

# LensWalk: Agentic Video Understanding by Planning How You See in Videos

## Supplementary Material

### A. Method Details

This appendix provides detailed designs of the LensWalk agent, including the overall reason–plan–observe algorithm used in our framework, the concrete prompts for each module, and tool schemas.

#### A.1. Algorithm Demonstration

We present abstract pseudocode for the LensWalk reason–plan–observe loop in Algorithm 1. Using the notation of Section 3, it shows how each Reasoner plan  $a_t = (o_t, q_t, \mathcal{I}_t, \rho_{o_t})$  over tools  $\mathcal{O} = \{\textit{Scan Search}, \textit{Segment Focus}, \textit{Stitched Verify}, \textit{Finish}\}$  is translated into one or more VLM-visible context groups, how the Observer  $M_o$  is queried on those groups, and how the resulting time-anchored evidence and subject table state  $Sub_t$  are threaded through the multi-turn loop.

#### A.2. Prompt Design

We list the main prompts used by the planning *Reasoner* and the tool-specific *Observer* models. All prompts are shown verbatim as they are instantiated in our implementation.

**Reasoner Prompt.** The dedicated system prompt driving the multi-turn planning Reasoner is given in Fig. 5. It defines the THINK–PLAN–OBSERVE loop, the tool-usage guidelines, and the requirement to terminate via the *finish* tool. We pair it with the user prompt used in our experiments, as in Fig. 6.

**Tool-Call Decision Output Schema.** At each reasoning turn, the Reasoner emits a tool call in the format natively defined by the OpenAI official SDK[18]. Concretely, its raw response contains a `tool_calls` list where each element has the standard structure `{ "type": "function", "function": { "name": 'name of the tool', "arguments": 'tool arguments' } }`. This format would be parsed the `function.name` and `JSON-stringify function.arguments` fields into a Python dictionary before dispatching to execute this tool.

**Observer Prompts (per Tool).** Each video-observation tool is paired with its own Observer prompt. We show one figure of each tool, such as “Scan Search”, “Segment Focus”, and “Stitch Verify”, in Figure 7, 8, 9, respectively.

**Subject Memory Update Prompts.** When subject-level memory is enabled, the agent uses a dedicated memory-update prompt to reconcile subject information from per-step observations with a global subject registry. The concrete memory-update prompt is described in Figure 10.

#### A.3. Tool Schema, Format, and Signatures

We next describe the tool schema exposed to the Reasoner. Tool calls of Reasoner must follow the given JSON schema. It is provided with a tool name, a short natural-language description, and an `arguments` object whose fields include names, types, optionality, and brief semantic descriptions. The full instantiation for the LensWalk tools is shown in Fig. 11.

### B. Evaluation Details

#### B.1. Implementation Details

**Models.** Table 5 lists the models used as Reasoner/Observer in our experiments in Section 4. “Releases” specifies the time of the first public API or open-weight announcement, and “Version” column corresponds to the exact alias used in our experiments.

**LensWalk Agent settings.** We restrict the dialogue to at most 20 tool-calling turns. For the budgets of three observation tools, *Scan Search* samples 30 frames per slice with a default sampling rate of 0.25 fps; *Segment Focus* defaults to 1 fps with at most 32 frames for the requested interval; *Stitch Verify* stitches plan-specified segments, defaulting to 0.5 fps globally and 1 fps per segment with an overall cap of 128 frames. Requested sampling in each tool call is first converted from the provided (interval, fps) pairs to a frame budget and then proportionally rescaled so the total does not exceed each tool’s `max_total_frame`.

**Deployment and Service.** Closed-source models (o3, GPT-4.1/4.1-mini/5) and Qwen3-235B-A22B are queried through their official APIs. All other open-weight models, including Qwen2.5-VL-72B/7B, are served with vLLM Engine on 4×NVIDIA H100 80 GB GPUs.

#### B.2. Benchmark Datasets

We assess LensWalk on six long-form or reasoning-focused benchmarks. **LVBench**[45] provides 1,549 multiple-choice questions drawn from 103 curated hour-long movies (mean duration 67 min); transcripts are unavailable, so the agent relies on frames only. **LongVideoBench-long**[49] contributes 564 MCQs sourced from 188 validation videos in

---

**Algorithm 1:** LensWalk Reason–Plan–Observe loop with Subject Table Memory update.

---

**Input** : Video  $V$ , query  $Q$ , metadata  $V_{\text{info}}$ , tool set  $\mathcal{O} = \{\text{Scan Search}, \text{Segment Focus}, \text{Stitched Verify}, \text{Finish}\}$ , max steps  $T_{\text{max}}$

**Output:** Answer to  $Q$

Initialize history  $L_0 \leftarrow \{Q, V_{\text{info}}\}$ ;  
Initialize subject table  $Sub_0 \leftarrow \emptyset$ ;

