**

Pool context then gives the largest precision gain at comparable recall with the fewest frames. Overall, set conditioning contributes 1.3 F1 over an identically trained pointwise head (7.7% relative) with 16% fewer frames, again by improving precision rather than recall.

5 Related Work

Frame selection and temporal search. Long video models reduce visual input through uniform or heuristic sampling (Li et al., 2025a; Zhang et al., 2025; Chen et al., 2024; Bai et al., 2025). Question-guided methods select frames by relevance (Liang et al., 2024; Hu et al., 2025a;b), prediction utility (Yu et al., 2025), or spatiotemporal relationships (Li et al., 2023; Guo et al., 2025b), while diversified selection reduces repetition through determinantal point processes or similarity objectives (Carbonell & Goldstein, 1998; Kulesza & Taskar, 2012; Chen et al., 2018; Celis et al., 2018; Shen et al., 2025; Sun et al., 2025; Tang et al., 2025). Adaptive methods progressively explore videos through tool use in VideoAgent (Wang et al., 2024), hierarchical organization in VideoTree (Wang et al., 2025), coarse-to-fine search in T* and Iterative Zoom-In (Ye et al., 2025; Li et al., 2025b), and temporal spotlighting with reflection in TimeSearch (Pan et al., 2025). None of them learns what a candidate adds relative to the frames already selected.

Reinforcement learning for multimodal search. GRPO learns reasoning from outcome feedback (Shao et al., 2024; Guo et al., 2025a), and Search-R1 extends this approach to adaptive text retrieval (Jin et al., 2025). Multimodal applications include visual reasoning in MM-Eureka and LMM-R1 (Meng et al., 2025; Peng et al., 2026), spatial inspection in DeepEyes (Zheng et al., 2026), and video reasoning in Video-R1 (Feng et al., 2025). More closely related, TimeSearch-R combines temporal retrieval, reasoning, and self-verification (Pan et al., 2026a), while TSPO learns event-aware frame selection using QA and localization rewards (Tang et al., 2026), evaluated mainly through downstream QA; we evaluate the selected evidence directly. These methods supervise complete trajectories, leaving individual selection contributions implicit; SetGain instead adds intermediate coverage targets to learn set-conditioned marginal gain.

6 Conclusion

We introduced **SetGain**, a temporal retrieval framework for constructing compact, question-dependent evidence sets from long videos. Rather than scoring frames independently, SetGain predicts each candidate's additional contribution relative to the evolving selected set and stops once further frames add little. Marginal Gain Learning trains this policy in two stages, regressing marginal coverage at intermediate states and then optimizing complete trajectories by Coverage-Guided Policy Optimization against answer quality and evidence coverage. On Haystack-LVBench it attains 2.23× the temporal F1 of TimeSearch-R with higher precision at comparable recall and fewer frames, improves average QA accuracy over TimeSearch-R on LongVideoBench, MLVU, and VideoMME, and runs 2.15× faster end to end than the fastest prior search method. A limitation is that SetGain can fail when the answer lies outside the scene the question describes, such as questions asking where something appears again: it stops inside that scene before reaching the evidence elsewhere in the video (Appendix C.4). Tracking coverage per decomposed event is a natural direction for future work.

AI USE STATEMENT

We used large language models solely to polish the writing of this paper, including checking grammar, simplifying sentences, and improving the fluency of the text. LLMs were not used for literature review or research ideation. All research ideas, experiments, and analyses were conceived and carried out by the authors, who reviewed all LLM-assisted edits and take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

We have made every effort to ensure the reproducibility of our results. Anonymized source code is available at `https://anonymous.4open.science/r/SetGain`. The full method, including candidate proposal, the set-conditioned gain predictor, and both training stages, is described in Section 2, and the datasets, baselines, and implementation details are given in Section 3. Appendix A defines the marginal-coverage targets, and Appendix A.1 lists all hyperparameters. Appendix B describes the Haystack-Ego4D evaluation protocol. Appendix D reports the training and preprocessing compute, and Appendix E provides every prompt used by the proposer, scorer, and answerer. All training and evaluation datasets are publicly available.

REFERENCES

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

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pp. 335–336, New York, NY, USA, 1998. Association for Computing Machinery. doi: 10.1145/290941.291025. URL `https://doi.org/10.1145/290941.291025`.

Elisa Celis, Vijay Keswani, Damian Straszak, Amit Deshpande, Tarun Kathuria, and Nisheeth Vishnoi. Fair and diverse DPP-based data summarization. In *International Conference on Machine Learning*, pp. 716–725. PMLR, 2018.

