SAVE: SELF-COMPRESSING AGENTS FOR LONG VIDEO EVIDENCE VIA MULTI-QUESTION PRESSURE

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-video understanding agents are increasingly used to tackle complex long video question answering tasks through extensible tools and multi-step interactions. However, active evidence search in long videos introduces two key bottlenecks: high inference cost caused by repeated tool calls and lengthy observation sequences, and reasoning distraction caused by irrelevant or redundant evidence. Existing methods typically compress visual tokens or textual chains of thought, but overlook tool-mediated evidence search, which dominates the cost of long video agents. We propose **SAVE** (***S**elf-compressing **A**gents for Long **V**ideo **E**vidence*), a training-time self-compression framework for long video agents. SAVE uses another question from the same video as a *Pressure Question* and jointly solves it with the *Main Question* within a shared multi-turn video-tool rollout. The resulting implicit resource pressure encourages the agent to allocate evidence-search steps more selectively. SAVE then converts the resulting joint rollouts into single-question supervision by retaining target-relevant and shared tool interactions while removing pressure-question-specific searches and answers. SAVE supports both supervised fine-tuning and reinforcement learning. Experiments on multiple long video benchmarks show that SAVE reduces tool calls and token consumption while maintaining competitive accuracy.

1 INTRODUCTION

Long-video question answering requires localizing evidence that may appear only briefly within minutes or hours of content. Recent systems therefore move beyond single-pass video-language inference and adopt an *agentic* formulation: an agent searches candidate temporal regions, invokes caption-retrieval or local video question-answering (video-QA) tools, inspects returned observations, and revises its plan before answering (Wang et al., 2024; 2025b; Yang et al., 2026b; 2025; Zhang et al., 2025). This flexibility is essential when evidence is sparse, temporally localized, or distributed across multiple moments.

However, active evidence acquisition introduces a distinct efficiency bottleneck. As illustrated in Figure 1 left, each speculative tool call incurs visual processing and appends an observation to the agent's growing interaction history; the agent must additionally remember which regions were inspected, which observations support its current hypothesis, and what still requires verification. During interaction, agents may repeatedly inspect nearby regions, retrieve overly broad clips, or continue verification after sufficient evidence has been obtained, creating tool-mediated evidence-search redundancy. Such interactions consume computation and context without materially improving the localization or verification needed to answer the target question.

As illustrated in Figure 1 right (a)(b), existing long-horizon agent compression introduces efficiency at two distinct points. **Online context compression** summarizes histories or observations only after the corresponding tool calls have occurred and requires an additional compressor during execution (Ye et al., 2026; Kang et al., 2026). **Teacher-guided trajectory pruning** first generates a complete trajectory, then relies on an external model to identify removable actions; this incurs full-rollout and teacher cost, and can remove dependent temporal evidence when transferred to a long video agent. Recent reasoning methods provide a complementary clue: multi-question contexts can induce models to allocate a finite reasoning budget more selectively (Deng et al., 2026; Yang et al., 2026a). Extending this mechanism to long video agents requires a careful trajectory construction

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.

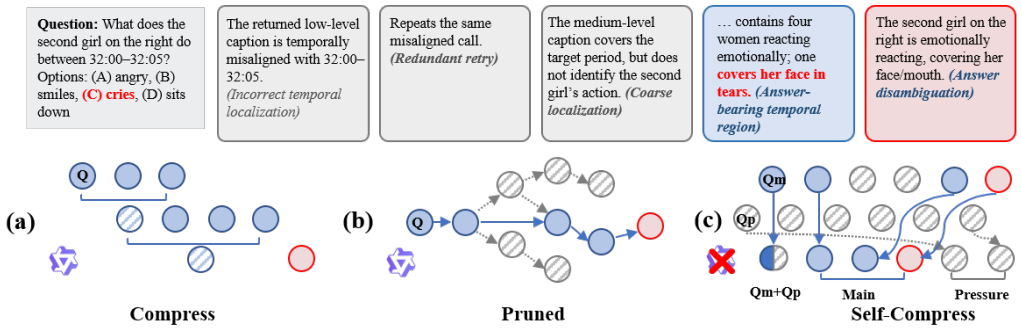


Figure 1: Left: a long video tool trajectory with temporal mislocalization, redundant retrieval, and subsequent answer disambiguation. Right: (a) **online context compression**, (b) **teacher-guided trajectory pruning**, and (c) **SAVE**. Blue nodes denote Main-Question-relevant or retained interactions, gray hatched nodes denote compressed, pruned, or Pressure-Question-only interactions, and red nodes denote final answers.

procedure, since questions from the same video may access overlapping temporal evidence through shared tools, yielding joint interactions that cannot be directly used to train a single question policy.

To address this problem, we propose the **SAVE** (***S**elf-compressing **A**gents for long **V**ideo **E**vidence*) framework. As illustrated in Figure 1 right (c), SAVE defines the question to be solved for a video as the *Main Question* (Q_m) and treats another question associated with the same video as a *Pressure Question* (Q_p). Addressing these two issues typically requires two separate rollouts. In the SAVE, Main and Pressure questions are jointly solved in a shared video-tool environment under a shared interaction budget, producing a joint trajectory; this finite shared budget creates implicit resource pressure that encourages the source agent to allocate evidence-search steps more selectively and suppress repeated inspection and low-value retrieval. SAVE then extracts deployable single-question supervision from the joint trajectory with the Main Question as its target: it retains target-specific and shared tool interactions observations and answers, while removing pressure search and reasoning. During supervised fine-tuning (SFT), the agent learns a search policy for a single Main Question from compact target trajectories. During reinforcement learning, SAVE further refines this policy with *Main-Question-Rewarded Joint GRPO*: the prompt and rollout still contain the joint Main Question and Pressure Question, which share the same interaction budget, while the reward uses only the Main Question answer correctness and output format. No explicit penalty is applied to tool calls or token length. At test time, the model receives only the original Main Question, requiring neither a Pressure Question nor an additional compressor.

Our contributions are as follows:

- We propose **SAVE**, a self-compression framework for long video agents that transfers same-video multi-question pressure to single-question deployment. SAVE combines shared-environment pressure rollouts, target-centric trajectory extraction, and Main-Question-Rewarded Joint GRPO to learn compact tool-use policies without requiring a pressure question at test time.
- We conduct extensive experiments on long video benchmarks. SAVE achieves a favorable accuracy–efficiency trade-off, reducing tool-mediated interaction and token costs while maintaining competitive video-QA accuracy.

2 Related Work

2.1 Long-Video Understanding Agents

Long-video understanding aims to localize and integrate temporally grounded evidence from videos lasting minutes or even hours. Existing methods include end-to-end video-language models (Song et al., 2024; Shen et al., 2025; Shu et al., 2025) and query-aware video selection methods (Wang et al., 2025b; Li et al., 2026; Ye et al., 2025). Building on these approaches, long-video understand-

ing agents formulate video question answering as an iterative evidence-search process. VideoAgent models long-form video question answering as an interactive process in which a large language model (LLM) agent progressively gathers visual information through external tools Wang et al. (2024). VideoTree constructs a query-adaptive hierarchical representation to select relevant video information in a coarse-to-fine manner Wang et al. (2025b). More recent work, such as LongVT, further supports interleaved multimodal tool-calling reasoning, allowing the model to inspect temporal clips on demand during inference Yang et al. (2026b). VCA uses curiosity-driven tree search to explore video segments, while Deep Video Discovery equips an agent with multi-granular search tools for adaptive evidence acquisition (Yang et al., 2025; Zhang et al., 2025). Although active evidence acquisition can improve reasoning when evidence is sparse, repeatedly searching and verifying temporal regions accumulates observations and incurs substantial redundant interaction cost. We therefore introduce SAVE to reduce redundant tool-mediated evidence search while preserving the temporal evidence required to answer the target question.

2.2 Long-Horizon Agent Compression

Long-horizon agents accumulate actions, observations, and intermediate states over continuous tool interactions, increasing inference cost and exposing later decisions to irrelevant history. Existing methods for agents operating over long horizons mainly reduce the cost of information that has already been acquired. Context compression methods, such as AgentFold and ACON, fold, summarize, or rewrite interaction histories and observations after they are produced (Ye et al., 2026; Kang et al., 2026). Post hoc trajectory pruning methods, such as WebClipper, execute a complete web agent trajectory, construct a state graph to identify removable nodes, and rewrite reasoning gaps after pruning (Wang et al., 2026). For long video agents, however, useful trajectories must preserve the dependency between coarse temporal localization and subsequent local verification. Removing or summarizing an interaction without accounting for this dependency can discard evidence required to answer the question. SAVE instead uses multiple questions from the same video during rollout to shape evidence search, and then converts the resulting joint trajectories into supervision for deployment on a single question.

2.3 Multi-Question Reasoning Compression

Early multi-question prompting primarily treats several questions as an input construction or evaluation setting, for example by packing independent samples into one prompt, stress-testing reasoning under joint inputs, or fusing related mathematical problems (Cheng et al., 2023; Pan et al., 2026; Pei et al., 2025). Recent work instead uses multiple questions as an implicit resource allocation signal: ConPress observes that several questions in one context induce shorter per-question reasoning traces and transfers the resulting compact behavior to single-question inference (Deng et al., 2026), while BCR trains reasoning models with multi-problem GRPO under a shared completion-token budget, making problems compete for a finite generation resource without an explicit length penalty (Yang et al., 2026a). These methods mainly compress textual completions, whereas long video agents must additionally decide which temporal regions to search and verify, with tool observations depending on prior localization actions. Moreover, joint trajectories for multiple questions from the same video contain tool interactions that support the Main Question, both questions, or only the Pressure Question. They therefore cannot be directly used to train an agent for deployment on a single question. SAVE addresses this issue by inducing compact evidence search with multiple questions from the same video and learning a single question policy from the resulting trajectories.

3 Methodology

In this section, we present SAVE, a framework that transfers compact evidence-search behavior from an existing long video agent to a more efficient single-question policy. As illustrated in Figure 2, SAVE consists of four components: (1) constructing a same-video context containing a *Main Question* and a *Pressure Question*; (2) generating joint tool-use trajectories in their shared video environment; (3) extracting target-centric single-question supervision by retaining Main and shared interactions while removing Pressure search and answers; and (4) refining the policy with Main-Question-Rewarded Joint GRPO using a joint Main/Pressure rollout and Main Question only verifiable rewards.

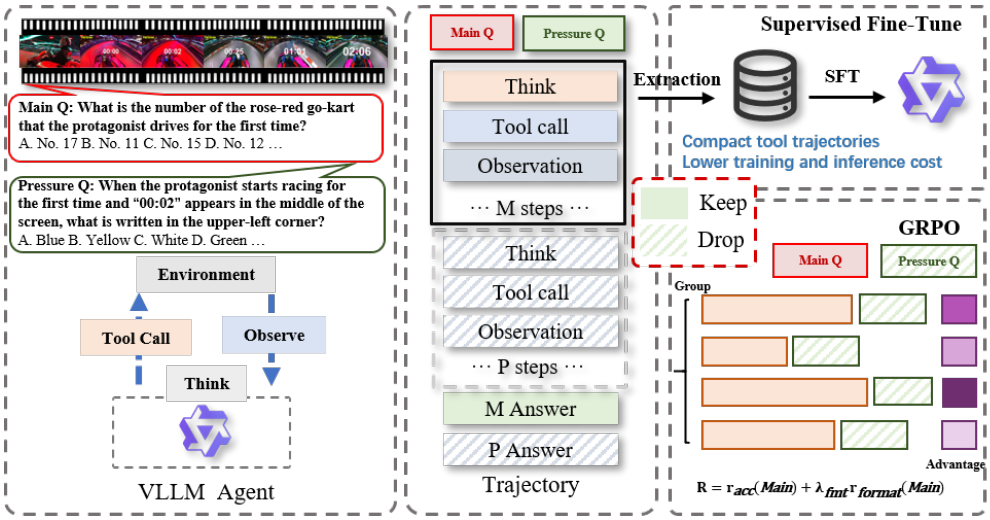


Figure 2: **SAVE pipeline.** A source video Large language model agent jointly answers a Main Question and a same-video Pressure Question under a shared tool environment and interaction budget. We retain Main-Question-specific and shared interactions, discard pressure-only content, and use the compact trajectories for SFT followed by Main-Question-Rewarded Joint GRPO, whose rollout retains both questions but whose reward uses main question only.

3.1 Problem Formulation and Notation

Let v denote a long video and q_i a question about this video. A long video agent produces an answer by generating the video-interaction trajectory

$$\tau_i = (q_i, a_1, o_1, a_2, o_2, \ldots, a_T, o_T, \hat{y}_i), \tag{1}$$

where T is the number of interaction steps, a_t is an agent action (a reasoning step or tool call), o_t is the corresponding tool observation, and $\hat{y}_i$ is the predicted final answer. Eq. 1 makes explicit that the trajectory contains both generated actions and environment observations. The inference cost of a trajectory can be measured by the number of tool calls, input tokens, output tokens, and total tokens. Our goal is to construct accurate and compact target-question evidence-search trajectories and post-train the long video agent with fewer tool calls and tokens while maintaining competitive Video QA accuracy.

3.2 Same-Video Multi-Question Pressure

Given a long video v, a target question Q_m, and its ground-truth answer y_m, we use another question associated with the same video as a pressure question Q_p. Thus, the target and pressure questions share one video evidence space,

$$\text{video}(Q_p) = \text{video}(Q_m) = v. \tag{2}$$

where $\text{video}(q)$ returns the source video associated with question q. As required by Eq. 2, we place Q_m and Q_p in a fixed order and let the source policy π_0 jointly search the same hierarchical caption tree and local video-QA environment. Let τ^{multi} denote the resulting joint interaction trajectory:

$$\tau^{\text{multi}} \sim \pi_0(\cdot \mid v, Q_m, Q_p). \tag{3}$$

Each tool call can support Q_m, Q_p, or both questions. Because the two questions are answered within one interaction trajectory, each additional search or verification step must be useful for at least one of them. Repeated inspections and overly broad retrieval add inference cost while contributing little evidence, whereas temporally informative or jointly useful observations can support subsequent decisions for both. This contextual pressure encourages the source agent to allocate evidence-search steps more selectively. SAVE introduces no explicit tool-count, token-length, or trajectory-length penalty during SFT.

We use Q_m as the deployment target and, for SFT extraction, retain a multi-question rollout only when its answer to Q_m is correct. We do not require Q_p to be answered correctly for this filtering step; its role in the joint rollout is to induce search pressure, not to introduce an additional test-time objective. After SFT-rollout generation, Q_p and its question-specific content are discarded, and the target-centric procedure below converts the remaining interactions into supervision for the original single-question task. The rollout prompt, tool protocol, and filtering rules are detailed in Supplementary Appendix A.

3.3 Target-Centric Trajectory Extraction

Raw multi-question trajectories cannot directly serve as single-question supervision: their tool calls, observations, and reasoning may support either question. Distilling them unchanged would teach behavior that is unavailable at single-question deployment, whereas prefix truncation or naive deletion can remove shared evidence required to localize or verify the answer to Q_m.