**for**  $t = 1, 2, \dots, T_{\text{max}}$  **do**

- $a_t = (o_t, q_t, \mathcal{I}_t, \rho_{o_t}) \leftarrow M_r(Q, V_{\text{info}}, L_{t-1}, Sub_{t-1})$ ;
- if**  $o_t = \text{Finish}$  **then**
  - return**  $q_t$  as final answer;
- $G_t \leftarrow \text{BuildContextGroups}(o_t, \mathcal{I}_t, \rho_{o_t})$ ; // Maps tool-level sampling arguments into VLM-visible context groups
- //  $G_t = \{(I_t^{(k)}, s_t^{(k)})\}_{k=1}^{K_t}$ , where  $I_t^{(k)}$  is the  $k$ -th time interval and  $s_t^{(k)}$  collects its sampling settings (e.g., fps, frame budget)
- //  $K_t = 1$  for *Segment Focus* and *Stitched Verify*;  $K_t > 1$  (parallel slices) for *Scan Search*
- for each**  $(I_t^{(k)}, s_t^{(k)}) \in G_t$  **do**
  - $F_t^{(k)} \leftarrow \text{SampleFrames}(V, I_t^{(k)}, s_t^{(k)})$ ; // Extracts frames in  $I_t^{(k)}$  from  $V$  according to sampling config  $s_t^{(k)}$
  - $y_t^{(k)} \leftarrow M_o(q_t, F_t^{(k)}, I_t^{(k)})$ ;
- $e_t \leftarrow \text{Aggregate}(\{y_t^{(k)}\}_{k=1}^{K_t})$ ; // Simple join: concatenate per-group replies in temporal order into a single observation string
- $L_t \leftarrow L_{t-1} \cup \{(a_t, e_t)\}$ ;
- if** *subject memory enabled* **then**
  - $Sub_t \leftarrow \text{MemoryUpdate}(Sub_{t-1}, a_t, e_t)$ ;
- else**
  - $Sub_t \leftarrow Sub_{t-1}$ ;

**return** *unknown*;

---

the 900–3,600s bucket; this split is the only one where we feed automatic transcripts to the Reasoner, because the benchmark ships ASR JSON files. **Video-MME long**[13] is the long split of Video-MME covers 900 MCQs over 300 videos lasting 30–60 min (mean 41 min) across knowledge, skill, and perception domains; we keep the official multiple-choice format and do not use subtitles for evaluation. For **MMVU**[67], 625 multiple-choice questions from the validation set spanning 40+ academic subjects are used. **Video-MMMU**[17] aggregates 900 MCQs evenly split across the Perception, Comprehension, and Adaptation subsets, each averaging 8.4-minute clips ranging from short 48 s labs to 30+ minute lectures. **EgoSchema**[27] supplies the official validation set of 500 egocentric three-minute clips with 500 multiple-choice questions.

### B.3. Baseline Reproduction

Most baseline results are obtained from their official papers or public leaderboards (e.g., Deep Video Discovery[63], Mr. Video[32], VideoAgent[46], Ego-RI[43]). We

only re-run GPT-4.1, GPT-5 on the considered benchmarks, and Qwen2.5-VL-72B on LongVideoBench-long and EgoSchema-val. For GPT-5, we fix *reasoning\_effort* to `minimal` both inside our agent and in the baseline setting. Frame sampling budgets follow prior works: GPT-4.1/5 consume 256 frames, while local Qwen2.5-VL-72B is limited to 128 frames due to our limited computing resources, matching the max per-forward cap of the *observer* tool.

### C. More Results

In this section, we provide additional qualitative results and trajectory visualizations to complement the main experiments. All case studies are drawn from the long split of the Video-MME benchmark, and all trajectories are produced by the same o3 model acting as both the Reasoner and the Observer. All visualizations follow a unified annotation scheme. Within each reasoning trajectory, key pieces of evidence from visual observations that are crucial for driv-

**You are a helpful assistant designed to answer multi-step questions about video content by sequentially invoking tools for video observation.** Your sole purpose is to reason and act to uncover the correct answer. Operate in a rigorous **THINK → PLAN**

→ **OBSERVE** loop with at most `MAX_CALL` tool calls.

**THINK:** Based on the user’s query and all previous observations, reason step by step about what is known, what is uncertain, and what to do next. Justify the next action by stating the goal, the best tool, chosen time ranges, and the sampling strategy (e.g., higher

FPS for fast actions, lower for context).

**ACT:** Call exactly one video observation tool with the decided parameters. Time ranges and sampling intensity should match the planned evidence needs, avoiding both information loss and redundancy.

**OBSERVE:** Use the returned structured observation to update understanding and continue the next **THINK** cycle.

Continue until you have sufficient evidence, then call `finish` with the final answer.

**Core directives:** (1) Never guess; inspect the video when uncertain. (2) Plan within frame budget. (3) Iterate and refine. (4) Verify and synthesize after multiple related segments. (5) Ground arguments in evidence; do not invent timestamps.

Figure 5. System prompt used for the *LensWalk Reasoner*.

Based on your observations and tool outputs, provide a concise answer that directly addresses the question. Only provide the option’s letter from the given choices as the final answer for multiple-choice questions.