Laming Chen, Guoxin Zhang, and Eric Zhou. Fast greedy MAP inference for determinantal point process to improve recommendation diversity. *Advances in Neural Information Processing Systems*, 31, 2018.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24185–24198, 2024.

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. *Advances in Neural Information Processing Systems*, 38:99114–99137, 2025.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025a. doi: 10.1038/s41586-025-09422-z.

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

Jian Hu, Zixu Cheng, Chenyang Si, Wei Li, and Shaogang Gong. CoS: Chain-of-shot prompting for long video understanding. *arXiv preprint arXiv:2502.06428*, 2025a.

Kai Hu, Feng Gao, Xiaohan Nie, Peng Zhou, Son Tran, Tal Neiman, Lingyun Wang, Mubarak Shah, Raffay Hamid, Bing Yin, et al. M-LLM based video frame selection for efficient video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13702–13712. IEEE, 2025b.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Conference on Language Modeling*, 2025.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, 2012.

Ariel Kwiatkowski, Natasha Butt, Ismail Labiad, Julia Kempe, and Yann Ollivier. Likelihood-based reward designs for general LLM reasoning. *arXiv preprint arXiv:2602.03979*, 2026.

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

Chenglin Li, Qianglong Chen, Fengtao, and Yin Zhang. Iterative Zoom-In: Temporal interval exploration for long video understanding. *arXiv preprint arXiv:2507.02946*, 2025b.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, et al. Qwen3-VL-Embedding and Qwen3-VL-Reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

Yicong Li, Junbin Xiao, Chun Feng, Xiang Wang, and Tat-Seng Chua. Discovering spatio-temporal rationales for video question answering. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13823–13832. IEEE, 2023.

Hao Liang, Jiapeng Li, Tianyi Bai, Xijie Huang, Linzhuang Sun, Zhengren Wang, Conghui He, Bin Cui, Chong Chen, and Wentao Zhang. KeyVideoLLM: Towards large-scale video keyframe selection. *arXiv preprint arXiv:2407.03104*, 2024.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, et al. MM-Eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.

Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual VQA: A cause-effect look at language bias. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12695–12705, 2021.

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*, 2019.

Junwen Pan, Rui Zhang, Xin Wan, Yuan Zhang, Ming Lu, and Qi She. TimeSearch: Hierarchical video search with spotlight and reflection for human-like long video understanding. *arXiv preprint arXiv:2504.01407*, 2025.

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. In *International Conference on Learning Representations*, 2026a.

Junwen Pan, Qizhe Zhang, Rui Zhang, Ming Lu, Xin Wan, Yuan Zhang, Chang Liu, and Qi She. Author response to reviewers for TimeSearch-R: Adaptive temporal search for long-form video understanding via self-verification reinforcement learning. OpenReview discussion, ICLR 2026, 2026b. URL https://openreview.net/forum?id=gqb1hvuGcj.

Yingzhe Peng, Gongrui Zhang, Miaosen Zhang, Zhiyuan You, Jie Liu, Qipeng Zhu, Kai Yang, Xingzhong Xu, Xin Geng, and Xu Yang. LMM-R1: Empowering 3B LMMs with strong reasoning abilities with two-stage rule-based RL. *Frontiers of Computer Science*, 2026. doi: 10.1007/s11704-026-51776-8.

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

CJ van Rijsbergen. *Information retrieval*. Butterworth-Heinemann, 1979.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. LongVU: Spatiotemporal adaptive compression for long video-language understanding. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 54582–54599, 2025.

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

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

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29118–29128, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Liaoyaqi Wang, Chunsheng Zuo, William Jurayj, Benjamin Van Durme, and Anqi Liu. Process supervision of confidence margin for calibrated LLM reasoning. In *Conference on Language Modeling*, 2026.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. VideoAgent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pp. 58–76, 2024.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3272–3282, 2025.

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

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

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

Xiangyu Zeng, Zhiqiu Zhang, Yuhan Zhu, Xinhao Li, Zikang Wang, Changlian Ma, Qingyu Zhang, Zizheng Huang, Kun Ouyang, Tianxiang Jiang, et al. Video-o3: Native interleaved clue seeking for long video multi-hop reasoning. In *International Conference on Machine Learning*, 2026.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952, 2023.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video instruction tuning with synthetic data. *Transactions on Machine Learning Research*, 2025.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. DeepEyes: Incentivizing "thinking with images" via reinforcement learning. In *International Conference on Learning Representations*, 2026.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701, 2025.