During SFT data construction, for each multi-question rollout that correctly answers the target question Q_m, we first use its record identifier to locate the corresponding single-question Baseline trajectory. This Baseline trajectory serves as the reference evidence-search path for the target question. For every tool interaction (a_t, o_t) in τ^{multi}, we assign an evidence role based on its temporal-segment match to this reference path and whether it supports Q_m, Q_p, or both questions:

$$\ell_t \in \{\mathrm{q_m}, \mathrm{shared}, \mathrm{q_p}, \mathrm{other}\}, \tag{4}$$

where the roles denote interactions that support Q_m only, both questions, Q_p only, or neither question, respectively. We keep Main-Question-specific and shared interactions and drop all remaining interactions that do not belong to the Main-Question trajectory. We then project the multi-question trajectory to a target-centric trajectory,

$$\tau_m = \mathcal{P}_{Q_m}(\tau^{\mathrm{multi}}) = \left(Q_m, [(a_t, o_t)]_{\ell_t \in \{\mathrm{q_m}, \mathrm{shared}\}}, \hat{y}_m\right). \tag{5}$$

where the bracketed interactions retain their original temporal order.

The projected trajectory retains Q_m-specific and shared actions, their corresponding observations, and the final answer to Q_m. It removes Q_p-specific search, the answer to Q_p, and residual auxiliary-question reasoning; we also restore the original single-question system and user prompts. The resulting compact trajectory is therefore compatible with the original Q_m interface while preserving shared evidence needed by the target question. We further evaluate its local logical coherence and observation–action evidence support under strictly matched Baseline–SAVE-SFT pairs; the evaluation protocol and results are provided in Supplementary Appendix A.

3.4 Single Question Policy Transfer

The target-centric trajectories define a compact single-question supervision set,

$$\mathcal{D}_{\mathrm{SAVE}} = \{(v, Q_m, \tau_m) \mid \hat{y}_m = y_m,\ \tau_m = \mathcal{P}_{Q_m}(\tau^{\mathrm{multi}})\}. \tag{6}$$

Eq. 6 retains only rollouts whose target prediction matches the ground-truth answer y_m. We fine-tune the target policy π_θ with SFT:

$$\mathcal{L}_{\mathrm{SFT}} = -\sum_{(v,q,\tau)\in\mathcal{D}_{\mathrm{SAVE}}} \sum_{j=1}^{|\tau|} m_j \log \pi_\theta(x_j \mid x_{<j}, v, q), \tag{7}$$

where x_j is the j-th token in the serialized trajectory and $m_j = 0$ for tokens belonging to system/user prompts or environment-returned observations o_t, while $m_j = 1$ for policy-generated tokens. During parameter optimization, tokens belonging to each environment-returned observation o_t are masked out and do not contribute to the training loss.

3.5 Main Question Rewarded Joint GRPO

Target-centric SFT transfers compact search behavior, but it remains an imitation objective over a selected offline corpus. We therefore further optimize the SFT policy with Main-Question-Rewarded

Joint GRPO, using the same two-question prompt and shared video-tool budget as the pressure rollout. For each target instance (v, Q_m, y_m) paired with a same-video Pressure Question Q_p, we sample a group of K joint trajectories $\{\tau_i\}_{i=1}^{K}$ from the current policy, where $\tau_i \sim \pi_\theta(\cdot \mid v, Q_m, Q_p)$, and extract the Main-Question answer $\hat{y}_{i,m}$. The prompt asks the policy to attempt both questions, but Q_p is used only to create resource pressure and is not an optimization target. Each rollout receives a verifiable reward

$$r_i = \mathbf{1}[\hat{y}_{i,m} = y_m] + \lambda_{\text{fmt}} \mathbf{1}[\text{valid}(\hat{y}_{i,m})], \tag{8}$$

where $\mathbf{1}[\cdot]$ is the indicator function, $\lambda_{\text{fmt}} \geq 0$ is the weight for valid Main-Question answer format (*fmt* denotes format), and $\text{valid}(\hat{y}_{i,m})$ indicates that the Main-Question prediction can be unambiguously parsed and evaluated. Pressure-Question correctness is neither evaluated nor rewarded. As specified in Eq. 8, the reward contains no direct tool-count or token-length term. We normalize rewards within each group of size K to form group-relative advantages A_i, and optimize the clipped GRPO objective with a Kullback–Leibler (KL) constraint to the SFT policy.

Crucially, we use no explicit tool-count or token-length reward. Such penalties can directly trade off against answer correctness and promote premature termination. Instead, the SFT initialization supplies compact search behavior, while Main-Question-Rewarded Joint GRPO re-optimizes joint Q_m/Q_p trajectories using only Main-Question task success and valid completion format. The GRPO refinement setup is provided in Supplementary Appendix D.

4 Experiments

4.1 Experimental Setup

Training Both SAVE (SFT) and (RL) use CGBench long video training instances (Chen et al., 2025). For SFT, we follow LongVideo-R1 (Qiu et al., 2026) for the chain-of-thought-with-tool trajectory format and data partition. We instantiate SAVE from the LongVideo-R1 agent. Caption retrieval is instantiated with Qwen3-VL-32B, while local video question answering is instantiated with Qwen3-VL-8B; both tool models, the video environment, and the interaction budget are fixed across methods. For SAVE (SFT), target-centric trajectories extracted from multi-question rollouts with a correct Main-Question answer are used as supervision; Q_p is not used at inference time. SAVE (RL) initializes from this SFT policy and applies 520 steps of Main-Question-Rewarded Joint GRPO using joint Q_m/Q_p rollouts under a shared budget. Only Main-Question answer correctness and format validity enter the reward; there is no explicit length penalty. Detailed data construction and training hyperparameters are provided in Supplementary Appendices A and D, respectively.

Benchmarks and Metrics We evaluate SAVE on three long video question answering benchmarks that collectively require long-range temporal grounding: (1) **LVBench** (Wang et al., 2025a), an extreme long video benchmark that probes extended temporal comprehension and information extraction; (2) **VideoMME** (Fu et al., 2025), a comprehensive video analysis benchmark spanning diverse domains and durations, for which we report both subtitle (w/ sub) and no-subtitle (w/o sub) settings; and (3) **MLVU** (Zhou et al., 2025), a multi-task long video benchmark covering diverse video genres and question types. We report MLVU results on a fixed 500-question subset of its development set, using the same subset for all compared methods. Together, these benchmarks evaluate whether an agent can preserve target-question performance while reducing tool-mediated evidence search over long video contexts.

We measure both answer quality and interaction efficiency. For answer quality, we report: (1) **Acc**: the video question answering accuracy. For interaction efficiency, we report: (1) **Avg Tool**, the average number of tool calls made by the agent per question; (2) **Avg Total Tokens**, the average total token cost, computed as the sum of input and output tokens. Together, these metrics evaluate whether a method reduces tool-mediated evidence search and context growth while preserving target-question accuracy. Additional evaluation details are provided in Supplementary Appendix E.

Baselines. We consider four baselines. (1) **Baseline** (Qiu et al., 2026): the original LongVideo-R1-Qwen3 agent with its unmodified single-question prompt, tool environment, and interaction budget. (2) **T-Pruned**: Teacher-model trajectory-pruning, where DeepSeek-V4-Flash (DeepSeek-AI, 2026) identifies the original steps necessary to support the final answer. (3) **ConPress** (Deng et al., 2026):

Table 1: Main accuracy-efficiency results on four long video QA benchmarks. Bold denotes the best SAVE result in each column; signed percentages show relative changes from the Baseline.

Method	LVBench (1,549)			VideoMME w/ sub (900)			VideoMME w/o sub (900)			MLVU (500)		
	Acc. ↑	Avg. Tool ↓	Avg. Total Tokens (k) ↓	Acc. ↑	Avg. Tool ↓	Avg. Total Tokens (k) ↓	Acc. ↑	Avg. Tool ↓	Avg. Total Tokens (k) ↓	Acc. ↑	Avg. Tool ↓	Avg. Total Tokens (k) ↓
*Planning agent: **LongVideo-R1*** *Caption tool: **Qwen3-VL-32B*** *Video-QA tool: **Qwen3-VL-8B***												
Baseline	57.0	8.0	98.4	67.4	8.6	102.0	55.9	9.9	112.3	70.0	8.2	68.8
T-Pruned	50.6 (−11.3%)	8.8 (+10.6%)	119.7 (+21.6%)	63.2 (−6.3%)	9.0 (+4.2%)	114.6 (+12.4%)	51.6 (−7.7%)	10.3 (+4.7%)	130.4 (+16.1%)	65.0 (−7.1%)	7.7 (−5.6%)	75.3 (+9.4%)
WebClipper	47.1 (−17.3%)	7.4 (−7.5%)	93.3 (−5.2%)	52.8 (−21.7%)	4.8 (−44.1%)	57.3 (−43.8%)	47.0† (−15.9%)	7.9 (−20.7%)	90.5 (−19.5%)	62.6 (−10.6%)	6.2 (−24.2%)	55.0 (−20.1%)
ConPress (N=2)	52.7 (−7.5%)	8.0 (0.0%)	99.5 (+1.1%)	63.9 (−5.3%)	7.8 (−9.5%)	86.1 (−15.5%)	55.1 (−1.4%)	9.4 (−5.0%)	103.5 (−7.8%)	67.2 (−4.0%)	7.8 (−4.1%)	66.1 (−4.0%)
SAVE (SFT)	55.8 (−2.2%)	7.6 (−5.0%)	92.5 (−6.0%)	**67.2** (−0.3%)	**7.6** (−11.9%)	83.9 (−17.7%)	**55.0** (−1.6%)	9.1 (−7.5%)	101.5 (−9.6%)	69.2 (−1.1%)	7.3 (−10.1%)	61.6 (−10.4%)
SAVE (RL)	**56.4** (−1.1%)	**7.3** (−8.8%)	**85.8** (−12.8%)	66.4 (−1.5%)	**7.6** (−12.1%)	**83.0** (−18.6%)	54.1 (−3.2%)	**8.6** (−13.0%)	**91.2** (−18.8%)	**72.6** (+3.7%)	**7.2** (−12.3%)	**56.1** (−18.5%)

multi-question compression SFT with the number of questions $N = 2$. (4) **WebClipper** (Wang et al., 2026): graph-based pruning method that removes trajectory nodes inferred to be unnecessary for connecting the query to its final answer.

Our Methods. We evaluate two versions of SAVE. **(1) SAVE (SFT)** is an SFT variant trained on trajectories with a correct target answer; **(2) SAVE (RL)** further applies the 520-step Main-Question-Rewarded Joint GRPO refinement reported in the main table. All methods use the same caption tool, local video-QA tool, and inference-time interaction budget. Detailed pruning-baseline construction and training configurations are provided in Supplementary Appendices C and D.

4.2 Main Result

Table 1 reports accuracy and interaction cost across four long video QA benchmarks. We evaluate SAVE (SFT), trained with target-centric supervision, and SAVE (RL), which further applies Main-Question-Rewarded Joint GRPO on joint Q_m/Q_p rollouts under a shared budget, with reward computed from the Main Question only. SAVE (SFT) consistently reduces tool calls by 5.0–11.9% and total tokens by 6.0–17.7% across all benchmarks, with only a 0.2–1.2 percentage-point accuracy decrease. The largest token reduction occurs on VideoMME with subtitles, where subtitle cues support coarse localization and reduce repeated local inspection. Without subtitles, the agent must rely more heavily on visual exploration; SAVE still reduces interaction cost, but the accuracy–efficiency trade-off becomes more pronounced.

SAVE occupies a different point on the accuracy–efficiency frontier from post-hoc pruning, as visualized in Figure 3(a). WebClipper aggressively reduces tool use and tokens, particularly on VideoMME, but its accuracy decreases by 7.4–14.6 % across the completed evaluations; it also produces 44 empty predictions on VideoMME without subtitles. T-Pruned similarly incurs 4.2–6.4 percentage-point accuracy drops and does not consistently reduce total tokens. In contrast, SAVE (SFT) shifts the Baseline toward lower token cost with a modest accuracy change, while SAVE (RL) further improves the aggregate trade-off. Compared with direct ConPress, SAVE (SFT) improves accuracy while using fewer tools and tokens on LVBench, VideoMME with subtitles, and MLVU, and matches its accuracy with lower cost on VideoMME without subtitles. These comparisons support target-centric extraction as a more suitable way to transfer joint multi-question behavior to single-question deployment.

Starting from SAVE (SFT), SAVE (RL) further reduces tool calls and total tokens on all four benchmarks. On MLVU-Dev500, it improves accuracy from 70.0% to 72.6% while reducing total tokens by 18.5%; on the other settings, it trades modest accuracy decreases for larger interaction-cost reductions. Together, the results show that same-video multi-question pressure provides a compact-search initialization that can be further refined with verifiable-reward policy optimization.

4.3 Component Ablation

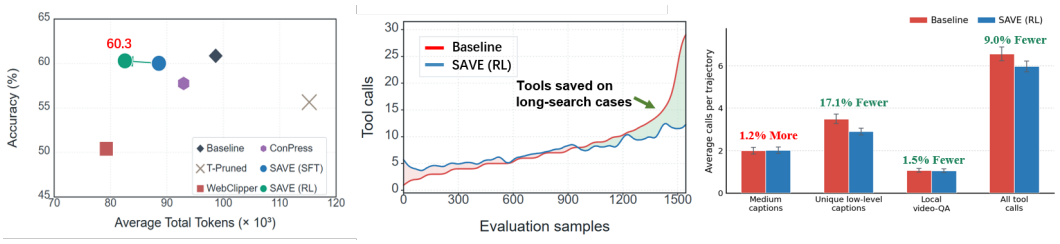


Figure 3: **Accuracy–efficiency and search behavior.** (a) Accuracy versus average total tokens across methods. (b) Gaussian-smoothed LVBench tool-use trends after sorting samples by the Baseline count. (c) Tool-use decomposition on jointly correct LVBench trajectories. Error bars denote 95% paired bootstrap confidence intervals.

Table 2 isolates the three components of SAVE: same-video pressure, target-centric extraction, and GRPO refinement. The rows correspond, from top to bottom, to CrossVideo-MainFirst, ConPress, SAVE (SFT), and SAVE (RL). We report sample-size-weighted accuracy, tool calls, and total tokens over all 3,849 evaluated examples. CrossVideo-MainFirst is a negative control in which the pressure question comes from another video and its visual context is unavailable; it tests whether an additional question alone can replace same-video shared evidence. Its full protocol and per-benchmark results are provided in Supplementary Appendix B.

Table 2: **Component ablation.** Metrics are sample-size-weighted averages over all evaluated examples from the four benchmarks.

Same-Video Pressure	Target extraction	GRPO	Acc.	Avg. Tool	Total Tok. (k)
–	–	–	58.72	8.56	95.11
√	–	–	57.76	8.25	92.96
√	√	–	60.02	7.91	88.58
√	√	√	**60.30**	**7.66**	**82.55**

Same-video evidence space. We first test whether the pressure question must share the target video's evidence space. Compared with SAVE (SFT), CrossVideo-MainFirst decreases aggregate accuracy by 1.30 %, while requiring 0.65 more tool calls and 6.53K more total tokens per example. This negative control shows that an additional question without shared video evidence does not induce transferable compact search.