Total video length: `VIDEO_LENGTH` seconds.

Question: `QUESTION_PLACEHOLDER`

Figure 6. User prompt used for *LensWalk Reasoner*.

You are a highly focused visual understanding assistant. Your task is to analyze a short sequence of sparsely sampled video frames from a single segment and assess its relevance to the user’s query. You need to provide a concise summary of the visual content and explain why it is, or is not, related to the query.

**IMPORTANT:** Your scope is strictly limited to the provided frames from this single segment. Your conclusion must be based only on what is visible here. Do not make any assumptions about the rest of the video or events outside these frames. Your response should be a brief, self-contained analysis of this segment’s relevance.

Figure 7. Prompt for the *Observer* in *Scan Search* tool.

You are a meticulous and factual video analysis assistant. Your task is to analyze a set of sparsely sampled frames from a single, continuous time segment of a video. Provide a detailed and objective description that directly answers the user’s query.

**IMPORTANT:** Base your entire analysis strictly on the visual information present in the frames provided. The frames are sampled uniformly. Do not infer or imagine events happening beyond the frames. Your output should be a concise and structured text focused solely on the evidence at hand.

Figure 8. Prompt for the *Observer* in *Segment Focus* tool.

You are an expert video analyst specializing in synthesizing information from multiple, distinct key moments. You will be shown a “stitched” collection of frames sampled from various, potentially non-contiguous time intervals of a video. Your goal is to provide a cohesive analysis that addresses the user’s query, drawing connections or noting changes between the different moments shown.

**IMPORTANT:** The frames represent a curated, sparse, and non-uniform selection. Do not assume what happens in the time gaps between the provided segments. Base your entire analysis strictly on the visual evidence presented, treating each time-stamped group as a distinct snapshot.

Figure 9. Prompt for the *Observer* in *Stitched Verify* tool (*stitched\_observer*).

**Memory role.** You are a specialized assistant responsible for maintaining a `subject_registry` for a video analysis agent. You analyze the latest observation and its subjects, compare them against the existing registry, and produce a consolidated registry only (no narrative summary).

**Input provided.**

- Current subject registry (master list so far).
- History: past tool calls and their results.
- Current turn: latest tool call, its raw output, and a `[new observed subject registry]` extracted from that call.

**Update rules.**

1. **Merge by synthesis, not stacking.** When updating a subject’s description list, synthesize concise, high-value descriptions instead of appending redundant strings. Example: from “man in red shirt” and “man in red shirt, sitting at desk” produce “man in red shirt” and “sitting at desk” (or “man in red shirt (was standing, now sitting at desk)”), not duplicated phrases.
2. **Add new subjects** when they cannot be merged.
3. **Prune to the 15 most recently observed.** After merge/add, keep at most 15 subjects prioritized by most recent appearance, giving current-turn subjects highest priority.
4. **No subjects** ⇒ output an empty object {}.

**Output format (single JSON object).**

```
{
  "updated_subject_registry": {
    "A unique and consistent identifier for a subject": {
      "description": [
        "Key appearance/role/state strings (e.g., 'man in red shirt', 'sitting at
        desk', 'talking to S2')."
      ],
      "appeared_intervals": [
        "[start_sec , end_sec]"
      ]
    }
  }
}
```

Figure 10. Subject Table Memory–Update prompt and required JSON output.

ing the reasoner’s inference are highlighted in **red**. On the temporal axis, sampled contexts covered by different tools are indicated with colored intervals: **green** for **ScanSearch**, **blue** for **Segment Focus**, and **yellow** for **Stitched Verify** respectively.

### C.1. Additional Quantitative Analyses

**Shared-backbone comparison and end-to-end cost.** Table 6 compares LensWalk with prior video agents on LVBench under a shared o3 reasoner. LensWalk requires no offline captioning or index construction, so its end-to-end cost is substantially lower than retrieval-based agents once

Table 5. Models used in our experiments.

Organization	Model	Release	Version
OpenAI	GPT-4.1[30]	2025-04	gpt-4.1-2025-04-14
	GPT-4.1-mini[30]	2025-04	gpt-4.1-mini-2025-04-14
	o3[29]	2025-04	o3-2025-04-16
	GPT-5[31]	2025-08	gpt-5-2025-08-07
Alibaba	Qwen3-235B-A22B[55]	2025-04	qwen3-235b-a22b-instruct
	Qwen2.5-VL-72B[2]	2025-02	qwen2.5-vl-72b-instruct
	Qwen2.5-VL-7B[2]	2025-02	qwen2.5-vl-7b-instruct
Z.ai	GLM-4.7[59]	2025-08	glm-4.7
DeepSeek	DeepSeek-V3[10]	2025-03	DeepSeek-V3-0324

preprocessing is counted. It also uses 14–32× fewer frames per query while outperforming MR, Video and VideoAgent. Compared with DVD, LensWalk trades some accuracy for zero preprocessing and much lower total cost.