APPENDIX

This appendix provides further details on our method, evaluation protocol, selection behavior, training cost, and prompts, organized as follows:

- Section A: Coverage Compatibility and Marginal-Coverage Targets, with hyperparameters
- Section B: Comparison on Haystack-Ego4D
- Section C: What Determines the Number of Selected Frames, with case studies
- Section D: Training and Preprocessing Cost
- Section E: Prompts

A COVERAGE COMPATIBILITY AND MARGINAL-COVERAGE TARGETS

Section 2.4 defines the marginal-coverage target $\widehat{G}_i^{(r)}$ (Eq. 6) through a compatibility matrix $D_{ji} \in [0,1]$ between an annotated evidence segment $g_j = [\tau_j^{\mathrm{s}}, \tau_j^{\mathrm{e}}]$ and a candidate $x_i = (I_i, t_i)$. This appendix specifies D_{ji} and the two places it is used: the regression targets of Eq. 6 and the soft F_β evidence-coverage reward R_{cov} of Eq. 7.

Definition. All quantities here are in seconds: t_i is the absolute timestamp of candidate x_i, not the normalized feature $\tilde{t}_i$ of Eq. 4. Let d_{ji} denote the distance from t_i to the segment g_j,

$$d_{ji} = \max\big(\tau_j^{\mathrm{s}} - t_i,\ t_i - \tau_j^{\mathrm{e}},\ 0\big)\,, \tag{10}$$

so that $d_{ji} = 0$ whenever t_i falls inside the annotated segment. The compatibility mixes a temporal and a semantic term,

$$D_{ji} = (1-\alpha)\underbrace{\exp\!\left(-\frac{d_{ji}}{\sigma_d}\right)}_{\text{temporal}} + \alpha\underbrace{\sigma\!\left(\frac{z_{ji}-z_0}{\gamma_s}\right)}_{\text{semantic}}, \tag{11}$$

where $\sigma(\cdot)$ is the logistic function, σ_d sets the temporal tolerance, z_0 and γ_s center and scale the semantic term, and $\alpha \in [0,1]$ balances the two. The temporal term decays smoothly with d_{ji} rather than truncating at the segment boundary, so a candidate just outside a segment still receives a nonzero value.

Semantic agreement z_{ji}. The semantic term compares a candidate against the *appearance* of an annotated segment, using the proposal encoder's image tower only; neither the question nor any text embedding enters z_{ji}. Let $\bar{\mathbf{e}}_i$ be the normalized embedding of candidate x_i and let

$$\bar{\mathbf{e}}_{g_j} = \frac{\sum_{v \in g_j} \bar{\mathbf{e}}_v}{\left\| \sum_{v \in g_j} \bar{\mathbf{e}}_v \right\|} \tag{12}$$

be the segment centroid, the renormalized mean embedding of the annotated frames of g_j, renormalized so that it lies on the unit sphere like the candidate embeddings. The raw agreement is the cosine similarity $s_{ji} = \langle \bar{\mathbf{e}}_{g_j}, \bar{\mathbf{e}}_i \rangle$.

The raw cosine is standardized per segment against a within-video background. Let $\mathcal{B}_j = \{\langle \bar{\mathbf{e}}_{g_j}, \bar{\mathbf{e}}_r \rangle\}$ be the similarities of g_j to R frames sampled uniformly at random from the same video, and let μ_j and ς_j be its mean and standard deviation. Then

$$z_{ji} = \frac{s_{ji} - \mu_j}{\varsigma_j}, \tag{13}$$

so z_{ji} is expressed in standard deviations of what a typical frame of that video scores against the segment. We use $R = 64$ background frames with a fixed sampling seed, deduplicated, and require at least 8 to remain; when fewer survive, the semantic term is unavailable for that video and the compatibility falls back to the temporal term.

The standardization sets the units of z_0 and γ_s: both are expressed in standard deviations of the within-video background, so $z_0 = 2$ places the midpoint of the logistic two standard deviations above that background and $\gamma_s = 1$ spreads it over one.

Settings. We use $\sigma_d = 4.0\,\text{s}$, $z_0 = 2.0$, and $\gamma_s = 1.0$ throughout. The temporal term retains 0.78 at one second outside a segment and 0.29 at five seconds. The mixing weight α differs between the two uses of D: the regression targets of Eq. 6 use $\alpha = 0$, so they depend on temporal distance alone, while the evidence-coverage reward of Eq. 7 uses $\alpha = 0.5$, so appearance agreement enters the reward but not the targets.