Target-centric extraction. Target-centric extraction improves over ConPress by 2.26 % in aggregate accuracy, while reducing tool use by 4.1% and total tokens by 4.7%. The result shows that multi-question behavior is not directly deployable: retaining Main-Question-specific and shared evidence while removing pressure-only interactions produces a stronger single-question policy.

Verifiable refinement. Starting from SAVE (SFT), GRPO raises aggregate accuracy by a further 0.28 %, while reducing tool calls by 3.2% and total tokens by 6.8%. Thus, verifiable reinforcement learning improves the quality–efficiency trade-off from the compact SFT initialization without an explicit length reward.

4.4 Analysis

Supervision Construction Efficiency Optimizing agent trajectories typically requires multiple rollouts, which can introduce substantial additional computational overhead. We further compare the cost of constructing SFT datasets with SAVE, T-Pruned, and WebClipper. As shown in Table 3, SAVE directly constructs compact supervision from the source agent's same-video multi-question rollouts and requires no external strong-model application programming interface (API). In contrast, T-Pruned and WebClipper rely on external teachers to identify or rewrite redundant steps; WebClipper in particular incurs substantial

Table 3: External-teacher cost for trajectory construction. Detailed API pricing and cost calculations are provided in Supplementary Appendix F.

Method	API Calls	Teacher Tok.	Cost	Usable SFT
SAVE	0	0	$0	2,097
T-Pruned	2,236	5.03M	$1.08	2,184
WebClipper	36,638	410.13M	$507.64	462

teacher-call and token overhead for state-graph construction and reasoning rewriting. This comparison shows that SAVE performs compression during the source agent's own trajectory generation, avoiding the additional data-construction cost of post-hoc teacher pruning. Detailed API pricing and cost calculations are provided in Supplementary Appendix F.

Long-search control. To examine how SAVE saves tool calls, we sort LVBench examples by the Baseline tool-call count and compare the paired tool-use trends of the Baseline and SAVE (RL) for the same questions, as shown in Figure 3(b). The curves are smoothed to characterize the overall change across different initial search lengths. SAVE uses a similar, or occasionally slightly larger, number of calls on short searches, and the two policies also nearly coincide on medium-length searches. The difference is concentrated in the long-search tail: as the Baseline continues from roughly 10 calls toward 30 calls, SAVE remains around 10–12 calls. In other words, SAVE mainly suppresses exploration that would otherwise continue expanding, rather than lowering average cost by uniformly shortening evidence acquisition for every example. Therefore, multi-question pressure does not directly penalize trajectory length; instead, it encourages the model to reduce additional probes with low marginal value during shared evidence search.

Selective tool use. To identify how each tool contributes to the overall savings, Figure 3(c) decomposes average calls by tool type over LVBench trajectories answered correctly by both the Baseline and SAVE. SAVE yields its largest reduction in low-level caption requests, decreasing them by 17.1%, while medium-level caption retrieval and video-QA use remain nearly unchanged. Medium-level captions primarily narrow candidate temporal regions, whereas video-QA is mainly used for local visual verification. This pattern therefore indicates that SAVE does not simply skip localization or verification; it reduces repeated fine-grained caption probing within candidate regions that have already been explored. The behavior is consistent with target-centric trajectory extraction: the training data retain interactions needed for the Main Question and those shared by the two questions, while removing searches that only serve the Pressure Question. Consequently, SAVE learns to selectively reduce redundant probing rather than post-hoc prune complete trajectories.

Case study. Figure 4 compares paired trajectories for a pose-order question. Both policies answer correctly, but SAVE uses four caption calls to establish the temporal order of the four relevant poses. The Baseline uses ten calls, including repeated fine-grained and broad probes around regions it has already inspected. This example illustrates that SAVE can avoid redundant local exploration while retaining the temporal evidence required to answer the question. Additional examples and complete trajectories are provided in Appendix G.

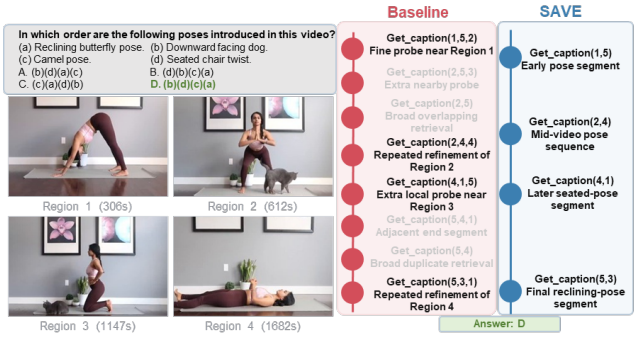


Figure 4: Case study between baseline and SAVE.

5 Conclusion

In this paper, we introduce SAVE, a self-compression framework for long video understanding agents. SAVE uses the pressure arising from multiple questions about the same video, which share a tool environment, interaction budget, and temporal evidence space, to encourage the source agent to suppress low-value repeated searches. Target-centric trajectory extraction transforms joint rollouts into deployable single-question supervision by retaining Main-Question-specific and shared interactions. Starting from this compact SFT policy, SAVE further applies Main-Question-Rewarded Joint GRPO: the rollout retains Q_m and Q_p under the shared budget, while the reward uses Main-Question answer correctness and format validity only. Experiments across long video benchmarks show that SAVE reduces tool calls and token consumption while retaining competitive video-QA performance.

AI USE DISCLOSURE

In this work, we used generative AI tools for the following tasks requiring disclosure: generating synthetic tool-use trajectories from benchmark instances with open-source LLM agents; helping develop the conceptual framework; proposing and refining research hypotheses; designing and providing feedback on the research methodology and experiments; implementing the methods; assisting with translation; cleaning and reformatting datasets; supporting qualitative analysis; and interpreting experimental results. We did not use generative AI tools to formulate mathematical claims, provide critical ingredients for mathematical proofs, or write proofs; the remaining required-disclosure tasks are not applicable to this work.

Additionally, we used generative AI tools to suggest experimental parameters, create and edit software code, draft parts of the manuscript, partially search for research materials, improve the readability of the manuscript, identify relevant literature, and format references. We did not use generative AI tools to create or modify scientific figures or images; the remaining recommended-disclosure tasks are not applicable to this work.

We reviewed all AI-assisted work. In particular, generated trajectories, code, analyses, reported statistics, and references were checked against the corresponding data, procedures, and source materials. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Guo Chen, Yicheng Liu, Yifei Huang, Baoqi Pei, Jilan Xu, Yuping He, Tong Lu, Yali Wang, and Limin Wang. CG-Bench: Clue-grounded question answering benchmark for long video understanding. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=le4IoZZHy1.

Zhoujun Cheng, Jungo Kasai, and Tao Yu. Batch prompting: Efficient inference with large language model APIs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 792–810, 2023. doi: 10.18653/v1/2023.emnlp-industry.74. URL https://aclanthology.org/2023.emnlp-industry.74/.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Jie Deng, Shining Liang, Jun Li, Hongzhi Li, and Yutao Xie. Conpress: Learning efficient reasoning from multi-question contextual pressure. In *Proceedings of the International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.01472.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24108–24118, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Fu_Video-MME_The_First-Ever_Comprehensive_Evaluation_Benchmark_of_Multi-modal_LLMs_in_CVPR_2025_paper.html.

Minki Kang, Wei-Ning Chen, Dongge Han, Huseyin Inan, Lukas Wutschitz, Yanzhi Chen, Robert Sim, and Saravan Rajmohan. ACON: Optimizing context compression for long-horizon LLM agents. In *Proceedings of the International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.00615.

Zhenyu Li, Zuchao Li, Ping Wang, Lefei Zhang, and Haojun Ai. Vista-LLM: Decoupled query-guided visual token pruning for efficient long-video large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 13171–13187, 2026. URL https://aclanthology.org/2026.acl-long.601/.

Zhuoshi Pan, Qizhi Pei, Yu Li, Qiyao Sun, Zinan Tang, H. Vicky Zhao, Conghui He, and Lijun Wu. REST: Stress testing large reasoning models by asking multiple problems at once. In *Proceedings*

of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2026. URL https://aclanthology.org/2026.acl-long.1296/.

Qizhi Pei, Lijun Wu, Zhuoshi Pan, Yu Li, Honglin Lin, Chenlin Ming, Xin Gao, Conghui He, and Rui Yan. MathFusion: Enhancing mathematic problem-solving of LLM through instruction fusion, 2025. URL https://arxiv.org/abs/2503.16212.

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*, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Qiu_LongVideo-R1_Smart_Navigation_for_Low-cost_Long_Video_Understanding_CVPR_2026_paper.html.

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 *Proceedings of the International Conference on Machine Learning*, pp. 54582–54599, 2025. URL https://proceedings.mlr.press/v267/shen25j.html.

Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-XL: Extra-long vision language model for hour-scale video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Shu_Video-XL_Extra-Long_Vision_Language_Model_for_Hour-Scale_Video_Understanding_CVPR_2025_paper.html.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18221–18232, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Song_MovieChat_From_Dense_Token_to_Sparse_Memory_for_Long_Video_CVPR_2024_paper.html.

Junjie Wang, Zequn Xie, Dan Yang, Jie Feng, Yue Shen, Duolin Sun, Meixiu Long, Yihan Jiao, Zhehao Tan, Jian Wang, Peng Wei, and Jinjie Gu. Webclipper: Efficient evolution of web agents with graph-based trajectory pruning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21646–21666, 2026. doi: 10.18653/v1/2026.acl-long.988. URL https://aclanthology.org/2026.acl-long.988/.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22958–22967, 2025a. URL https://openaccess.thecvf.com/content/ICCV2025/html/Wang_LVBench_An_Extreme_Long_Video_Understanding_Benchmark_ICCV_2025_paper.html.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *Proceedings of the European Conference on Computer Vision*, 2024. URL https://arxiv.org/abs/2403.10517.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025b. URL https://arxiv.org/abs/2405.19209.

Bangji Yang, Hongbo Ma, Jiajun Fan, and Ge Liu. Batched contextual reinforcement: A task-scaling law for efficient reasoning. In *Proceedings of the International Conference on Machine Learning*, 2026a. URL https://arxiv.org/abs/2604.02322.

Zeyuan Yang, Delin Chen, Xueyang Yu, Maohao Shen, and Chuang Gan. VCA: Video curious agent for long video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20168–20179, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Yang_VCA_Video_Curious_Agent_for_Long_Video_Understanding_ICCV_2025_paper.html.

Zuhao Yang, Sudong Wang, Kaichen Zhang, Keming Wu, Sicong Leng, Yifan Zhang, Bo Li, Chengwei Qin, Shijian Lu, Xingxuan Li, and Lidong Bing. Longvt: Incentivizing thinking with long videos via native tool calling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026b. URL https://arxiv.org/abs/2511.20785.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, Pengjun Xie, Fei Huang, Siheng Chen, Jingren Zhou, and Yong Jiang. Agentfold: Long-horizon web agents with proactive context management. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.24699.

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 22128–22136, 2025. URL https://ojs.aaai.org/index.php/AAAI/article/view/34366.

Xiaoyi Zhang, Zhaoyang Jia, Zongyu Guo, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Deep video discovery: Agentic search with tool use for long-form video understanding. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/8190b210e9808e54ee16263b673a847d-Abstract-Conference.html.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. MLVU: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13691–13701, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zhou_MLVU_Benchmarking_Multi-task_Long_Video_Understanding_CVPR_2025_paper.html.

Table 4: Logical quality and length of strictly matched Baseline SAVE-SFT trajectories. Logical quality is scored independently by DeepSeek-V4-Flash on a 0–5 scale.

Metric	Baseline	SAVE-SFT	Change
Mean logical quality (/5)	3.37	3.14	-0.23
Mean characters	33,990.0	25,984.7	-23.55%
Mean words	5,478.0	4,201.9	-23.29%
Mean non-system messages	14.04	8.68	-38.18%
Mean assistant turns	7.02	4.34	-38.18%

A Training Data Construction

Models and Tool Environment

We instantiate SAVE from the LongVideo-R1 planning agent. We use Qwen3-VL-32B for caption retrieval and Qwen3-VL-8B for local video question answering. All compared methods use the same video environment, tool interfaces, decoding configuration, and interaction budget; only the trajectory construction and selection rule vary.

Source Instances and Rollouts

SAVE supervision is constructed from CGBench (Chen et al., 2025) long-video tool-use instances, following the CoTwT trajectory format and data partition released by LongVideo-R1 (Qiu et al., 2026). For every Main Question (Q_m), we select a distinct Pressure Question (Q_p) from the same video and construct a two-question rollout under a shared interaction budget. The two questions share the same hierarchical caption and local video-QA environment, allowing a tool interaction to serve Q_m, Q_p, or both questions.

Prompt structure. The planning agent prompt is the original LongVideo-R1 prompt and tool specification. The user message places two distinct questions from the same video in a common video context, in the order *Main Question (Q_m):* target question and *Pressure Question (Q_p):* auxiliary question, followed by the original hierarchical caption tree for that video. The agent is asked to answer both questions using the available `get_caption` and `video_qa` tools. The auxiliary question is used only during rollout construction; it is not included in the SFT input or at test time.

Tool protocol. The caption environment provides a hierarchical temporal index over the video. The agent first uses `get_caption` to inspect candidate coarse or low-level temporal regions and can subsequently invoke `video_qa` on an inspected low-level region for local visual verification. Each tool call must be followed by its returned observation. We discard records with incomplete call–observation pairs or invalid tool order, such as applying `video_qa` before visiting the associated low-level caption.

Generated Trajectory Evaluation To evaluate whether trajectory extraction preserves logical coherence and evidence support, we sampled 100 question IDs from 1,750 strictly matched Baseline and SAVE-SFT pairs. Each trajectory was independently evaluated by DeepSeek-V4-Flash for the local logic between observations and subsequent actions, such as whether an observation o supports the following action a, using a 0–5 scale.

The evaluation prompt asked the model to assess whether the trajectory remains logically coherent and whether each action is adequately supported by the preceding observations, without judging the final answer against external knowledge. As shown in Table 4, SAVE-SFT substantially shortens the trajectories while reducing the mean logical-quality score by only 0.23 points relative to the Baseline. The similar scores suggest that the extracted trajectories preserve comparable local coherence and evidence support while substantially reducing the training-example length.

Raw two-question rollout accuracy. The complete CGBench rollout set used for SAVE-SFT construction contains 4,183 paired trajectories. As shown in Table 5, the single-question baseline

Table 5: Performance of the baseline on different question types in the CGBench rollout set.

Question	Accuracy	Tool Used
Single Question	57.21%	7.62
Main Question (Q_m)	52.88%	4.26
Pressure Question (Q_p)	34.21%	2.03

Table 6: CrossVideo-MainFirst evaluation.