**Efficiency and adaptive budget allocation.** Table 7 shows that LensWalk does not impose a fixed reasoning overhead. It typically converges in 2.6–2.8 steps on short or simple benchmarks, but increases both interaction steps and frame usage as videos become longer or questions demand more reasoning. Together with Table 6, this indicates that LensWalk improves efficiency not only by avoiding expensive full-video preprocessing, but also by allocating observation budget only when additional evidence is needed.

**Additional reasoners and a static extracted-frames baseline.** Table 8 separates the benefit of active observation from simply reusing better frames. Feeding the union of frames visited by LensWalk to a single-pass VLM yields only modest gains (+0.8 to +2.6 points), indicating that the improvement does not come from frame selection alone. The same table also shows that LensWalk remains effective with additional reasoners such as GLM-4.7 and DeepSeek-V3, indicating that the framework is not tied to a single planner.

## C.2. Behaviour Analysis

Section 4.4 has introduced the taxonomy of tool-call behavior patterns; this section provides a detailed analysis of representative instances for each behavior.

**Direct Inquiry.** Direct Inquiry is the simplest tool-use pattern, where the agent can answer with only one or two tool calls. It is mainly used for straightforward temporal-localization questions such as “What happens at the beginning of the video?” or “What occurs between 10:21 and 10:25?”, as well as questions about the overall video type.

As shown in Fig. 12, the agent only queries the ending interval for a specific option and then finds that it is correct.

**Progressive Zoom-in.** The Progressive Zoom-in pattern begins with a coarse global scan and gradually narrows the candidate segments. Using approximate information from earlier steps, the agent then conducts increasingly fine-grained observations within a smaller temporal window. As illustrated in Fig. 13, the model first performs a global *ScanSearch* to locate intervals containing key visual cues mentioned in the question (e.g., the German team and the diving scene), and then progressively restricts its focus around the queried “10m high platform diving competition”, eventually arriving at the correct answer.

**Integrative Verify.** In the Integrative Verify pattern, after several preliminary observations, the agent formulates an explicit hypothesis and then tests it by jointly inspecting multiple related segments. As illustrated in Fig. 14, given a query such as “Is there any segment where they say an earthquake occurred?”, the model systematically reviews each clip for mentions or visual evidence of an earthquake and, upon finding none, concludes that no such event is reported.

**Strategic Reflection.** Strategic Reflection is a higher-level reasoning pattern in which the agent realizes that its current observational strategy is failing, then backtracks to explore broader temporal regions before refining again, rather than over-focusing on an uninformative segment. As illustrated in Fig. 15, once the model recognizes that the subtitles in the initial interval do not explain the game rules, it strategically shifts to another part of the video, enabling it to observe the optimal game strategy and ultimately obtain the correct answer.

Table 6. Shared-backbone comparison on LVBench using o3 as the reasoner. We report accuracy, online inference time, offline preprocessing time, and average frames consumed per query. Retrieval-based agents require dense pre-processing of the full video, while LensWalk does not.

Method	Acc. (%)	Online Inference (s)	Offline Preprocessing (s)	Avg. Frames / Query
o3	57.1	38.9	0	256
LensWalk	68.6	190.35	0	290.3
DVD	74.2	153.3	2180.4	8202 (2 fps)
MR. Video	65.5	326.2	4135.2	9227 (2.25 fps)
VideoAgent	64.1	200.5	1131.3	4101 (1 fps)

Source. Results for methods other than o3 and LensWalk are taken from DVD (<https://openreview.net/forum?id=oQYq9L1NVT>).

Table 7. Adaptive scaling of LensWalk (o3/GPT-4.1) across benchmarks with different video lengths and reasoning demands. Gains are relative to the single-pass GPT-4.1 baseline.

Benchmark	Split	Acc. (%)	Gain	Avg. Video Length (s)	Reasoning-Intensive	Avg. Steps	Avg. Frames Used
Video-MME	Short	82.3	+1.7	80.7	No	2.8	89.7
	Medium	79.9	+8.3	515.9	No	4.2	233.0
	Long	70.0	+6.9	2466.7	No	6.8	387.1
EgoSchema	—	74.8	+2.6	180.0	No	2.6	89.7
Video-MMMU	—	77.1	+9.7	506.2	Yes	4.8	178.4
MMVU	—	80.9	+2.0	51.4	Yes	3.1	58.5

Table 8. Additional model recipes and a static extracted-frames baseline on Video-MME (long split). Each group header shows the corresponding single-pass observer baseline. *Extracted Frames* feeds the union of frames visited by LensWalk under the same 256-frame budget into a single-pass VLM.