Use in the regression targets. Given D, the current coverage of segment g_j by the selected set is $c_j^{(r)} = \max_{x_\ell \in \mathcal{S}_{r-1}} D_{j\ell}$, and the marginal target $\widehat{G}_i^{(r)}$ of Eq. 6 averages the positive part of $D_{ji} - c_j^{(r)}$ over segments. The target is therefore non-zero only for candidates whose compatibility with some segment exceeds the coverage the selected set already provides: a candidate close to an already covered segment contributes $D_{ji} - c_j^{(r)} \approx 0$ for that segment, however large D_{ji} is on its own. The stopping target $y_{\text{stop}}^{(r)}$ fires once no remaining candidate exceeds the threshold $\epsilon = 0.05$; the same threshold defines the stop targets of $\mathcal{L}_{\text{stop}}$ (Eq. 8).

Use in the coverage reward. The evidence-coverage reward R_{cov} of Eq. 7 reuses the same matrix. For a completed trajectory with selected set $\mathcal{S}$, the soft precision and recall are

$$P_{\text{soft}} = \frac{1}{|\mathcal{S}|} \sum_{x_i \in \mathcal{S}} \max_j D_{ji}, \qquad R_{\text{soft}} = \frac{1}{M} \sum_{j=1}^{M} \max_{x_i \in \mathcal{S}} D_{ji}, \tag{14}$$

and

$$R_{\text{cov}} = \frac{(1+\beta^2)\, P_{\text{soft}} R_{\text{soft}}}{\beta^2 P_{\text{soft}} + R_{\text{soft}}}. \tag{15}$$

Soft recall averages, over the annotated segments, the compatibility of the best-matching selected frame, and soft precision averages, over the selected frames, the compatibility of the best-matching segment. Both use a maximum, so neither term separates a frame that covers a segment for the first time from a second frame covering the same segment; that distinction is made by the marginal targets of Eq. 6 and by the stopping criterion. Adding a frame changes P_{soft} according to whether its compatibility with the nearest segment is above or below the current mean, and can only leave R_{soft} unchanged or increase it. β sets the weight of recall relative to precision; we use $\beta = 1.5$ (Sec. 3).

A.1 Hyperparameters

Table 4 lists the hyperparameters of Eqs. 6–9 and of D_{ji} for the reported run; runs and ablations share these values unless an ablation states otherwise. Several terms of Sec. 2.4 are defined only in words there, and we make them precise here.

Answer margin and baseline. For a trajectory whose selected frames yield answer distribution p, the margin is $\delta = \log p(a^\star) - \max_{a \neq a^\star} \log p(a)$. The baseline b_q is the same margin computed by the same frozen answerer on eight frames spaced uniformly by index over the 128-frame candidate pool, in timeline order. The margin term $\sigma((\delta - b_q)/\gamma)$ therefore exceeds 0.5 only when a trajectory separates the correct answer more clearly than uniform sampling of the same pool. The correctness and margin weights sum to one, $w_{\text{mar}} = 1 - w_{\text{acc}}$.

Coverage weight. The weight on R_{cov} in Eq. 7 depends on the number of annotated segments M of the question, $\lambda = \lambda_0 \min(1, M/2)$, which gives 0.35 for single-segment questions and 0.7 for questions with two or more segments.

Stopping supervision. The greedy rollout of Eq. 8 always runs all $H = 8$ steps, the maximum frame budget, regardless of the stop targets it encounters, and the loss is the per-step BCE averaged over those steps. The threshold ϵ is shared by the regression targets of marginal coverage regression and the stop targets of $\mathcal{L}_{\text{stop}}$.

Sampling temperature. The temperature applied when sampling trajectories decreases linearly from 1.5 to 0.8 over the first 800 of the 1,000 steps and is held at 0.8 afterwards.

Table 4: **Hyperparameters of the reported run.** M is the number of annotated evidence segments of the question.