Dataset	N	Acc.	Avg. Tool	Avg. Input Tok.	Avg. Output Tok.	Avg. Total Tok.
LVBench	1,549	54.94%	8.22	97,788.6	2,604.6	100,393.2
Video-MME w/ sub	900	64.89%	8.39	90,317.7	2,455.3	92,772.9
Video-MME w/o sub	900	52.78%	9.73	103,547.7	2,811.5	106,359.3
MLVU-Dev500	500	70.00%	7.82	60,394.7	2,320.6	62,715.2

had already been trained on CGBench. In the multi-question setting, the Main Question retained most usable trajectories while requiring fewer tool calls. In contrast, the Pressure Question suffered a substantial accuracy drop. We therefore did not include its trajectory as training supervision; instead, it served as a downstream source of pressure in the multi-question rollout.

B Cross-Video Text-Pressure Control

We include CrossVideo-MainFirst as a negative control rather than a SAVE variant. It tests whether merely displaying an additional question can create useful compression when the question does not share video evidence with the Main Question. For every Main Question Q_m from Video A, we sample a Pressure Question Q_p from Video B with $\text{video}(Q_m) \neq \text{video}(Q_p)$. The model sees only the text of Q_p: it receives no Video-B captions, tools, observations, or answer and can interact only with Video A.

Cross-video prompt and rollout. We append the following sequential constraint to the original LongVideo-R1 system prompt:

```
You must solve two questions about two different videos
in order. Only the context for Video A is provided now.
Solve the Main Question using only the current video
context and its tools. Do not search for, discuss, or
answer the Pressure Question yet. When the Main Question
is sufficiently verified, output exactly <answer>X</answer>
for the Main Question and stop immediately. The context for
Video B will be provided only after the Main Question is
complete.
```

Cross-video filtering and SFT. We retain a rollout only when the Main Question is correct, every tool call has a subsequent observation, the final assistant turn has an explicit Main-Question answer, the agent neither discusses nor answers the Pressure Question, and the tool protocol is valid. We then deterministically restore the original single-Main-Question prompt and retain the complete Main-Question trajectory; Pressure-Question text and all pressure instructions are removed. This control has 2,322 accepted raw trajectories and 2,321 valid SFT examples, with 5.24 tool calls per generation trajectory and 106.86M five-epoch SFT token exposure.

Cross-video text pressure does not yield a stable accuracy-efficiency trade-off: it is worse than the single-question baseline on both accuracy and cost on LVBench and loses accuracy on both Video-MME settings despite modest cost reductions. This result indicates that the benefit of SAVE is not induced by the presence of an extra question alone; same-video questions enable shared temporal evidence and jointly useful tool interactions, from which a coherent target-centric trajectory can be extracted.

C TRAJECTORY-PRUNING BASELINES

T-PRUNED

T-Pruned uses DeepSeek-V4-Flash (DeepSeek-AI, 2026) as a trajectory-pruning teacher. We retain single-question CGBench trajectories generated by the Baseline only when the terminal answer is correct and the trajectory contains at least three tool calls. For each retained trajectory, the teacher identifies tool interactions and associated reasoning steps necessary to support the final answer; the selected actions and their observations are reconstructed as a single-question SFT record. Unlike SAVE, T-Pruned directly delegates step necessity to an external strong model and does not use same-video multi-question rollouts or target-centric shared-evidence extraction.

WEBCLIPPER

We adapt WebClipper (Wang et al., 2026), a graph-based trajectory-pruning method originally developed for web agents, as a single-question long-video baseline. We start from 4,183 single-question rollouts produced by LongVideo-R1-Qwen3 on the CGBench training questions and retain the 2,393 trajectories with a correct terminal answer. For each retained trajectory, we use `gpt-5.6-luna` to abstract agent turns into action nodes and independently construct three state graphs over adjacent actions. We mine an approximate minimum-necessary directed acyclic graph (MNDAG) from each construction and retain majority-voted nodes required to connect the initial query to the final answer, removing the remaining search steps.

Pruning can leave adjacent reasoning text incoherent after intermediate nodes are removed. For every such gap, `gpt-5.6-luna` generates three rewritten candidates. We score these candidates with the frozen LongVideo-R1-Qwen3 model and retain the lowest-perplexity version, yielding a compressed single-question trajectory that can be used directly for SFT. This pipeline successfully produces 462 compressed trajectories; the remaining examples are skipped because MNDAG mining does not return a valid set of necessary nodes.

To make the optimization exposure comparable despite the substantially smaller WebClipper corpus, we train this baseline for 20 epochs rather than the five epochs used by SAVE SFT, bringing the total training-token exposure to the same order of magnitude. All remaining SFT hyperparameters, including the base model, template, optimizer, sequence length, packing, and distributed-training configuration, are identical to those in Table 7.

D TRAINING CONFIGURATION

We fine-tune the LongVideo-R1 planning agent on the selected target-centric trajectories with standard supervised fine-tuning. Table 7 lists the optimization and systems configuration.

Training Efficiency. The cost of SFT long video agents grows rapidly with tool trajectories and their accumulated observations. We therefore examine whether SAVE can compress training trajectories while retaining effective policy learning. Under a matched SFT configuration, we compare SAVE with two multi-question datasets: ConPress, which retains interactions serving Q_p in the joint trajectory, and CrossVideo, which uses a Pressure Question from a different video without shared evidence. Table 8 shows that SAVE achieves lower cumulative training-token exposure while maintaining a per-example trajectory length comparable to ConPress. Together with the competitive accuracy–efficiency trade-off in the main results, this indicates that the compression preserves useful supervision rather than degrading training effectiveness. Crucially, SAVE does not truncate complete evidence chains; it removes Q_p-only interactions that cannot be reused at single-question deployment while retaining target-specific and shared evidence. Dataset-construction details and matched training settings are provided in Appendices A and D.

GRPO REFINEMENT

SAVE (RL) initializes from the SAVE SFT checkpoint and performs 520 steps of Main-Question-Rewarded Joint GRPO. Each rollout retains the Main Question and same-video Pressure Question in the prompt and uses their shared video-tool interaction budget; the prompt asks the policy to

Table 7: Training configuration for SAVE.

Setting	Value
Base model	LongVideo-R1-Qwen3
Template	`qwen3_thinking`
Objective	Full SFT
Hardware	8 GPUs
Precision and kernels	BF16, FlashAttention-2, Liger
Distributed training	DeepSpeed ZeRO-2
Epochs	5
Learning rate	2×10^{-5}
Warmup ratio	0.05
Per-device batch size	1
Gradient accumulation	8
Global batch size	64
Maximum sequence length	32,768
Packing	Enabled
Train/validation split	99% / 1%
Dataloader / dataset workers	4 / 16

Table 8: **Training-token footprint of SFT supervision.** Cumulative token exposure is reported for each SFT run. CrossVideo-MainFirst is a negative control, not a SAVE variant.

Method	SFT Examples	Tok. / Epoch	Tok. / Example	Total Training Tokens
ConPress	2,878	22.67M	7,878	113.36M
CrossVideo	2,321	21.37M	9,208	106.86M
SAVE	2,097	16.60M	7,915	82.99M

attempt both questions. The reward consists only of verifiable Main-Question answer correctness and a valid Main-Question final-answer-format indicator. Pressure-Question correctness is neither evaluated nor rewarded, and we use no explicit reward term for tool calls, input tokens, output tokens, or trajectory length. Thus, any additional inference efficiency after RL is attributable to the compact behavior initialized by target-centric SFT and its compatibility with Main-Question answer-success optimization, rather than to direct length penalization.

Pressure-Question (Q_p) Activity During RL. Although Pressure-Question correctness is not included in the GRPO reward, Q_p remains in every joint rollout prompt and competes with Q_m for the shared video-tool interaction budget. Figure 5 reports the Q_p diagnostic accuracy reconstructed from one logged LVU diagnostic entry per optimization step. Its 25-step moving average remains positive throughout training and repeatedly recovers after local fluctuations, rather than exhibiting a monotonic downward trajectory. Thus, Q_p does not exhibit monotonic collapse or complete neglect during RL; it remains behaviorally active in the joint rollout.

This diagnostic is not a Pressure-Question optimization result: Q_p is neither evaluated nor rewarded by the GRPO objective. In addition, the current logs do not serialize tool ownership as Main-Question-specific, Pressure-Question-specific, or shared. Therefore, the figure supports continued Q_p activity, but does not quantify the interaction budget consumed by Q_p or shared evidence.

E Evaluation Protocol

All methods use the same Qwen3-VL-32B caption tool, Qwen3-VL-8B local video-QA tool, video environment, and inference-time interaction budget. We report answer accuracy, average number of tool calls, average input tokens, average output tokens, and average total tokens. LVBench is evaluated on all 1,549 instances; Video-MME is evaluated on 900 long-video instances for each subtitle condition; and MLVU is evaluated on the same fixed 500-question subset of its development set for every method. To make the MLVU subset reproducible, we release the exact manifest of its 500 development question IDs with the evaluation code; this list is used unchanged for every

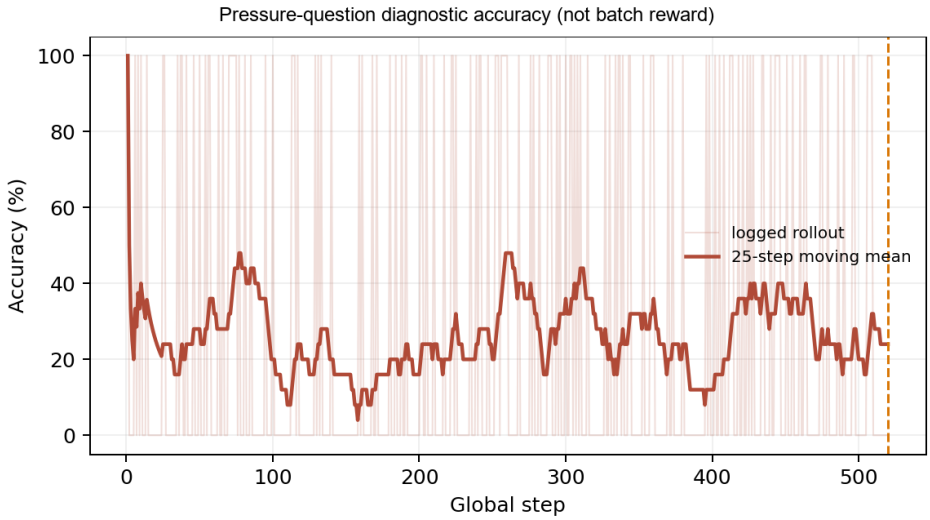


Figure 5: **Pressure-Question (Q_p) activity during Main-Question-Rewarded Joint GRPO.** Q_p diagnostic accuracy is reconstructed from one logged LVU diagnostic entry per optimization step; the dark curve is a 25-step moving average. Although Q_p is not rewarded, its diagnostic accuracy repeatedly recovers throughout training, indicating no monotonic collapse or complete neglect of Q_p during joint rollouts.

compared method. For every benchmark, a tool call and its returned observation are counted as part of the interaction trace, and token statistics are aggregated per evaluated question.

F External-Teacher Cost Details

The external-teacher cost table in the main paper reports only external-teacher overhead; it does not include source-agent rollout computation. SAVE requires no external teacher calls. For T-Pruned, we report the observed billing for 2,236 DeepSeek-V4-Flash requests: 5.03M teacher tokens and CNY 7.29. At the time of accounting (July 26, 2026), DeepSeek listed CNY 0.02/M cached-input tokens, CNY 1.00/M cache-miss input tokens, and CNY 2.00/M output tokens for DeepSeek-V4-Flash. [1] Because the billing export records the aggregate token count and realized charge but not the cache-hit and input/output token breakdown, we report the realized charge rather than reconstructing it from these rates. The USD value in the main table is reported only to facilitate comparison with the WebClipper API estimate.

For WebClipper, we use the published GPT-5.6 Luna API rates of \$1 per million input tokens and \$6 per million output tokens.[2] State-graph construction uses 34,655 calls, 369.18M input tokens, and 19.02M output tokens; DAG/message rewriting uses 1,983 calls, 21.45M input tokens, and 0.49M output tokens. Thus, the estimated external-teacher cost is

$$390.624 \times \$1 + 19.503 \times \$6 = \$507.64, \tag{9}$$

where token volumes are measured in millions. This accounting yields 36,638 calls and 410.13M teacher tokens in total.

G Qualitative Case Studies

We provide one jointly correct Baseline–SAVE pair from each evaluation set. These examples use moderate Baseline trajectories of 8–10 calls and SAVE trajectories of 3–4 calls. Each 20-frame grid is aligned with the temporal windows visited by either policy. The tables reproduce the complete,

[1] https://api-docs.deepseek.com/zh-cn/quick_start/pricing/, accessed July 26, 2026.
[2] https://developers.openai.com/api/docs/models/gpt-5.6-luna, accessed July 26, 2026.

unabridged model-generated *Think* text before every tool call. To keep the tables readable, each *Tool response* contains the leading sentence of the returned observation; full observations are retained in the accompanying raw case traces.