GPT-4.1 Observer (baseline: 63.1)		GPT-4.1-mini Observer (baseline: 59.4)		o3 Observer (baseline: 64.7)	
Reasoner	Acc. (%)	Reasoner	Acc. (%)	Reasoner	Acc. (%)
GLM-4.7	68.3 (+5.2)	GLM-4.7	66.0 (+6.6)	o3	71.4 (+6.7)
DeepSeek-V3	66.4 (+3.3)	DeepSeek-V3	63.3 (+3.9)	—	—
<i>Extracted Frames</i>	65.0 (+1.9)	<i>Extracted Frames</i>	62.0 (+2.6)	<i>Extracted Frames</i>	65.5 (+0.8)

**Scope Partitioning.** Scope Partitioning adopts a divide-and-conquer strategy: the agent partitions the video into coarse segments and sequentially checks whether each region contains clues worth closer inspection. As shown in Fig. 16, it first scans the intervals 0–600s, 600–1200s, and 1200–1800s to locate the position of the clue about the third set, then identifies 1200–1800s as the most promising range, performs more fine-grained observations there, and finally derives the correct answer.

**Static Repetition.** In contrast to *Strategic Reflection*, *Static Repetition* denotes a rigid mode in which the model becomes trapped in an unproductive strategy. Rather than exploring alternative temporal regions or adjusting its observation queries, it repeatedly issues nearly identical observe calls with the same intent on the same segment, even after multiple failed attempts to find supporting evidence, as shown in Fig. 17. This fixation on a single interval prevents the model from discovering the correct answer elsewhere in

the video.

### C.3. Failure Modes and Analysis

In this section, we illustrate a few failure modes observed during our evaluation. Analyzing these error patterns provides valuable insights into the limitations of current Large Language Models (LLMs) acting as reasoning planners and highlights critical directions for future active video agent design. We identify four primary failure patterns:

**Premature Conclusion.** The agent terminates the reasoning loop too early (often within 1–2 turns) upon encountering the most frequent cue “pyramid complex” in the initial observation, as shown in Fig. 18. By failing to validate this cue against potential distractors or cross-reference it with other video segments, the model exhibits unwarranted overconfidence. This failure mode suggests that future agents require a stronger internal “critic” or verification mechanism to balance token efficiency with rigorous hypothesis testing.

**Evidence Dilution.** Conversely, excessive observation can degrade performance. We observe cases, as in Fig. 19, where the agent discovers strong, decisive evidence in early turns e.g. “STRENGTH UNDER PRESSURE” or “MENTAL ABILITY”, to the correct choice “Mental ability and vast experience”, but continues to explore irrelevant segments. As the context window fills with “loose” or noisy information, the strong initial evidence gets diluted. The agent finally succumbs to recency bias or confusion, hallucinating an answer based on the latest weak observations. This underscores the importance of active memory management—agents must learn not just when to seek information, but when to prune irrelevant context to maintain reasoning clarity.

**Persistent Ambiguity.** The agent may persistently search along an uninformative trajectory and never obtain decisive visual evidence, ultimately being forced to give an ambiguous, semantics-only guess rather than a visually grounded answer. As shown in Fig. 20, the agent keeps focusing on related subtitles instead of frame-wise visual changes, misses the key opening event, and therefore predicts incorrectly.

**Static Repetition.** Finally, as noted in the behavioral analysis, the agent may fall into a degenerate loop, repeatedly issuing identical tool calls for the same timestamp without adjusting its parameters. This reflects a failure in the agent’s meta-cognition or state tracking; the planner fails to recognize that its current strategy is unproductive, highlighting the need for mechanisms that detect stagnation and trigger strategic shifts (e.g., from *Segment Focus* to *Strategic Reflection*).

```

[
  {
    "type": "function",
    "function": {
      "name": "segment_observer",
      "description": "Probe one interval with an MLM under specified
sampling.",
      "parameters": {
        "type": "object",
        "properties": {

          "interval": {
            "type": "object",
            "properties": {
              "start_sec": { "type": "number", "minimum": 0 },
              "end_sec": { "type": "number", "exclusiveMinimum": 0 }
            },
            "required": ["start_sec", "end_sec"]
          },
          "query": { "type": "string" },
          "fps": { "type": "number", "exclusiveMinimum": 0, "default": 1 },

          "max_total_frames": { "type": "integer", "minimum": 1, "default":
32 }
        },
        "required": ["interval", "query"]
      }
    },
  },
  {
    "type": "function",
    "function": {

      "name": "stitched_observer",
      "description": "Probe multiple segments, stitch frames, then answer
one question.",
      "parameters": {
        "type": "object",
        "properties": {
          "segments": {
            "type": "array",
            "minItems": 1,
            "items": {

              "type": "object",
              "properties": {
                "start_sec": { "type": "number", "minimum": 0 },
                "end_sec": { "type": "number", "exclusiveMinimum": 0 },
                "fps": { "type": "number", "exclusiveMinimum": 0,
"default": 1 }
              },
              "required": ["start_sec", "end_sec"]
            }
          },
          "query": { "type": "string" },
          "global_interval": {
            "type": "object",
            "properties": {