Symbol	Role	Value
Answer reward (Eq. 7)		
w_{acc}	correctness weight	0.5
w_{mar}	margin weight	0.5
γ	margin temperature	2.0
b_q	margin baseline	margin on 8 pool frames spaced uniformly by index
Coverage reward (Eqs. 7, 15)		
λ	weight on R_{cov}	$\lambda_0 \min(1, M/2)$, $\lambda_0 = 0.7$
β	F_β balance	1.5
σ_d	temporal tolerance of D_{ji}	4.0 s
$z_0,\ \gamma_s$	semantic center, scale	2.0, 1.0
α	semantic mix	0 (targets), 0.5 (reward)
Stopping (Eqs. 6, 8)		
ϵ	stop threshold on marginal gain	0.05
H	greedy rollout length	8
λ_{stop}	weight on $\mathcal{L}_{\mathrm{stop}}$	1.0
Policy optimization (Eq. 9)		
K	trajectories per question	8
–	questions per step	8
–	sampling temperature	$1.5 \rightarrow 0.8$, linear over 800 steps
β_{KL}	KL weight	0.05
–	optimizer, learning rate	AdamW, 3×10^{-4}
–	training steps	1,000
Inference		
τ	stop threshold on p_{stop}	0.5
$k_{\min},\ k_{\max}$	frame budget	4, 8

B Comparison on Haystack-Ego4D

Figure 4a reports QA accuracy on Haystack-Ego4D in two settings for TimeSearch-R (Pan et al., 2026a). In the controlled setting, its searched frames are passed to the same frozen Qwen2.5-VL-7B answerer used by SetGain, giving 38.5; its published end-to-end result, obtained with its own TimeSearch-R-7B answerer, is 53.5 and is shown in gray.

The gap between these two numbers is far larger on Haystack-Ego4D than on Haystack-LVBench, where the same substitution moves accuracy only from 52.1 to 52.3. TimeSearch-R trains on Ego4D data (Pan et al., 2026a), so its own answerer has been adapted to the visual domain and question style of this split. Its published Haystack-Ego4D accuracy therefore reflects answering ability acquired during training rather than a better allocation of the frame budget: the frames supplied to the answerer are identical in both settings, and only the answerer differs.

This is precisely what the controlled protocol is designed to separate. Since SetGain trains no answerer and leaves the answerer frozen, comparing against the published number would confound evidence selection with answerer adaptation. We therefore rank the controlled setting, in which both methods answer from the same frozen model and differ only in which frames they select, and report the published result in gray for reference.

C What Determines the Number of Selected Frames

SetGain adapts the number of frames to the evidence required by each question. We examine how selection length varies with video duration and how the termination decision responds to the frames already selected, followed by qualitative examples. At inference, selection terminates when at least four frames have been selected and $p_{\mathrm{stop}} \geq 0.5$, or when the maximum of eight frames is reached.

C.1 Relationship with Video Duration

Figure 6 shows that SetGain generally selects more frames for longer videos, with the increase saturating at the upper end. Across the 1,337 LongVideoBench validation questions, the mean number of selected frames rises from 5.11 in the shortest duration group to 6.73 in the longest. However, the two longest groups have nearly identical means of 6.72 and 6.73 frames. The adaptive mechanism therefore produces selection lengths that correlate with video duration, while the increase levels off for the longest videos.

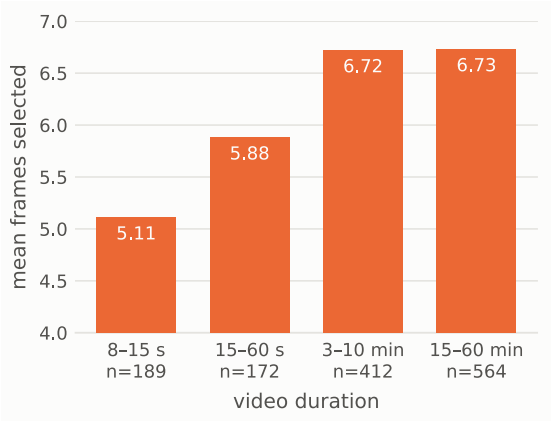


Figure 6: **Mean selected frames by video duration** on the LongVideoBench validation set, using its four duration groups; n is the number of questions in each group.

C.2 Dependence on the Selected Frames

Changes during selection. Selected-set features allow the stopping head to learn how its decision should change as frames are added. Table 5 shows that the mean p_{stop} increases from 0.018 before selection to 0.387 after eight frames. Without these features, the head receives the same input at every step, so its output remains unchanged and never triggers termination in the reported ablation. Consequently, all LV-HAYSTACK questions reach the maximum of eight frames (Table 3).

Table 5: **Mean p_{stop} against the number of frames already selected**, averaged over LV-HAYSTACK.