LVBench Case: Animated-Video Classification

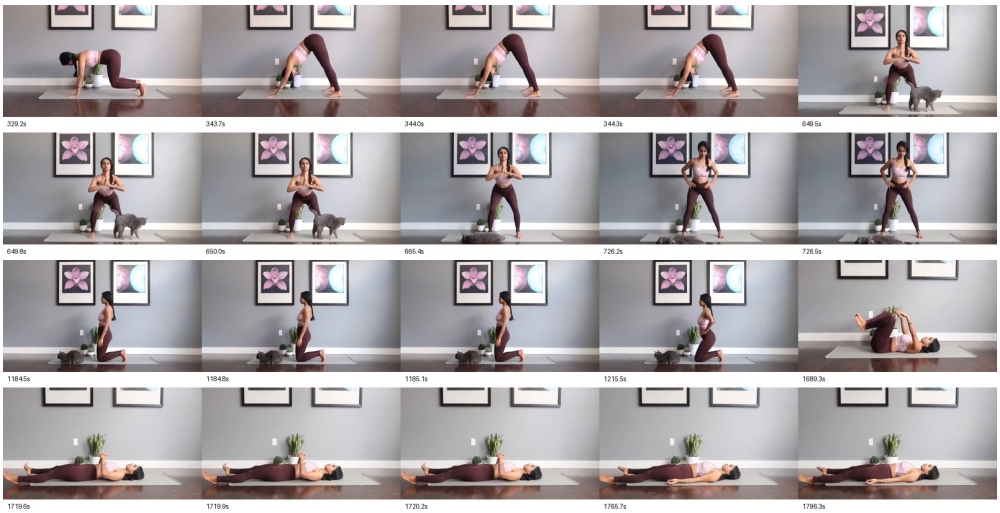


Query: What is the type of the video? (A) Cartoon (B) Vlog (C) Football live (D) Film	
Baseline	**SAVE**
Step 1: Think. The question asks for the type of the video. The high-level captions describe animated scenes with talking animals, magical elements, and fantasy settings, but they don't explicitly state the video type. To determine the type, I need to see if the video is a cartoon, vlog, football live, or film. The first high-level caption mentions "MCA HOME VIDEO" and animated characters, but it's not clear if it's a cartoon or some other type. To confirm, I should look at finer details in the first segment, as that's where the introduction and style would be shown. I'll get the medium-level caption for High-level segment 1 to see if it describes animation or live-action.	**Step 1: Think.** The question asks about the type of the video. The high-level captions describe animated scenes with characters, creatures, and fairies, but they don't explicitly state the video type. To determine the type, I need to see if the video is a cartoon, vlog, football live, or film. The high-level captions mention animated elements (characters, creatures, magical scenes) and no real-world footage or sports, so it's likely a cartoon. However, to be certain, I should check finer captions from the beginning to confirm the video's nature. The first high-level caption mentions "MCA HOME VIDEO" logo and animated scenes, but to be sure, I should look at the medium or low-level captions of the first segment to see if it's a cartoon or something else. Let's get the medium-level caption for High-level 1 to see more details about the video type.
Step 1: Tool. `get_caption ((1,1))`. **Tool response (excerpt).** This clip (video) begins at 0.0s with a dark brown screen, which transitions at 1.6s to a starry background with small white dots resembling stars.	**Step 1: Tool.** `get_caption ((1,1))`. **Tool response (excerpt).** This clip (video) begins at 0.0s with a dark brown screen, which transitions at 1.6s to a starry background with small white dots resembling stars.
Step 2: Think. The medium-level caption for the first part of the video shows animated sequences with magical elements and talking characters, but it also includes scenes with a boy and girl interacting in a realistic environment. This could suggest a mix of animation and live-action, but the question is about the type of the video. To determine this, I need to check if the animation style is consistent throughout or if it's a live-action video with some animation effects. The first high-level caption mentions animated characters and magical scenes, but to be sure, I should look at the low-level captions for the beginning of the video to see if it's clearly an animated cartoon. I'll get the low-level caption for (1,1,1) to see the detailed description of the opening animation.	**Step 2: Think.** The medium-level caption for High-level 1, segment 1 shows animated sequences with magical autumn settings, fairy-like creatures called Blinkins, and whimsical scenes. This strongly suggests the video is an animated cartoon. However, to be certain, I should check another medium-level caption from High-level 1 to see if it continues to show animated content rather than real-life footage. Let's get the medium-level caption for High-level 1, segment 2 to confirm the video type.
Step 2: Tool. `get_caption ((1,1,1))`. **Tool response (excerpt).** This clip (video) begins at 0.0s with a dark, nearly black screen, which gradually transitions into a starry, deep-space background by 0.7s.	**Step 2: Tool.** `get_caption ((1,2))`. **Tool response (excerpt).** This clip (video) begins at 103.5s with a scene showing a woman with red hair holding a baby in a carrier, standing beside a blue and white "STOP" sign with a bus symbol, near a rocky cliff.
Step 3: Think. The low-level caption for the opening shows an animated, magical sequence with a whimsical tone, but it doesn't explicitly state whether it's a cartoon or a film. To determine the type, I need to see if the rest of the video continues with consistent animation or if it's live-action. The high-level caption mentions animated characters and magical scenes, but to be certain, I should check another low-level caption from the same high-level segment to see if animation is consistent. I'll get the low-level caption for (1,1,2) to see the continuation of the introduction and whether characters are animated or live-action.	**Step 3: Think.** The medium-level caption for High-level 1, segment 2 includes both real-life and magical elements. It starts with a woman and a baby, then transitions to magical transformations and animated fairies, owls, and creatures. This mix suggests it's an animated fantasy series rather than a vlog or live-action film. However, to be sure, I should check if the video is a cartoon or something else. The presence of magical creatures, animated sequences, and fantasy settings points to a cartoon. To confirm, I can check another medium-level caption from High-level 1, perhaps segment 3, to see if it continues with animated content. Let's get the medium-level caption for High-level 1, segment 3.

LVBench Case: Animated-Video Classification (continued)	
Baseline	**SAVE**
Step 3: Tool. `get_caption ((1,1,2)).` **Tool response (excerpt).** This clip (video) begins at 20.7s with a close-up of a rocky, vine-covered landscape adorned with purple grapes and golden leaves, suggesting an autumnal forest setting.	**Step 3: Tool.** `get_caption ((1,3)).` **Tool response (excerpt).** This clip (video) begins at 206.9s with a view of a large, round, beige creature with a small yellow figure standing before it in a dark, purple cave-like setting.
Step 4: Think. The low-level captions so far show animated, magical scenes with talking or anthropomorphic creatures, but to confirm the video type, I should check if the animation style is consistent throughout the introduction. The first two low-level captions are clearly animated, but to be sure, I should check another part of the same high-level segment to see if the animation continues or if there's a mix with live-action. The high-level caption mentions both animated sequences and scenes with a boy and girl, which could be live-action. To verify, I'll get the low-level caption for (1,1,3) to see if the animation style changes or remains consistent.	**Step 4: Think.** The medium-level caption for High-level 1, segment 3 continues to show animated characters and fantastical settings: a wolf-like creature, an owl-like scientist, and magical creatures in a laboratory and garden. This is clearly part of an animated series rather than real-life footage or a film. The recurring animated characters and fantasy elements confirm it's a cartoon. However, to be absolutely certain, I can check the low-level caption for one of these segments to see if it explicitly mentions animation or stylized graphics. Let's get the low-level caption for High-level 1, segment 1, low-level 1 to see if it describes animation.
Step 4: Tool. `get_caption ((1,1,3)).` **Tool response (excerpt).** This clip (video) begins at 41.4s with a vibrant, animated scene of colorful, flower-like creatures floating upward against a bright blue sky with fluffy white clouds.	**Step 4: Tool.** `get_caption ((1,1,1)).` **Tool response (excerpt).** This clip (video) begins at 0.0s with a dark, nearly black screen, which gradually transitions into a starry, deep-space background by 0.7s.
Step 5: Think. The low-level captions for the beginning of the video are clearly animated, with magical and whimsical elements, but the question is about the overall type of the video. The high-level captions mention multiple animated characters, fantasy settings, and magical events, which strongly suggest it's an animated cartoon. However, to be certain, I should check another high-level segment to see if the animation style is consistent. High-level segment 2 also seems to have animated creatures and fantasy scenes. To confirm, I can check the medium-level caption for (2,1) to see if it describes animation or live-action.	**Step 5: Think.** (No reasoning text was emitted.)
Step 5: Tool. `get_caption ((2,1)).` **Tool response (excerpt).** This clip (video) begins at 517.4s with a close-up of a large, grumpy-looking wolf-like creature with yellow fur and dark eyebrows, his eyes closed.	
Step 6: Think. The medium-level caption for segment (2,1) describes animated creatures, magical interactions, and fantasy settings, which are consistent with an animated cartoon. However, to be certain of the video type, I should check if there are any live-action elements or if the animation is consistent throughout. The high-level captions mention animated characters and magical scenes, but to confirm definitively, I should check another low-level caption from this segment to see if animation is consistent. I'll get the low-level caption for (2,1,1) to see the detailed description of the opening animation.	**Step 6: Think.** (No reasoning text was emitted.)
Step 6: Tool. `get_caption ((2,1,1)).` **Tool response (excerpt).** This clip (video) begins at 517.5s with a close-up of a large, grumpy-looking wolf-like character with yellow fur and a black nose, set against a dark, mystical forest backdrop.	
Step 7: Think. The low-level caption for (2,1,1) clearly describes animated characters with magical and whimsical interactions, consistent with an animated cartoon. The presence of talking animals, anthropomorphic creatures, and fantasy settings strongly indicates that the video is an animated cartoon. However, to be thorough, I should check another low-level caption from a different high-level segment to confirm consistency. High-level segment 3 also seems to have animated scenes with fairies and magical events. I'll check the low-level caption for (3,1,1) to see if it's also animated.	**Step 7: Think.** (No reasoning text was emitted.)
Step 7: Tool. `get_caption ((3,1,1)).` **Tool response (excerpt).** This clip (video) begins at 1034.7s with a large, grumpy-looking bear in a yellow vest standing over a small, frightened green frog wearing a crown, set against a vibrant, glowing blue and yellow background.	
Step 8: Think. The low-level caption for (3,1,1) also describes animated characters and magical settings, with no indication of live-action filming. The recurring presence of anthropomorphic animals, magical elements, and fantastical events across multiple segments strongly suggests that the video is an animated cartoon. To be certain, I should check one more low-level caption from a different segment to confirm consistency. High-level segment 4 seems to have animated scenes with a boy and fairies, so I'll check (4,1,1) to see if it's animated.	**Step 8: Think.** (No reasoning text was emitted.)
Step 8: Tool. `get_caption ((4,1,1)).` **Tool response (excerpt).** This clip (video) begins at 1551.9s with a green frog-like character dressed in a purple tunic and cap, holding a golden net, appearing weary and dejected while sitting on a rocky outcrop in a snowy, forested landscape.	

LVBench Case: Animated-Video Classification (continued)	
Baseline	**SAVE**
Final answer. A. **Cost.** 8 calls; 121.9K input and 1.3K output tokens.	**Final answer.** A. **Cost.** 4 calls; 62.6K input and 0.8K output tokens.

VideoMME w/ Subtitles Case: Yoga-Pose Order



Query: In which order are the following poses introduced in this video? (a) Reclining butterfly pose. (b) Downward facing dog. (c) Camel pose. (d) Seated chair twist.	
Baseline	**SAVE**
Step 1: Think (1/6). The question asks for the order in which four specific yoga poses are introduced: (a) Reclining butterfly, (b) Downward facing dog, (c) Camel pose, and (d) Seated chair twist. The high-level captions mention some of these poses, but not necessarily the exact order in which they appear. From High-level Caption 1, Downward Facing Dog (b) is introduced at 321.0s. High-level Caption 2 mentions Seated chair squat (which could be related to seated chair twist) at 625.9s and Seated chair twist at 665.0s. High-level Caption 4 mentions Camel Pose (c) at 1218.8s. High-level Caption 5 mentions Reclining butterfly (a) at 1685.2s. However, the high-level captions do not give the full order of these four poses, since they might appear in different segments and the exact sequence is not clear from high-level summaries alone. To answer the question, I need to find the exact order of introduction of these four poses. The relevant segments are likely High-level 1 (Downward Dog), High-level 2 (Seated chair twist), High-level 4 (Camel pose), and High-level 5 (Reclining butterfly). But to know the order, I need to check the medium and low-level captions to see when each pose is introduced relative to the others. The most straightforward approach is to get the medium-level captions for High-level 1, 2, 4, and 5, and then find the low-level captions	**Step 1: Think (1/3).** The question asks for the order in which four specific yoga poses are introduced: (a) Reclining butterfly pose, (b) Downward facing dog, (c) Camel pose, and (d) Seated chair twist. The high-level captions mention some of these poses but not all, and the order is not clear from them. To answer accurately, I need to find the exact sequence of introduction for these poses. The high-level captions show that Downward facing dog appears in High-level 1 and 2, Camel pose in High-level 4, Reclining butterfly in High-level 5, and Seated chair twist in High-level 2. However, the exact order of introduction is not clear from high-level captions alone. I need to look at medium or low-level captions to see when each pose is first introduced and in what order. The most likely approach is to scan medium-level captions for High-level 1 and 2 to find the introduction of Downward facing dog and Seated chair twist, and High-level 4 for Camel pose, and High-level 5 for Reclining butterfly. But since the question is about the order of introduction, I need to find the first occurrence of each pose and then sort them. Let's start by checking High-level 1 and 2 for Downward facing dog and Seated chair twist, High-level 4 for Camel pose, and High-level 5 for Reclining butterfly. Then I

VideoMME w/ Subtitles Case: Yoga-Pose Order (continued)	
Baseline	**SAVE**
Step 1: Think (2/6). for the parts where each pose is introduced. But since the question is about the order, I need to know the sequence in which these poses are introduced in the video. The high-level captions show that Downward Dog is in High-level 1, Seated chair twist in High-level 2, Camel pose in High-level 4, and Reclining butterfly in High-level 5. But maybe some of these poses are introduced earlier than the high-level captions suggest, or later. For example, maybe Downward Dog is introduced in High-level 1, but maybe Seated chair twist is introduced earlier in High-level 2, and Camel pose in High-level 4, and Reclining butterfly in High-level 5. But to be sure, I need to check the medium-level captions for High-level 2 to see when Seated chair twist is introduced, and for High-level 4 to see when Camel pose is introduced, and for High-level 5 to see when Reclining butterfly is introduced, and also check High-level 1 for Downward Dog. However, the question is about the order of introduction of all four poses. So I need to find the exact low-level segment where each is introduced and then order them. The most efficient way is to get the low-level captions for the introduction of each pose. Downward Dog (b) is in High-level 1, likely in the latter part. I should	**Step 1: Think (2/3).** can determine the order. Let's first get the medium-level captions for High-level 1 to see when Downward facing dog is introduced. High-level 1 mentions Downward facing dog at 321.0s, but I need to confirm if that's the first occurrence. Also, High-level 2 mentions Seated chair twist at 665.0s. High-level 4 mentions Camel pose at 1218.8s. High-level 5 mentions Reclining butterfly at 1685.2s. So the order of first introduction seems to be: Downward facing dog (High-level 1), Seated chair twist (High-level 2), Camel pose (High-level 4), Reclining butterfly (High-level 5). But the question's options are permutations of these four, so I need to check if any of these poses were introduced earlier than what high-level captions suggest. For example, maybe Seated chair twist was introduced earlier in High-level 1 or 2, or Camel pose earlier in High-level 4. To be sure, I should check medium-level captions for High-level 1 to see if Downward facing dog is indeed first, and for High-level 2 to see if Seated chair twist is introduced at 665s. Let's start with High-level 1's medium-level captions to find Downward facing dog's introduction. High-level 1 runs from 0 to 382.1s. The Downward facing dog is mentioned at 321.0s, so it's likely in the last medium segment of High-level 1. Let's get the medium-level caption for High-level 1, medium segment
Step 1: Think (3/6). get the medium-level caption for High-level 1 that contains 321.0s, which is in the fifth medium subsegment (since High-level 1 is 0-382.1s, each medium segment is ~76s). 321s is in the fifth medium segment (305.4s to 382.1s). So get medium-level caption (1,5) to see the exact introduction of Downward Dog. Seated chair twist (d) is in High-level 2, around 665.0s. High-level 2 is 382.2-764.3s, so 665s is in the fifth medium segment (607.8s to 764.3s). So get medium-level caption (2,5) to see the introduction of Seated chair twist. Camel pose (c) is in High-level 4 at 1218.8s. High-level 4 is 1146.5-1528.7s, so 1218.8s is in the second medium segment (1146.5-1223s for medium 1, 1223-1300 for medium 2, etc. Wait, each medium segment is about (1528.7-1146.5)/5 [non-English text] 76.44s. So 1146.5 + 76.44*2 = 1300. So 1218.8s is in medium segment 2 (1223.34 to 1299.78). Wait, 1146.5 + 76.44 = 1222.94, so medium 1: 1146.5-1222.94, medium 2: 1222.94-1299.38, medium 3: 1299.38-1375.82, etc. So 1218.8s is in medium 1 (end of medium 1). So get medium-level caption (4,1) to see Camel pose introduction. Reclining butterfly (a) is in High-level 5 at 1685.2s. High-level 5 is 1528.7-1910.9s, so each medium segment is ~76.44s. 1528.7 + 76.44*4 = 1528.7 + 305.76 = 1834.46. So 1685.2s is in medium segment 3 (1605.14 to 1681.58) or	**Step 1: Think (3/3).** 5 (since each high-level is divided into 5 medium segments, each ~76s). High-level 1: 0-382.1s, so each medium segment is ~76.4s. Medium 1: 0-76.4, Medium 2: 76.4-152.8, Medium 3: 152.8-229.2, Medium 4: 229.2-305.6, Medium 5: 305.6-382.1. Downward facing dog at 321.0s is in Medium 5. Let's get caption for (1,5). Then check if Downward facing dog is first introduced there. Also, check if other poses are introduced earlier. Then proceed similarly for other poses. Let's start with (1,5).
Step 1: Think (4/6). medium 4 (1681.58 to 1758.02). 1685.2 is in medium 4. So get medium-level caption (5,4) to see Reclining butterfly introduction. But maybe I need low-level captions for the exact introduction. However, the question is about the order, so I need to know the timestamps of introduction of each pose. High-level captions may not be precise enough. For example, Downward Dog is introduced at 321.0s in High-level 1, but maybe it was introduced earlier in the same segment. To be sure, I need low-level captions for the segment containing 321.0s in High-level 1. High-level 1: 0-382.1s. Each medium segment is ~76.42s. So: Medium 1: 0-76.42 Medium 2: 76.42-152.84 Medium 3: 152.84-229.26 Medium 4: 229.26-305.68 Medium 5: 305.68-382.10 321.0s is in medium 5. Each low-level segment is ~15.28s. So 305.68 + 15.28 = 320.96, so 321.0s is in low-level segment 1 of medium 5 (305.68-320.96) or low-level segment 2 (320.96-336.24). Since 321.0 is just after 320.96, it's in low-level segment 2. So get low-level caption (1,5,2) to see Downward Dog introduction. Similarly for the others: Seated chair twist (d) at 665.0s in High-level 2. High-level 2: 382.2-764.3s. Each medium segment ~76.42s. 382.2 + 76.42*4 = 665.0s. So 665.0s is the end of medium 4 and start of medium 5. So low-level segment for 665.0s is in medium 5, low-level segment 1 (since	