```

Figure 11. JSON Tool Schema used for *Reasoner* of LensWalk (part 1).

```

        "start_sec": { "type": "number", "minimum": 0 },
        "end_sec":   { "type": "number", "exclusiveMinimum": 0 }
    },
    "required": ["start_sec", "end_sec"]
},
"fps": { "type": "number", "exclusiveMinimum": 0, "default": 0.5 },
"max_total_frames": { "type": "integer", "minimum": 1, "default":
128 }
},
"required": ["segments", "query"]

}
},
{
"type": "function",
"function": {
"name": "scan_observer",
"description": "Scan a global interval by slices and summarize each
slice.",
"parameters": {

"type": "object",
"properties": {
"global_interval": {
"type": "object",
"properties": {
"start_sec": { "type": "number", "minimum": 0 },
"end_sec":   { "type": "number", "exclusiveMinimum": 0 }
},
"required": ["start_sec", "end_sec"]
},

"num_slices": { "type": "integer", "minimum": 1 },
"slice_duration_sec": { "type": "number", "exclusiveMinimum": 0 },
"query": { "type": "string" },
"fps": { "type": "number", "exclusiveMinimum": 0, "default": 0.25
},
"max_total_frames": { "type": "integer", "minimum": 1, "default":
180 }
},
"required": ["global_interval", "query"]
}

}
},
{
"type": "function",
"function": {
"name": "finish",
"description": "Return the final answer and end the dialog.",
"parameters": {
"type": "object",
"properties": {

"answer": { "type": "string" }
},
"required": ["answer"]
}
}
}
]

```

Behaviour Mode: "Direct Inquiry"



0.0 2190.1



Question: What does the video featuring?  
 A. Chelsea winning the FA Cup after a decisive match.  
 B. A regular season Premier League match between Chelsea and Liverpool.  
 C. Liverpool winning the FA Cup against Chelsea in a penalty shootout.  
 D. A pre-match analysis of the FA Cup final between Chelsea and Liverpool.

Turn 1:  
 <Tool>: Segment Focus, query = "What is seen happening? Are players lifting a trophy or are there pundits in studio?"  
 <Observation>: "Throughout 2145-2175 s, football players wearing red kits stand on a stadium balcony emblazoned with 'FA CUP WINNERS 2022'. Several of them successively raise a large silver cup with red-white ribbons while teammates cheer, pose and pass the trophy along. No studio set or pundits are visible—only on-field victory celebrations followed by a plain red 'CLICK TO SUBSCRIBE' end-card (2175-2190 s)."

<tool call args>:  
 interval=[2145, 2190],  
 fps=1


0.0 2145.0 2190.1



The final answer should be "C".

Figure 12. Example of Direct Inquiry behavior.

## Behaviour Mode: "Progressive Zoom-in"



0.0 2302.7

**Question:** In the video, what is the problem with the German team's third jump in the women's synchronized 10m high platform diving competition?

A. They were missing a turn in the air.  
 B. They made big splashes compared to clean entries.  
 C. They entered the water feet first.  
 D. None of the above.

**Turn 1:**  
 <Tool>: **Scan Search**, query = "Find the segments with the women synchronized 10m platform diving, especially the German team (Germany/GER flag or text). Identify the third dive of the German pair and any visible problem on that dive."  
 <Observation>: "... [1320-1440s] GER pair on 10 m platform; overlay shows *Round 3/5 (55.68)* → confirms **3rd dive**. Dive 1380-1400s (no entry seen), **so fault like splash/feet-first not confirmed**..."

0.0 120.0 240.0 ... 2160.0 2280.0 2302.7

**Turn 2:**  
 <Tool>: **Stitch Verify**, query = "Analyze the German pair's Round 3 dive. Did they miss a rotation/turn? Did they enter feet-first or hands/head-first? How big were the entry splashes relative to typical clean entries?"  
 <Observation>: "... **Insufficient visual evidence**: No frames show **GER Round 3 dive** or visible "**GER Round 3/5**" overlay. Thus, **rotation, entry orientation, and splash size** cannot be determined. Need frames covering **GER's Round 3 approach, rotation, entry**, and overlay for verification...."

0.0 120.0 240.0 1386.0 1400.0 1402.0 1410.0 2302.7

**Turn 3:**  
 <Tool>: **Segment Focus**, query = "Find the German pair's Round 3 dive in this window. Confirm GER and Round 3/5 via overlay, then describe their mid-air rotation and, crucially, the water entry: Was there a missed turn? Did they enter feet-first or hands/head-first? Were the splashes large compared to a clean entry?"  
 <Observation>: [1375-1415s] GER Round 3 dive: synchronized takeoff → tuck → **head-first entry with big splash**; overlay confirms *Round 3/5 55.68 pts*, no missed turn.

0.0 1365.0 1415.0 2302.7

The final answer should be "B".

Figure 13. Example of **Progressive Zoom-in** behavior.

## Behaviour Mode: “Integrative Verify”



0.0 2713.8

**Question:** According to the video, which of the following news items did NOT appear in this news segment?

- A. An earthquake occurred in the United States.
- B. Potential community spread of measles in Canada.
- C. Fire destroys First Nation's only school.
- D. Josh Liendo swims into the history books.