Frames selected	0	1	2	3	4	5	6	7	8
Mean p_{stop}	0.018	0.156	0.197	0.236	0.274	0.306	0.337	0.362	0.387

Quality of the accumulated selection. Table 6 compares different sets of four frames while keeping the question and candidate pool fixed. SetGain's first four selections produce the highest mean p_{stop} among these alternatives, at 0.274. Retaining its first selection and adding the three nearest neighbors gives 0.240, while adding the three least similar frames gives 0.205. Both exceed the 0.163 obtained by selecting four frames randomly.

Because the set size is fixed at four in every row, these differences cannot come from the number of selected frames. The stopping head therefore responds to which frames are selected, not only to how many.

C.3 Selection Length and Required Visual Context

Figure 7 compares two correctly answered questions from the same video (`mq6L8CnNJXc`) that lead to different selection lengths.

Bag fastener (lvbench_00216). "On a white shelf, there is a pair of mismatched shoes, and next to the shoes, there's a wooden-colored bag. What kind of fastener does the bag in the video have?"

Table 6: **What the stopping head reads from the selected set.** For each question the question and the candidate pool are unchanged and exactly four frames are marked as selected; only their identity differs across rows. Mean p_{stop} over LV-HAYSTACK.

Which four frames are marked as selected	Mean p_{stop}
The four the head itself picks first	0.274
Its first pick and that frame's three nearest neighbors	0.240
Its first pick and the three frames least similar to it	0.205
Four frames drawn at random from the pool	0.163

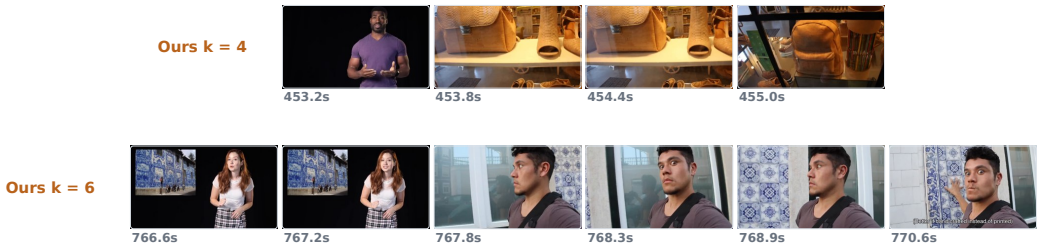


Figure 7: **Two questions on video `mq6L8CnNJXc`.** Top: `lvbench_00216`, which fastener the bag has, stops after four frames. Bottom: `lvbench_00217`, the color of the backpack straps, stops after six. Frames are shown in timeline order.

This question focuses on a localized bag attribute. Four selected frames support the correct answer, a silver square fastener, and p_{stop} reaches 0.94.

Backpack strap color (lvbench_00217). "In front of a wall with blue ceramic tiles, there's a man with a mustache, carrying a backpack and wearing a short-sleeved shirt. What is the color of the straps on the backpack he's carrying in the video?"

This question requires matching more details about the person and scene. After four selections, p_{stop} remains at 0.13. Two additional views raise it to 0.71, triggering termination with the correct answer, black.

These examples suggest that SetGain selects fewer frames for localized evidence and collects more when the question requires richer visual context.

C.4 Premature Termination When Additional Context Is Required

Figure 8 shows questions that describe a reference scene but ask about appearances elsewhere in the video. Resolving these questions requires broader context beyond the described scene. In all three cases, SetGain concentrates its selections around the reference scene and terminates with incomplete evidence, producing an incorrect answer.

Goateed man (lvbench_00318). "On a stretch of yellow sand bordered by jade-green seawater, a man wearing a grey short-sleeve shirt, glasses, and sporting a goatee is taking a selfie. In which of the following scenes does this goateed man appear?"

SetGain selects five frames from the beach scene at 7.0–9.0 s and terminates with $p_{\text{stop}} = 0.52$, missing the required appearance inside a large turtle tank.

Yellow hat (lvbench_00139). "The man wearing glasses with a black backpack is outside the airport. He is wearing a yellow hat, a black mask, and a grey hoodie. He is surrounded by white columns and large glass windows. Where else does the man's hat appear?"

The six selected frames concentrate on the airport sequence. Selection terminates at $p_{\text{stop}} = 0.51$, although the hat's appearance in the elevator remains uncovered.