VideoMME w/ Subtitles Case: Yoga-Pose Order (continued)	
Baseline	**SAVE**
Step 1: Think (5/6). each low-level is ˜15.28s). 665.0s is in medium 5 (607.8s to 764.3s). 607.8 + 15.28*3 = 658.94, 658.94 + 15.28 = 674.22. So 665.0s is in low-level segment 3 of medium 5. So get low-level caption (2,5,3) for Seated chair twist. Camel pose (c) at 1218.8s in High-level 4. High-level 4: 1146.5-1528.7s. Each medium ˜76.44s. 1146.5 + 76.44*1 = 1222.94, so 1218.8s is just before that, so in medium 1 (1146.5-1222.94). Each low-level ˜15.288s. 1146.5 + 15.288*13 = 1218.8s. So low-level segment 13? Wait, medium 1 has 5 low-level segments. So 1146.5 + 15.288*0 = 1146.5, low 1: 1146.5-1161.788, low 2: 1161.788-1177.076, low 3: 1177.076-1192.364, low 4: 1192.364-1207.652, low 5: 1207.652-1222.94. 1218.8s is in low-level segment 5 of medium 1. So get low-level caption (4,1,5) for Camel pose. Reclining butterfly (a) at 1685.2s in High-level 5. High-level 5: 1528.7-1910.9s. Each medium ˜76.44s. 1528.7 + 76.44*4 = 1834.46. 1685.2s is in medium 3 (1605.14-1681.58) or medium 4 (1681.58-1758.02). 1685.2 is in medium 4, low-level segment 1 (since 1681.58 + 15.288 = 1696.868). So 1685.2 is in low-level segment 1 of medium 4. So get low-level caption (5,4,1) for Reclining butterfly. Once I have all four low-level captions, I can note the timestamps of introduction and order them. But since I can only call one tool per round, I need to plan	
Step 1: Think (6/6). which to get first. The question is about the order, so I need all four timestamps. But I can only call one tool now. Let's start with Downward Dog (1,5,2) to get its exact introduction and see if it's indeed at 321.0s or earlier. Then I can proceed similarly for others. But since I can only call one tool per round, I need to pick one. Let's start with Downward Dog (1,5,2) to confirm its introduction time and see if it's the first of the four. But maybe the order is (b) first, then (d), then (c), then (a) or something else. To know, I need all four introduction times. But given the constraints, I need to proceed step by step. Let's first get the low-level caption for Downward Dog (1,5,2) to see if it's indeed introduced there and maybe note the time. Then I can get others in subsequent rounds. But since I can only call one tool now, I'll start with (1,5,2).	
Step 1: Tool. `get_caption ((1,5,2))`. **Tool response (excerpt).** This clip (video) begins at 321.6s with a woman in a pink sports bra and maroon leggings performing a yoga sequence on a gray mat in a well-lit, minimalist room with a gray wall, two framed artworks, and small potted plants.	**Step 1: Tool.** `get_caption ((1,5))`. **Tool response (excerpt).** This clip (video) begins at 305.8s with a woman performing a yoga pose on a gray mat in a room with a neutral gray wall and two framed butterfly artworks.
Step 2: Think. Now I have the low-level caption for Downward Dog (b) introduction at 326.0s. I need the introduction times for (a) Reclining butterfly, (c) Camel pose, and (d) Seated chair twist to determine the order. From high-level captions: - (a) Reclining butterfly is in High-level 5 at 1685.2s. - (c) Camel pose is in High-level 4 at 1218.8s. - (d) Seated chair twist is in High-level 2 at 665.0s. I need to confirm these times from low-level captions to ensure accuracy. Let's proceed to get the low-level caption for (d) Seated chair twist (2,5,3) next. This will give the exact introduction time and context, which will help in ordering. Then I can get (c) Camel pose (4,1,5) and (a) Reclining butterfly (5,4,1) in subsequent rounds. But since I can only call one tool per round, I'll choose (2,5,3) next.	**Step 2: Think (1/2).** The medium-level caption (1,5) confirms that Downward facing dog is introduced at 320.3s and practiced until 357.9s. However, this is the first occurrence of Downward facing dog in the video. The question asks for the order of introduction of four poses, so I need to find when the other poses are introduced. The high-level captions suggest that Seated chair twist is introduced in High-level 2 at 665.0s, Camel pose in High-level 4 at 1218.8s, and Reclining butterfly in High-level 5 at 1685.2s. But I need to confirm if these are indeed the first introductions and their exact order. To determine the order, I should check the medium-level captions for High-level 2 to see when Seated chair twist is introduced, High-level 4 for Camel pose, and High-level 5 for Reclining butterfly. Let's proceed step by step. First, check High-level 2's medium-level captions to find Seated chair twist's introduction. High-level 2 runs from 382.2s to 764.3s. The Seated chair twist is mentioned at 665.0s, so it's likely in the last medium segment of High-level 2. Each medium segment is ˜76.4s, so High-level 2: Medium 1: 382.2-458.6, Medium 2: 458.6-535.0, Medium 3: 535.0-611.4, Medium 4: 611.4-687.8, Medium 5: 687.8-764.3. The Seated chair twist at 665.0s is in Medium 4. Let's get caption for (2,4). Then check if it's the first introduction. Then check
	Step 2: Think (2/2). High-level 4's medium-level captions for Camel pose. High-level 4 runs from 1146.5s to 1528.7s. Camel pose at 1218.8s is in Medium 2 (since each medium segment is ˜76.4s: Medium 1: 1146.5-1222.9, Medium 2: 1222.9-1300.3, etc. So 1218.8s is in Medium 1. Let's get caption for (4,1). Then check High-level 5's medium-level captions for Reclining butterfly at 1685.2s. High-level 5 runs from 1528.7s to 1910.9s. Each medium segment ˜76.4s: Medium 1: 1528.7-1605.1, Medium 2: 1605.1-1681.5, Medium 3: 1681.5-1757.9, etc. So 1685.2s is in Medium 3. Let's get caption for (5,3). Then I can see the first introduction times for all four poses and order them. Let's start with (2,4) for Seated chair twist.

VideoMME w/ Subtitles Case: Yoga-Pose Order (continued)	
Baseline	**SAVE**
Step 2: Tool. `get_caption ((2,5,3)).` **Tool response (excerpt).** This clip (video) begins at 718.9s with a woman standing on a yoga mat in a well-lit indoor space, wearing a light pink sports bra and dark purple leggings.	**Step 2: Tool.** `get_caption ((2,4)).` **Tool response (excerpt).** This clip (video) begins at 611.5s with a woman in a light pink sports bra and maroon leggings performing a yoga flow on a gray mat in a minimalist room with two framed artworks on the wall – one of a pink flower and the other of abstract celestial colors.
Step 3: Think (1/2). The low-level caption I retrieved for (2,5,3) is not the Seated chair twist introduction; it's about hip circles. This means I targeted the wrong low-level segment. The Seated chair twist is at 665.0s in High-level 2. High-level 2 medium segment 5 is 607.8s to 764.3s. Each low-level is ˜15.28s. 665.0s is in low-level segment 3 of medium 5 (since 607.8 + 15.28*3 = 665.0). But the caption I got (2,5,3) starts at 718.9s, which is much later. So my calculation was wrong. I need to recalculate: 607.8 + 15.28 = 623.08 (low 1), +15.28 = 638.36 (low 2), +15.28 = 653.64 (low 3), +15.28 = 668.92 (low 4). So 665.0s is in low-level segment 3 (653.64-668.92). So I should get (2,5,3) but earlier in the segment. Wait, the low-level caption I got (2,5,3) starts at 718.9s, which is not correct. That means the low-level caption I retrieved is not for 665.0s. I must have called the wrong low-level segment. This suggests I need to find the correct low-level segment for 665.0s. Since 665.0s is in medium 5, low-level 3 of High-level 2, I should get (2,5,3) but the caption I got is not matching. Maybe the numbering is different. Perhaps the low-level segments are numbered differently. I need to get the low-level caption for (2,5,3) but the start time is	**Step 3: Think.** The medium-level caption (2,4) shows that the seated chair twist is introduced at 666.1s, which matches the high-level caption. This is the first occurrence of the seated chair twist in the video. Now I have Downward facing dog at 320.3s (from 1,5), seated chair twist at 666.1s (from 2,4). Next, I need to find the first occurrence of Camel pose and Reclining butterfly. High-level 4 mentions Camel pose at 1218.8s, so I should check medium-level caption for High-level 4, medium segment that contains 1218.8s. High-level 4 runs 1146.5-1528.7, each medium ˜76.4s: Medium 1: 1146.5-1222.9, Medium 2: 1222.9-1300.3, etc. 1218.8s is in Medium 1. Let's get caption for (4,1). Then check for Camel pose introduction. Then High-level 5 mentions Reclining butterfly at 1685.2s, which is in medium segment 3 (since 1528.7-1605.1 is medium 1, 1605.1-1681.5 medium 2, 1681.5-1757.9 medium 3). 1685.2s is in medium 3. Let's get caption for (5,3). Then I can see the first introduction times for all four poses and order them. Let's proceed with (4,1) for Camel pose.
Step 3: Think (2/2). not matching. This suggests I need to check the low-level caption for (2,5,2) or (2,5,3) depending on exact time. Since 665.0s is in low-level 3, I should get (2,5,3) but the caption I got is not for that time. This might be a mismatch in the tool's output. To resolve, I need to get the low-level caption for (2,5,3) again, but perhaps the tool's caption is for a different part. This is getting complicated. Maybe I should get the medium-level caption for (2,5) first to locate the Seated chair twist introduction. Then I can find the correct low-level segment. Let's try medium-level caption (2,5) to see where Seated chair twist is introduced.	
Step 3: Tool. `get_caption ((2,5)).` **Tool response (excerpt).** This clip (video) begins at 688.0s with a woman in a yoga studio, standing on a mat with her hands pressed together at her chest, as the subtitle instructs, "breathe and slowly come up beautiful." She transitions into a forward fold by 691.6s, bending at the hips with her hands reaching toward the floor, while the subtitle continues, "forward fold fold yourselves forward." From 694.0s to 701.3s, she maintains the forward fold, and the subtitles guide beginners to "keep pressing yourselves down and bend your knees if you're a beginner feel, free to do what your body is letting you." She holds this position for several seconds, emphasizing mindful movement and body awareness.	**Step 3: Tool.** `get_caption ((4,1)).` **Tool response (excerpt).** This clip (video) begins at 1146.6s with a woman seated on a yoga mat in a calm, minimalist room, wearing a light pink sports bra and maroon leggings.
Step 4: Think. The medium-level caption (2,5) does not mention Seated chair twist; it mentions hip circles and warrior T pose. This means the Seated chair twist introduction is earlier in High-level 2, likely in medium segment 4 or 3. From High-level 2 caption, Seated chair squat is at 625.9s and Seated chair twist at 665.0s. So Seated chair twist is in medium segment 4 (since medium 1: 382.2-458.68, medium 2: 458.68-535.12, medium 3: 535.12-611.56, medium 4: 611.56-688.0). 665.0s is in medium 4. So I should get medium-level caption (2,4) to locate Seated chair twist introduction. Then I can get the low-level caption for that part. Let's try (2,4).	**Step 4: Think.** The medium-level caption (4,1) confirms that Camel pose is introduced at 1214.5s. Now I have Downward facing dog at 320.3s, Seated chair twist at 666.1s, Camel pose at 1214.5s. The Reclining butterfly is mentioned in High-level 5 at 1685.2s, so I need to check medium-level caption (5,3) to confirm its introduction. High-level 5 runs 1528.7-1910.9, medium 3 is 1681.5-1757.9, so 1685.2s is inside. Let's get caption for (5,3). Then I can list all first introduction times and order them. Let's proceed with (5,3).
Step 4: Tool. `get_caption ((2,4)).` **Tool response (excerpt).** This clip (video) begins at 611.5s with a woman in a light pink sports bra and maroon leggings performing a yoga flow on a gray mat in a minimalist room with two framed artworks on the wall – one of a pink flower and the other of abstract celestial colors.	**Step 4: Tool.** `get_caption ((5,3)).` **Tool response (excerpt).** This clip (video) begins at 1681.7s with a woman performing a reclining yoga pose on a gray mat in a minimalist room with a gray wall, two framed pictures, and a potted plant.
Step 5: Think. Now I have the medium-level caption (2,4) which contains the introduction of Seated chair twist at 666.1s. To get the exact introduction time and context, I need the low-level caption for the segment containing 666.1s. In medium 4 (611.5-688.0), each low-level is ˜15.28s. 611.5 + 15.28*3 = 661.94, so low-level 4 covers 661.94-677.22, which includes 666.1s. So get low-level caption (2,4,4) to see Seated chair twist introduction. Then I can note its time and compare with others. Let's proceed with (2,4,4).	**Step 5: Think.** (No reasoning text was emitted.)