**Turn 1:**

<Tool>: **Scan Search, query** = "Indicate if within this slice, the news covers: 1) US earthquake, 2) Measles community spread in Canada, 3) Fire destroys First Nation's only school, 4) Josh Liendo swimming record."

<Observation>: "[0, 180.919]: Measles spread + brief Liendo; no earthquake or school fire. [180.919, 361.838]: Only measles coverage..... [2351.95, 2532.87]: Continued First Nation school-fire story. [2532.87, 2713.79]: Pet-psycho promo + police herding goats. And the studio anchor providing narration. **None of the four queried topics—US earthquake, measles spread in Canada, First Nation school fire, or Josh Liendo swimming record—are depicted or referenced in these frames.**"

0.0 180.9 361.8 ... 2351.9 2532.8 2713.8

**Turn 2:**

<Tool>: **Scan Search, query** = "For each slice, indicate if there is any coverage of an earthquake."

<Observation>: "[0, 226.149]: Protest, measles, skating, vaccines, interviews; **no earthquake coverage.**... [904.595, 1130.74]: Snowstorm, wildfire intros, officials, factory fire; **no earthquake coverage.**... [2487.64, 2713.79]: Park scene, pedestrians, pet-psycho promo, goats, anchor narration; **no earthquake coverage.**"

0.0 226.1 452.3 ... 2261.4 2487.6 2713.8

**The final answer should be "A".**

Figure 14. Example of **Integrative Verify** behavior.

## Behaviour Mode: "Strategic Reflection"



0.0

2794.8



**Question:** In the first game of the video's early part, what is the key to winning this game of building the tallest tower?  
 A. Contestants use their bodies to support the cans, achieving a taller tower.  
 B. Contestants do not knock down the already stacked can tower and use this tower to achieve a greater height.  
 C. Contestants knock down the already stacked can tower and use a new method to build a taller tower.  
 D. Cannot be determined.

### Turn 1:

<tool call args>  
 interval=[0, 800],  
 fps=0.25,  
 num\_slices=16

<Tool>: **Scan Search, query** = "Describe what is happening, focusing on if contestants are building towers with cans and if they knock down or support with their bodies."  
 <Observation>: "... [700-750s, 750-800s] shows blind-folded contestants taking turns in a "tallest tower" game, repeatedly **bumping and toppling a pyramid of cans**, then trying to **steady and restack it with their hands and bodies**—confirming that they **knock down and physically support the tower**...."

0.0 ..... 2794.8  
 50.0 ... 800.0

### Turn 2:

<tool call args>  
 interval=[550, 565],  
 fps=1

<Tool>: **Segment Focus, query** = "What does the task card say about building the tallest tower with cans? Look at any visible text or listen to spoken reading. Specifically, what instructions or key rule is stated?"  
 <Observation>: "Each contestant sits at the table, reads the white task card aloud and looks at the single tin can placed before them. Although the full text on the card is not readable in the frames, every reader reaches and reacts to the same highlighted instruction: **the card's clear rule is to "build the tallest tower with cans – the tallest tower wins."**

0.0 ..... 2794.8  
 550.0 565.0

### Turn 3:

<tool call args>  
 interval=[418, 440],  
 fps=2

<Tool>: **Segment Focus, query** = "What are they discussing about the tall object on screen? Any mention of how the winner achieved the tallest tower? Did they mention supporting with bodies or not knocking down?"  
 <Observation>: "There is **no visual clue** (e.g., subtitles, sketches, reenactments, or body-stacking shots) that refers to how a winner built the tallest tower, nor any sign of people supporting it with their bodies or of preventing it from being knocked down. The frames provide **no explicit evidence** that such topics are mentioned."

0.0 ..... 2794.8  
 418.0 440.0

### Turn 4:

<tool call args>  
 interval=[800, 1000],  
 fps=0.25,  
 num\_slices=8

<Tool>: **Scan Search, query** = "Describe what is happening, focusing on if contestants are building towers with cans and if they knock down or support with their bodies."  
 <Observation>: "... [900-925s, 925-950s, 950-975] shows contestants in the task room carefully stacking metal cans into a tall pyramid. The tower remains upright through most of the segment, with only a few cans rolling away. The visible footage clearly indicates that contestants **successfully keep the tower standing and steady it with their hands**, suggesting that **maintaining tower stability—rather than body support or new methods—is the key observed behavior**...."

0.0 ..... 2794.8  
 800.0 ... 1000.0

### Turn 5:

<tool call args>  
 segments=  
 ((1050, 1100), ... ,  
 (1150, 1180)), ... ,  
 fps=1

<Tool>: **Stitch Verify, query** = "Describe what is happening, focusing on if contestants are building towers with cans and if they knock down or support with their bodies."  
 <Observation>: "... he **left the massive one that was already there**'... 1080-1200 s: Later players topple the pyramid and scramble on the floor trying to restack cans; **their finished heaps reach only shin-height**. 1150-1200 s studio chat confirms Sam's untouched pyramid was, by far, the tallest. Key to winning was **"not knocking the existing tower over"**; **no body-support trick was used** – Sam simply treated the pre-built pyramid as his tower and walked away, leaving it standing."