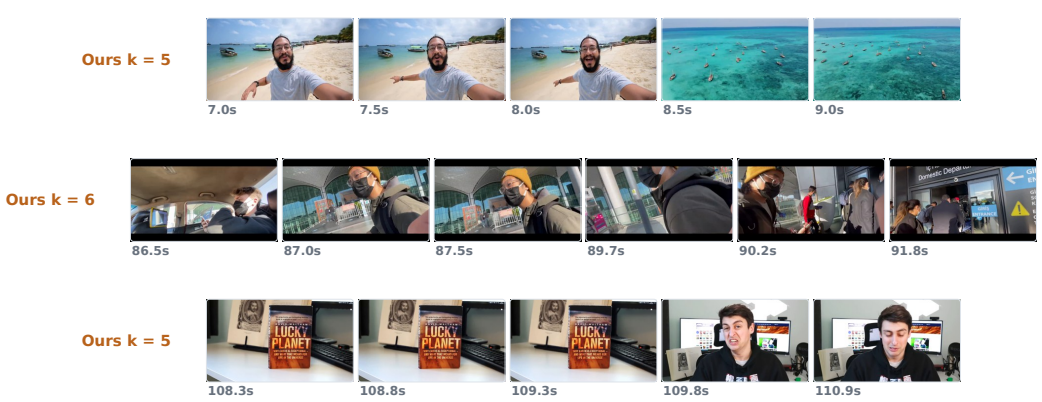


Figure 8: **Three questions whose answer lies outside the scene they describe.** Top: `lvbench_00318`, in which scenes the goateed man appears. Middle: `lvbench_00139`, where else the hat appears. Bottom: `lvbench_00242`, in which scenes the book appears. All three stop inside the described scene and answer incorrectly.

Book (lvbench_00242). "On a white desk with an open book and a black keyboard, there is an orange-red book with the words 'LUCKY PLANET' on the cover. In which of the following scenes has this book appeared?"

The five selected frames span only 2.6 s around the desk scene. Selection terminates at $p_{\text{stop}} = 0.62$, missing the book's appearance in the hands of a man wearing black scrubs.

These cases suggest that SetGain can underestimate the additional context needed when the question requires finding appearances beyond the described scene. Repeated evidence from that scene can trigger termination before the ambiguity is resolved. Tracking which required events remain uncovered could help address this limitation.

D Training and Preprocessing Cost

All experiments use NVIDIA GeForce RTX 3090 GPUs (24 GB). Training and evaluation use at most two of them, as stated in Sec. 3; the one-time preprocessing of Appendix D.1 is embarrassingly parallel and was sharded across four. No vision–language model is finetuned: the proposal encoder, the cross-encoder evidence scorer, and the answerer remain frozen throughout, so the only trained parameters are the 2.17M of the gain predictor and stopping head.

D.1 One-time offline preprocessing

The stages in Table 7 are paid once and shared by every run and every ablation. The cross-encoder pass dominates. Because the scorer is frozen, s_i is a constant of the pair (question, candidate pool) and is cached, with a 99.2% hit rate across both training stages and all ablations.

Table 7: **One-time preprocessing cost**, shared across all runs.

Stage	Scale	Wall clock	GPU-hours
Frame-feature cache (SigLIP-400M)	7,153 videos, 6.71M frames	8.5 h	32.0
Event decomposition (Qwen2.5-VL-7B)	14,372 sampled from Seeker-173K, 5.6 events each	0.6 h	2.3
Cross-encoder scores over pools	14,372 questions $\times$ 128 candidates $=$ 1.84M pairs	15.7 h	62.7
Training-cache assembly	14,166 usable questions	2 min	0.03
Total			$\approx$ 97.0

D.2 Stage 1: marginal coverage regression

No vision–language model is in the loop; the head trains directly from the cache on a single GPU. Training runs for the 1,000 steps of Sec. 3 at batch size 128, which is 128,000 question visits, or nine passes over the 14,166 cached questions, and completes in 145 minutes at a cost of **2.4 GPU-hours**. The final step is the checkpoint used; no intermediate checkpoint is selected.

D.3 Stage 2: coverage-guided policy optimization

The step budget is fixed in advance and identical for every run and every ablation, so no evaluation data enters training or checkpoint selection.