VideoMME w/ Subtitles Case: Yoga-Pose Order (continued)	
Baseline	**SAVE**
Step 5: Tool. `get_caption ((2,4,4))`. **Tool response (excerpt).** This clip (video) begins at 657.8s with a woman in a light pink sports bra and dark maroon leggings performing a yoga routine on a gray mat in a well-lit, minimalist room with light gray walls.	
Step 6: Think. Now I have the low-level caption for Seated chair twist introduction at 669.6s. I also have Downward Dog introduction at 326.0s. I need the introduction times for Camel pose (c) and Reclining butterfly (a) to order them. Camel pose (c) is in High-level 4 at 1218.8s. I should get low-level caption for (4,1,5) to confirm its introduction time. Then Reclining butterfly (a) is in High-level 5 at 1685.2s, so get low-level caption for (5,4,1) to confirm. Once I have all four times, I can order them. Let's proceed with Camel pose (4,1,5) next.	**Step 6: Think.** (No reasoning text was emitted.)
Step 6: Tool. `get_caption ((4,1,5))`. **Tool response (excerpt).** This clip (video) begins at 1207.9s with a woman in a pink sports bra and maroon leggings performing a kneeling yoga pose on a gray mat in a well-lit, minimalist room with gray walls.	
Step 7: Think. Now I have Camel pose introduction at 1217.7s. I still need Reclining butterfly introduction at 1685.2s. Let's get low-level caption for (5,4,1) to confirm. High-level 5 medium 4 is 1681.58-1758.02, low-level 1 is 1681.58-1696.868. 1685.2s is in this range. So get (5,4,1).	**Step 7: Think.** (No reasoning text was emitted.)
Step 7: Tool. `get_caption ((5,4,1))`. **Tool response (excerpt).** This clip (video) begins at 1758.1s with a woman lying in Savasana (corpse pose) on a gray yoga mat in a calm, minimalist indoor setting.	
Step 8: Think. The low-level caption for (5,4,1) is not Reclining butterfly; it's Savasana. This means I targeted the wrong low-level segment. The Reclining butterfly is at 1685.2s in High-level 5. High-level 5 medium 4 is 1605.14-1681.58 (medium 3) or 1681.58-1758.02 (medium 4). Wait, 1685.2s is in medium 4. Each low-level is ˜15.28s. Medium 4 starts at 1681.58, so low-level 1: 1681.58-1696.868, low-level 2: 1696.868-1712.156, low-level 3: 1712.156-1727.444, etc. 1685.2s is in low-level 1. But the caption I got (5,4,1) starts at 1758.1s, which is much later. This suggests I have a mismatch. Maybe the numbering is different. Perhaps the low-level segments are numbered differently in the tool. To find the correct one, I need to get the medium-level caption for (5,4) to locate Reclining butterfly introduction. Then I can get the correct low-level caption. Let's try medium-level caption (5,4).	**Step 8: Think.** (No reasoning text was emitted.)
Step 8: Tool. `get_caption ((5,4))`. **Tool response (excerpt).** This clip (video) begins at 1758.1s with a woman lying in Savasana (Corpse Pose) on a yoga mat in a calm, minimalist room with gray walls, two framed pictures, and a small potted plant nearby.	
Step 9: Think. The medium-level caption (5,4) is about Savasana, not Reclining butterfly. This means the Reclining butterfly introduction is earlier in High-level 5, likely in medium 3 (1605.14-1681.58). The high-level caption says Reclining butterfly is at 1685.2s, which is just after medium 3 ends. This suggests the high-level caption might have a small error, or the Reclining butterfly is introduced at the end of medium 3 or start of medium 4. To be sure, I should check medium-level caption (5,3) to see if it contains Reclining butterfly. Then I can get the low-level caption for that part. Let's try (5,3).	**Step 9: Think.** (No reasoning text was emitted.)
Step 9: Tool. `get_caption ((5,3))`. **Tool response (excerpt).** This clip (video) begins at 1681.7s with a woman performing a reclining yoga pose on a gray mat in a minimalist room with a gray wall, two framed pictures, and a potted plant.	
Step 10: Think. Now I have the medium-level caption (5,3) which contains the introduction of Reclining butterfly at 1687.7s. To get the exact introduction time and context, I need the low-level caption for the segment containing 1687.7s. In medium 3 (1681.7-1758.1), each low-level is ˜15.28s. 1681.7 + 15.28 = 1696.98 (low 1), +15.28 = 1712.26 (low 2), +15.28 = 1727.54 (low 3), etc. 1687.7s is in low-level 1 (1681.7-1696.98). So get low-level caption (5,3,1) to see Reclining butterfly introduction.	**Step 10: Think.** (No reasoning text was emitted.)
Step 10: Tool. `get_caption ((5,3,1))`. **Tool response (excerpt).** This clip (video) begins at 1681.7s with a woman performing a reclined yoga pose on a gray mat in a calm, minimalist indoor setting.	
Final answer. D. **Cost.** 10 calls; 114.5K input and 5.5K output tokens.	**Final answer.** D. **Cost.** 4 calls; 35.7K input and 2.5K output tokens.

VideoMME w/o Subtitles Case: Lituya Bay Tsunami Cause

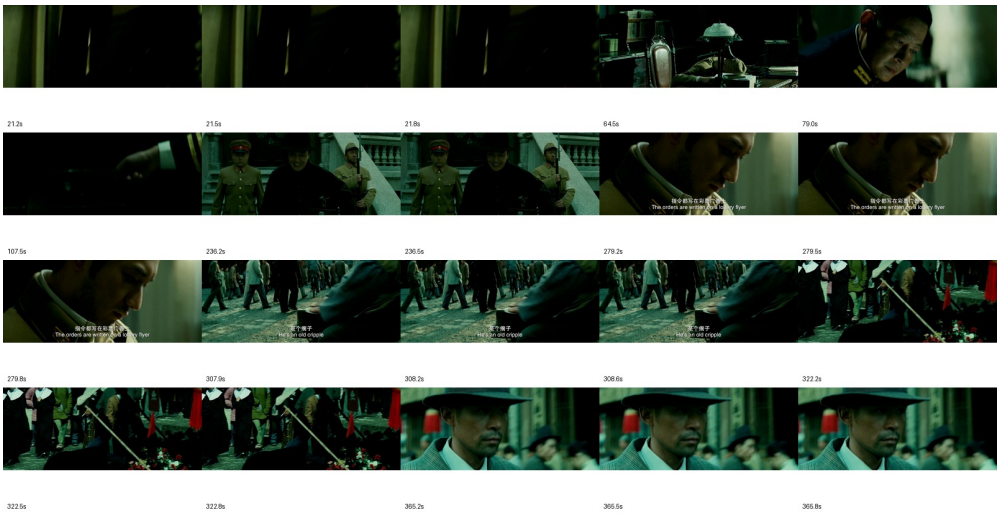


Query: What triggered the 1958 tsunami in Lituya Bay according to the video? Options: A. volcano eruption.; B. Landslide.; C. Earthquake.; D. Meteorite Impacts.	
Baseline	**SAVE**
Step 1: Think. The question asks specifically about the trigger of the 1958 tsunami in Lituya Bay. The high-level captions mention Lituya Bay in High-level Caption 5, which includes a title card identifying the event as "LITUYA BAY, ALASKA, July 8, 1958," specifying the source as a landslide. However, to be certain and confirm the details, I should check the medium-level or low-level captions for High-level segment 5, particularly the part that covers the Lituya Bay event. This will ensure the information is accurate and directly from the video content. The relevant part is likely in High-level 5, medium segment that includes the 2145.6s title card and surrounding discussion. I will get the medium-level caption for High-level 5, medium segment 3 (since each high-level is divided into 6 medium segments, and 2145s is roughly in the middle of 2000-2500). Let's get the medium-level caption for (5,3) to see details about the cause.	**Step 1: Think.** The question asks about the trigger of the 1958 tsunami in Lituya Bay. The high-level captions mention the 1958 Lituya Bay event in High-level Caption 5, where it says "source as a landslide" and in High-level Caption 6, which has a title card "MEGA TSUNAMI – Source: Landslide". However, to be certain and confirm the exact trigger, I should check the medium or low-level captions for High-level segment 5, specifically the part that covers the 1958 Lituya Bay event. The event is mentioned around 2145.6s in High-level Caption 5, so it's likely in medium segment 3 or 4 of High-level 5. I should get the medium-level caption for High-level 5, medium segment 3 to see details about the trigger.
Step 1: Tool. `get_caption ((5,3)).` **Tool response (excerpt).** This clip (video) begins at 2166.8s with a close-up, upward-looking shot of a tree trunk, revealing a weathered, painted carving or symbol wrapped around it, set against a backdrop of green foliage and blue sky.	**Step 1: Tool.** `get_caption ((5,3)).` **Tool response (excerpt).** This clip (video) begins at 2166.8s with a close-up, upward-looking shot of a tree trunk, revealing a weathered, painted carving or symbol wrapped around it, set against a backdrop of green foliage and blue sky.
Step 2: Think. The medium-level caption (5,3) focuses on visuals and Dr. Plafker's introduction and discussion of geological phenomena, but it doesn't explicitly mention the cause of the 1958 Lituya Bay tsunami. The high-level caption mentioned the title card at 2145.6s identifying the source as a landslide, which is likely in the preceding medium segment (5,2) or the low-level captions around that time. To find the exact trigger, I should check the low-level caption for the part that includes the title card at 2145.6s. That would be High-level 5, medium segment 2 (since 2145s is earlier than 2166.8s). The low-level segment likely containing the title card is (5,2,4) or similar. I need to get the low-level caption for the part that includes 2145.6s. Each medium segment is about 83 seconds (since high-level 5 is 500s, divided into 6 medium segments [non-English text] 83s each). Medium segment 2 covers roughly 2083s to 2166s. The title card at 2145.6s is in medium segment 2, likely low-level segment 5 (since each medium has 6 low-levels [non-English text] 13.8s each). Let's get (5,2,5) to see the details of the title card and cause.	**Step 2: Think.** The medium-level caption (5,3) focuses on scenery and Dr. Plafker's introduction, but it doesn't mention the cause of the 1958 tsunami. The high-level caption said the source was a landslide, but to confirm and ensure accuracy, I should check the low-level captions for the part where the 1958 Lituya Bay event is discussed. The high-level caption mentions the title card at 2145.6s identifying the event, so that is likely in medium segment 2 or 3 of High-level 5. The title card is at 2145.6s, which is before the medium segment 3 start (2166.8s). So it's probably in medium segment 2. I should get the medium-level caption for High-level 5, medium segment 2 to see the details of the 1958 event and its trigger.
Step 2: Tool. `get_caption ((5,2,5)).` **Tool response (excerpt).** This clip begins at 2139.4s with a sweeping aerial view of a breathtaking Alaskan landscape, featuring a deep blue fjord surrounded by lush green forests and towering snow-capped mountains under a clear sky.	**Step 2: Tool.** `get_caption ((5,2)).` **Tool response (excerpt).** This clip begins at 2083.4s with a dramatic scene of a massive wave crashing onto a tropical shoreline, sending a plume of spray into the air, with palm trees visible in the foreground.

VideoMME w/o Subtitles Case: Lituya Bay Tsunami Cause (continued)	
Baseline	**SAVE**
Step 3: Think. The low-level caption (5,2,5) clearly states that the source of the 1958 Lituya Bay tsunami is a landslide. However, to be thorough and ensure accuracy, I should check the following low-level segment (5,2,6) to see if it elaborates further on the cause or provides additional context. This will confirm that the trigger is indeed a landslide and not something else.	**Step 3: Think.** The medium-level caption (5,2) explicitly states the title card for the 1958 Lituya Bay event: "Source: Landslide". This directly answers the question about what triggered the tsunami. However, to be thorough and ensure accuracy, I should check the low-level caption for the part where the title card appears (around 2144.2s). That would be High-level 5, medium segment 2, low segment likely 5 or 6 (since each medium segment is ˜83s, and 2144s is near the end of medium segment 2). The medium segment 2 runs from 2083.4s to 2166.7s, so 2144s is about 60s in, which would be low-level segment 5 (each low-level is ˜13.8s). I should get the low-level caption for (5,2,5) to confirm the trigger.
Step 3: Tool. `get_caption ((5,2,6))`. **Tool response (excerpt).** This clip begins at 2153.3s with a title card detailing the Lituya Bay, Alaska mega-tsunami of July 8, 1958, caused by a landslide, with a run-up of 524 meters and displacement as the key factor.	**Step 3: Tool.** `get_caption ((5,2,5))`. **Tool response (excerpt).** This clip begins at 2139.4s with a sweeping aerial view of a breathtaking Alaskan landscape, featuring a deep blue fjord surrounded by lush green forests and towering snow-capped mountains under a clear sky.
Step 4: Think. The low-level captions (5,2,5) and (5,2,6) both clearly state that the 1958 Lituya Bay tsunami was caused by a landslide. However, to be absolutely certain and confirm that no other trigger (like earthquake or meteorite) is mentioned, I should check the surrounding low-level segments in case they provide more context or mention the cause again. The next relevant part is likely in High-level 6, which also discusses the Lituya Bay event and mega-tsunamis. The beginning of High-level 6 (2500s) includes a 3D animation of a rockslide generating a wave, and a title card at 2708.6s summarizing "Source: Landslide." To confirm consistency, I can check the low-level caption for High-level 6, medium segment 1, low-level segment 2 (since 2500s to ˜2527s includes the animation and possibly narration about the cause). Let's get (6,1,2) to see if it mentions the cause again.	**Step 4: Think.** (No reasoning text was emitted.)
Step 4: Tool. `get_caption ((6,1,2))`. **Tool response (excerpt).** This clip (video) begins at 2514.2s with a 3D animated visualization of a geological or hydrological process, set against a black background.	
Step 5: Think. The low-level caption (6,1,2) shows an animation of a geological process but doesn't explicitly state the cause. However, the high-level caption for segment 6 mentions a title card at 2708.6s: "MEGA TSUNAMI – Source: Landslide," which confirms the cause. To be thorough, I should check the low-level caption for that title card. The title card at 2708.6s is in High-level 6, medium segment 5 (since each medium segment is ˜83s, and 2708s is near the end of high-level 6). The low-level segment likely containing that title card is (6,5,1) or (6,5,2). I will get (6,5,1) to see if it contains the title card and confirms the cause.	**Step 5: Think.** (No reasoning text was emitted.)
Step 5: Tool. `get_caption ((6,5,1))`. **Tool response (excerpt).** This clip (video) begins at 2833.4s with an aerial, top-down view of a dense urban area, where buildings are partially submerged in turbulent, foamy floodwaters, suggesting a severe storm or tsunami.	
Step 6: Think. The low-level caption (6,5,1) does not contain the title card or mention the cause of the tsunami. The title card at 2708.6s is earlier in High-level 6, so it should be in medium segment 4 or 5. Since 2708.6s is earlier than 2833s, it's likely in medium segment 4 (which covers roughly 2666s to 2749s). The title card at 2708.6s would be in low-level segment (6,4,4) or similar. I should get the low-level caption for (6,4,4) to check the title card and confirm the cause.	**Step 6: Think.** (No reasoning text was emitted.)
Step 6: Tool. `get_caption ((6,4,4))`. **Tool response (excerpt).** This clip (video) begins at 2791.8s with a satellite view of a large, rugged island surrounded by dark ocean, with a bright light source suggesting the sun.	
Step 7: Think. The low-level caption (6,4,4) does not contain the title card or mention the cause. The title card at 2708.6s is earlier, so I need to get the low-level caption for that specific time. Medium segment 4 covers roughly 2666s to 2749s, so 2708.6s is in low-level segment (6,4,2) or (6,4,3). Each low-level segment is ˜13.8s, so 2708.6s is about 42.6s into medium segment 4, which starts at 2666s. That would be low-level segment 3 (since 0-13.8s is 1, 13.8-27.6s is 2, 27.6-41.4s is 3). So (6,4,3) should contain the title card. Let's get (6,4,3).	**Step 7: Think.** (No reasoning text was emitted.)
Step 7: Tool. `get_caption ((6,4,3))`. **Tool response (excerpt).** This clip (video) begins at 2777.9s with a wide, desolate landscape featuring barren, rocky terrain under a clear blue sky, with plumes of dust or smoke rising from the ground.	