Table 8: **Cost of coverage-guided policy optimization.**

Steps	1,000, 8 questions/step, $K = 8$ trajectories per question
Answerer passes	$1{,}000 \times 8 \times (8 + 1) = 72{,}000$ frozen 7B passes (K trajectories + b_q)
Throughput	0.186 questions/s end-to-end
Wall clock	11 h 57 min
GPUs	2 × RTX 3090 (policy and frozen answerer on separate devices)
Cost	**23.9 GPU-hours**

D.4 Totals

Training SetGain costs **approximately 26 GPU-hours** (2.4 + 23.9) on top of approximately 97 GPU-hours of one-time, reusable preprocessing, for 2.17M trainable parameters. Every ablation repays only Appendices D.2 and D.3, since the caches are shared. At inference, the policy adds one forward pass of the 2.17M-parameter head per selection step, and the 128-candidate cross-encoder sweep is never repeated inside the selection loop (Table 2).

E Prompts

Four prompts are used, given below in pipeline order; all are read by frozen models. Three are instruction prompts, and the fourth is the set of plain event captions given to the proposal encoder.

E.1 Event decomposition

This prompt turns one question into the events $\mathcal{Q} = \{q_k\}$ of Sec. 2.2, expressed as visual captions that the proposal encoder retrieves against. The system message is:

```
You are a precise assistant that extracts visual search queries from
questions about videos.
```

Every question in every benchmark we evaluate, including Haystack-Ego4D, carries answer options, so the same multiple-choice form is used throughout:

```
A multiple-choice question about a long video is given below. To
answer it, one must find specific scenes in the video by visual
search. Turn it into image-search captions.

Output a JSON array of short standalone image captions (what a frame
looks like), 4-15 words each, following these rules:

1. ANCHOR: if the question stem describes a specific scene (people,
   clothing, objects, setting), include that scene as one caption. It
   localizes the question.
2. Look at the OPTIONS:
   - If the options are REORDERINGS of the same set of events (a
     sequence / order / "what appears first" question where the same
     events recur in different order), list each underlying event
```

```
     exactly ONCE.
   - OTHERWISE the options are DISTINCT candidate answers: output one
     caption PER option, each describing the concrete scene or subject
     that option refers to, so every candidate can be located and
     checked.
3. No option letters, no ordering words (first/then/finally), no
   question phrases ("in which scene", "which of the following",
   "appears first"). Only concrete visual content - people, clothes,
   objects, colors, places.

Return ONLY the JSON array.

Question: {question}
Options:
{options}

JSON array:
```

The model returns 5.6 events per question on average over the 14,372 training questions. Each sampled frame is scored against every event and summarized by its best match, and the candidate pool is the top $B = 128$ frames of the video under that score (Eq. 3).

E.2 Proposal-encoder query

The proposal encoder takes no prompt template. In all reported runs, its text tower is given only the event captions of Appendix E.1, one string per event, with no question or option text; each frame is scored by its best-matching event (Eq. 3).

E.3 Evidence score

This prompt produces the evidence score s_i of Sec. 2.3. The query string, in the question-with-options mode used in all experiments, is:

```
{question}
{options}
```

and the retrieval instruction is passed as the system message:

```
Given a question about a video, retrieve the video frames that show
the visual evidence needed to answer it.
```

The reranker's own chat template assembles these into:

```
<|im_start|>system
Judge whether the Document meets the requirements based on the Query
and the Instruct provided. Note that the answer can only be "yes" or
"no".<|im_end|>
<|im_start|>user
<Instruct>: Given a question about a video, retrieve the video frames
that show the visual evidence needed to answer it.<Query>:{question}
{options}
<Document>:<|vision_start|><|image_pad|><|vision_end|><|im_end|>
<|im_start|>assistant
```

The score s_i is the yes-versus-no logit margin at the final position, read in fp32, rather than a generated token. One frame is processed per forward pass, which is what makes s_i independent across candidates and motivates set-conditioned selection.

E.4 Answering

The same prompt is used for the reward of Eq. 7 and for every reported QA number. The system message is:

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

The user turn contains the selected frames in chronological order, each preceded by its timestamp as text ("{t:.1f}s" followed by the image), and then:

```
Question: {question}
{options}
Answer with the option's letter from the given choices directly.
```

Nothing is decoded. The answer is the argmax over the option-letter token logits at the final position, log-softmax renormalized over the option letters only, which also yields the log-probability gap $\delta = \log p(a^\star) - \max_{a \neq a^\star} \log p(a)$ used by the margin term of Eq. 7.

Two further prompt styles exist in our codebase for baseline comparisons and are not used for any reported number: a free-form style that justifies the answer before emitting a final answer line, and a tool-call style that replays a frame-seeking tool call so that a tuned search model sees its own training format.