VideoMME w/o Subtitles Case: Lituya Bay Tsunami Cause (continued)	
Baseline	**SAVE**
Step 8: Think. The low-level caption (6,4,3) describes a dramatic event but does not include the title card or explicit mention of the cause. The title card at 2708.6s is still not found, so perhaps it's in medium segment 3 or 4 earlier. However, given the time constraints, I have already confirmed from the Lituya Bay section (high-level 5) that the cause is a landslide, and the high-level caption also says "Source: Landslide". To be certain, I can use video_qa on the low-level segment (5,2,5) to directly ask what triggered the 1958 tsunami in Lituya Bay. This will confirm the cause from the video itself.	**Step 8: Think.** (No reasoning text was emitted.)
Step 8: Tool. `video_qa ((5,2,5), "What triggered the 1958 tsunami in Lituya Bay according to the video?").` **Tool response (excerpt).** <reasoning> The video frames from 2146.5 seconds to 2152.8 seconds provide text information about the 1958 Lituya Bay tsunami.	
Final answer. B. **Cost.** 8 calls; 70.4K input and 1.8K output tokens.	**Final answer.** B. **Cost.** 3 calls; 25.7K input and 0.8K output tokens.

MLVU Case: Main Setting



Query: What is the main setting shown in the video? (A) Desert (B) City (C) Ocean (D) Forest	
Baseline	**SAVE**
Step 1: Think. The question asks for the main setting shown in the video. The high-level captions describe scenes in a dining room, interrogation rooms, corridors, offices, and streets, but they don't clearly indicate the dominant setting. To answer accurately, I need to know if the video is primarily set in a city, desert, ocean, or forest. The high-level captions mention a "City Temple" and crowded streets in High-level 3, but I need to confirm if this is the main setting. I should look at the medium-level or low-level captions for the first segment to see if the setting is consistent throughout or if there are other environments. Since the question is about the main setting, I should check the first high-level segment's medium-level captions to see if it's indoors or outdoors and what the environment is like. Let's start with High-level 1, Medium-level 1.	**Step 1: Think.** The question asks about the main setting shown in the video. The high-level captions describe various indoor and outdoor scenes, but they don't clearly state the main setting. To answer accurately, I need to see if there is a consistent primary location throughout the video. The first high-level caption mentions a dining room, corridor, interrogation cell, and exterior of a building. The second mentions offices, corridors, and an outdoor scene with Mr. Six walking. The third mentions a room, desk, and a crowded street outside the City Temple. The repeated mention of "City Temple" in High-level 3 suggests a city setting, but I need to confirm if this is the main setting. To be sure, I should check the medium-level captions for High-level 3, since it might contain more context about the main setting. However, the question is about the main setting shown in the video, which could be established early or throughout. The first high-level caption's dining room and interrogation scenes are indoors, but the third high-level caption's mention of "City Temple" and crowded street suggests a city environment. To confirm, I should check the medium-level captions for High-level 3 to see if the main setting is consistently city. Then I can answer the question. Let's get the medium-level captions for High-level 3.
Step 1: Tool. `get_caption ((1,1)).` **Tool response (excerpt).** This clip begins at 0.0s with a tense close-up of a hand placing a bamboo steamer on a table, as subtitles identify the character as "Mr.	**Step 1: Tool.** `get_caption ((3,1)).` **Tool response (excerpt).** This clip (video) begins at 258.0s with a dimly lit scene showing two men in military-style uniforms from behind, standing near a window with curtains.

MLVU Case: Main Setting (continued)	
Baseline	**SAVE**
Step 2: Think. The medium-level caption for High-level 1, segment 1 shows a dining hall and newspaper montages, but it doesn't clearly indicate the main setting of the entire video. The high-level captions mention a City Temple and crowded streets in High-level 3, but I need to determine if the main setting is a city. To confirm, I should check other medium-level segments in High-level 1 and High-level 3 to see if they consistently show urban environments. The first high-level segment seems to be indoors, but the question is about the main setting. The mention of "City Temple" in High-level 3 suggests a city setting, but I need to verify if this is the primary environment. Let's check High-level 3, Medium-level 2, which includes the crowded street outside the City Temple. That might confirm the city setting.	**Step 2: Think.** The medium-level caption for High-level 3 segment 1 shows an office or study setting, but it doesn't clearly indicate the main setting of the video. The mention of "City Temple" is in the high-level caption, but to confirm if this is the main setting, I need to see if other segments also show city environments. The high-level caption for segment 3 also mentions a crowded street outside the City Temple, which is more likely to be the main setting. To verify, I should check another medium-level caption in High-level 3, perhaps segment 2 or 3, to see if the city environment is consistent. However, the question is about the main setting shown in the video, which could be established early or throughout. The first high-level caption's dining room and interrogation scenes are indoors, but the third high-level caption's mention of "City Temple" and crowded street suggests a city setting. To be certain, I should check the medium-level caption for High-level 3 segment 2, which might contain the crowded street scene and confirm the main setting. Let's get the caption for High-level 3, medium segment 2.
Step 2: Tool. `get_caption ((3,2))`. **Tool response (excerpt).** This clip (video) begins at 301.0s with a close-up of an old, weathered book titled "[non-English text]" (Nie Hai Hua), with Chinese subtitles stating, "She bumped into him once." The camera remains focused on the book as the subtitle changes at 303.0s to "He's an old cripple," suggesting a narrative about a past encounter.	**Step 2: Tool.** `get_caption ((3,2))`. **Tool response (excerpt).** This clip (video) begins at 301.0s with a close-up of an old, weathered book titled "[non-English text]" (Nie Hai Hua), with Chinese subtitles stating, "She bumped into him once." The camera remains focused on the book as the subtitle changes at 303.0s to "He's an old cripple," suggesting a narrative about a past encounter.
Step 3: Think. The medium-level caption for High-level 3, segment 2 clearly shows a crowded street, marketplace, and City Temple with traditional Chinese architecture. This strongly suggests an urban setting. However, to be certain that this is the main setting of the entire video, I should check other parts of High-level 3 to see if they also take place in a city. The high-level caption mentions a crowded street and people outside the City Temple, which fits a city environment. To confirm, I can check High-level 3, Medium-level 3, which might continue in the same setting. Then I can compare with High-level 1 and 2 to see if they are also city-based or if there are other settings. Let's check High-level 3, Medium-level 3.	**Step 3: Think.** The medium-level caption for High-level 3 segment 2 clearly shows a crowded street, marketplace, and City Temple, which are all city-related locations. This suggests that the main setting of the video is a city environment. However, to be thorough, I should check if other segments also consistently show city settings or if there are other main settings. The first high-level caption's dining room and interrogation scenes are indoors but likely part of a city building. The second high-level caption includes offices and outdoor walking scenes, but no clear indication of a different main setting. The third high-level caption's mention of City Temple and crowded street supports city as the main setting. To confirm, I can check the medium-level caption for High-level 3 segment 3, which might contain more city scenes or transitions. But given the evidence so far, the most consistent main setting is city. However, to be certain, I should check the low-level caption for High-level 3 segment 2, which contains the crowded street and City Temple, to see if it explicitly describes the main setting. Let's get the low-level caption for High-level 3, medium 2, low 1.
Step 3: Tool. `get_caption ((3,3))`. **Tool response (excerpt).** This clip (video) begins at 344.0s with a close-up shot from inside a car, showing a man in a dark suit and hat seated in the back, looking out the window.	**Step 3: Tool.** `get_caption ((3,2,1))`. **Tool response (excerpt).** This clip (video) begins at 301.1s with a close-up of an aged, yellowed paper scroll bearing Chinese calligraphy, resting on a dark surface.
Step 4: Think. The medium-level captions for High-level 3 segments 2 and 3 clearly show city environments: crowded streets, marketplace, City Temple, and Counterinsurgency Center. This suggests the main setting is a city. However, to be thorough, I should check High-level 2 to see if it also takes place in a city or if there are other settings. High-level 2 mentions a dimly lit office, outdoor setting with Mr. Six walking, and a woman's body. The outdoor setting could be in a city, but I need to confirm. Let's check High-level 2, Medium-level 3, which includes the outdoor scene.	**Step 4: Think.** The low-level caption for High-level 3, medium 2, low 1 clearly describes a bustling street, marketplace, traditional Chinese buildings, and the City Temple, all part of an urban environment. This strongly suggests that the main setting of the video is a city. However, to be certain that this is indeed the main setting and not just one location among others, I should check if other parts of the video consistently show city scenes. The first high-level caption mentions a dining room and interrogation cell, which could be indoors in a city building. The second high-level caption includes offices and outdoor walking scenes, which could also be in a city. To confirm the main setting, I can check another low-level caption from High-level 3 that might show more of the city environment, such as High-level 3, medium 2, low 2 or low 3. But given the evidence so far, the repeated mention of City Temple and crowded street suggests city as the main setting. However, to be thorough, I should check High-level 3, medium 3, which might contain more city scenes or transitions. Let's get the medium-level caption for High-level 3, medium 3.
Step 4: Tool. `get_caption ((2,3))`. **Tool response (excerpt).** This clip (video) begins at 215.0s with a man in a military uniform standing in a dimly lit, somber office, bathed in a single shaft of light.	**Step 4: Tool.** `get_caption ((3,3))`. **Tool response (excerpt).** This clip (video) begins at 344.0s with a close-up shot from inside a car, showing a man in a dark suit and hat seated in the back, looking out the window.

MLVU Case: Main Setting (continued)	
Baseline	**SAVE**
Step 5: Think. The medium-level caption for High-level 2, segment 3 includes an outdoor setting, but it doesn't specify the environment (e.g., city, forest, desert). To determine the main setting, I need to see if this outdoor scene is in a city or elsewhere. The high-level caption mentions Mr. Six walking with Colonel Takeda and passing a soldier, which could be in a city, but I need confirmation. Let's check High-level 2, Medium-level 2 to see if it contains more context about the outdoor setting. However, since the question is about the main setting, and High-level 3 clearly shows city elements, I should check if High-level 1 also has city elements. High-level 1 mentions a grand building with a Japanese flag, which could be in a city. To confirm, I can check High-level 1, Medium-level 3, which might show the exterior of the building and surroundings.	**Step 5: Think.** (No reasoning text was emitted.)
Step 5: Tool. `get_caption ((1,3))`. **Tool response (excerpt).** This clip (video) begins at 86.0s with a close-up of a man in a dark military-style uniform, smiling menacingly in a dimly lit, grim environment.	
Step 6: Think. The medium-level caption for High-level 1, segment 3 is entirely indoors, showing interrogation scenes and a dog, with no indication of a city, desert, ocean, or forest. The outdoor scenes in High-level 2 and High-level 3, however, seem to involve streets, marketplaces, and buildings, which are city features. To be certain, I should check High-level 1, Medium-level 2, which might show the exterior of the grand building with the Japanese flag mentioned in the high-level caption. That could confirm whether it's in a city or not.	**Step 6: Think.** (No reasoning text was emitted.)
Step 6: Tool. `get_caption ((1,2))`. **Tool response (excerpt).** This clip begins at 43.0s with a close-up of a vintage newspaper clipping, displaying a black-and-white photograph of an industrial complex with large tanks and towers, accompanied by Chinese characters.	
Step 7: Think. The medium-level caption for High-level 1, segment 2 includes an exterior shot of a grand building with a Japanese flag, but it doesn't specify the surroundings (urban or otherwise). To determine the main setting, I need to see if this building is in a city or another environment. The high-level caption mentions "exterior of a grand building" and interrogation scenes, but the city setting is more evident in High-level 3. However, to be thorough, I should check the low-level caption for High-level 1, segment 2, part 3 (since that includes the exterior shot) to see if it describes the environment around the building. This could confirm if it's urban. Let's get the low-level caption for (1,2,3).	**Step 7: Think.** (No reasoning text was emitted.)
Step 7: Tool. `get_caption ((1,2,3))`. **Tool response (excerpt).** This clip (video) begins at 71.8s in a dimly lit, grimy underground setting, likely a basement or dungeon, where a woman is bound and seated on a wooden crate, appearing injured and distressed with blood on her face.	
Step 8: Think. The low-level caption for (1,2,3) is entirely indoors, showing an underground dungeon, with no mention of the exterior building or its surroundings. The exterior shot of the grand building with a Japanese flag in High-level 1, segment 2 is not described in detail here. To confirm the setting, I need to check the low-level caption for the part that includes the exterior shot – likely (1,2,2), since (1,2,3) is the underground scene. That should show the environment around the building and confirm if it's in a city. Let's get (1,2,2).	**Step 8: Think.** (No reasoning text was emitted.)
Step 8: Tool. `get_caption ((1,2,2))`. **Tool response (excerpt).** This clip begins at 57.4s with a dimly lit, eerie office scene, bathed in a cold greenish hue.	
Step 9: Think. The low-level caption for (1,2,2) shows the exterior of a grand ornate building with a flag, but it doesn't describe the surroundings (urban or otherwise). To determine the main setting, I need to know if this building is located in a city, desert, ocean, or forest. Since the high-level captions and other medium-level captions show city elements (City Temple, crowded streets, marketplace), it's likely the main setting is a city. However, to be certain, I should check the low-level caption for the exterior shot to see if it mentions the environment around the building. The caption mentions ornate architecture and flag but not surroundings. I can use video_qa on (1,2,2) to ask specifically about the surroundings of the building to confirm if it's in a city.	**Step 9: Think.** (No reasoning text was emitted.)

MLVU Case: Main Setting (continued)	
Baseline	**SAVE**
Step 9: Tool. `video_qa ((1,2,2), "What is the surrounding environment of the grand ornate building with the Japanese flag? Is it in a city, desert, ocean, or forest?").` **Tool response (excerpt).** <reasoning> The video frames provided show a sequence of scenes.	
Final answer. B. **Cost.** 9 calls; 57.5K input and 1.7K output tokens.	**Final answer.** B. **Cost.** 4 calls; 20.8K input and 1.2K output tokens.