0.0 ..... 2794.8  
 1050.0 1180.0



The final answer should be "B".

Figure 15. Example of Strategic Reflection behavior.

## Behaviour Mode: "Scope Partitioning"



Figure 16. Example of Scope Partitioning behavior.

## Behaviour Mode: "Static Repetition"



0.0 2378.2

**Question:** After #41 completed his free throws, how many points was the team the author was rooting for ahead or behind the other team?  
 A. 6 points ahead of the other team.  
 B. **Trailing the other team by 12 points.**  
 C. 12 points ahead of the other team.  
 D. 12 points ahead of the other team.

**Turn 1:**  
 <Tool>: **Scan Search, query** = "Identify moments where the scoreboard is visible and shows the current score in the basketball game."  
 <Observation>: "...[713.4, 951.2]: At the start of 856–903 s, the tabletop scoreboard is clearly visible. It shows **game clock 15:40**, and the score is **32 (right team) to 27 (left team)**...[951.2, 1189]: The large wall-mounted scoreboard is readable during several shots: during loose-ball action near **951 s**, half-court play around **1000 s**, and a free-throw after **1046 s**. A portable scoreboard is also readable around **1094 s** and again near **1189 s**, each confirming the **live running score**."

0.0 ..... 237.8 ..... 475.6 ..... 1982.6 ..... 2148.4 ..... 2378.2

**Turn 2:**  
 <Tool>: **Segment Focus, query** = "Describe what happens, especially any free throws by player #41 and the scoreboard after they finish."  
 <Observation>: "At 1298-1312 s, **orange-uniformed player #41 is shown standing at the free-throw line with the ball**, dribbling and taking his shots while both teams are lined up along the lane. Immediately afterward play is readied to resume and the courtside electronic scoreboard that comes into clear view (≈1308-1312 s) reads **30 for the team in the white jerseys and 32 for the team in the orange jerseys**, indicating his free throws were successful and have given the orange team a two-point lead."

0.0 ..... 1278.0 ..... 1340.0 ..... 2378.2

**Turn 3:**  
 <Tool>: **Segment Focus, query** = "What are the scores displayed on the scoreboard before and after #41's free throws? Which side corresponds to which team (colors)?"  
 <Observation>: "At 1298-1308 s, just before #41 (navy-and-orange uniform) begins his trip to the line, **the tabletop scoreboard on the far right shows 37–32**. The left side (37) belongs to the navy/orange team, while the right side (32) belongs to the white/blue team. In After #41 finishes shooting (frames 1308-1318 s), the same board now reads 38–32. Only the left total has risen, confirming the point(s) were credited to #41's navy/orange side."

0.0 ..... 1288.0 ..... 1320.0 ..... 2378.2

**Turn 4:**  
 <Tool>: **Segment Focus, query** = "What is the scoreboard score before #41 takes his free throws?"  
 <Observation>: "**No frame in the supplied 1288-1298 s interval shows a readable view of the scoreboard**; therefore, the score immediately before #41 steps to the free-throw line cannot be determined from the provided images."

0.0 ..... 1288.0 ..... 1298.0 ..... 2378.2

...Zoom in × 15...

**Turn 20:**  
 <Tool>: **Segment Focus, query** = "Look for scoreboard and read score before #41 free throws, between 1280-1295 seconds."  
 <Observation>: "Across all frames supplied from 1280 s to 1295 s, no image offers a clear, legible view of any scoreboard; consequently, **the score immediately before player #41's free-throw attempts cannot be determined from the visual evidence provided**."

0.0 ..... 1280.0 ..... 1295.0 ..... 2378.2

**Arrive the max tool call turns, there is no sufficient evidence to answer the question.**

Figure 17. Example of **Static Repetition** behavior. The notation "zoom in × n" indicates that several fine-grained but low-impact intermediate inspections are omitted from the visualization for brevity.

## Failure Type: "Premature Conclusion"



0.0

2664.0



**Question:** What specific evidence is presented to support the claim that the Maya in the Yucatan were more advanced than previously believed?

- A. The discovery of a massive, previously unknown pyramid complex.
- B. The intricate carvings on the vault stones found in the pyramid.
- C. The detailed analysis of dental plaque revealing a diverse diet.
- D. The construction of the "Stairway to Heaven" estate with its elaborate architecture.

**Turn 1:**

<Tool>: **Segment Focus** (interval=[0, 2664], fps=0.25, num\_slice=20 query = "Identify if this slice shows evidence that the Maya in the Yucatan were more advanced than previously believed and what specific evidence is discussed (pyramid complex discovery, intricate carvings on vault stones, dental plaque analysis, or Stairway to Heaven estate)")

<tool call args>  
interval=[0, 2664],  
fps=0.25,  
num\_slice=20

<Observation>: " [0, 133.2]: LiDAR + stepped pyramid → **pyramid complex**. [133.2, 266.4]: Pyramid excavation → **pyramid complex**. [666, 799.2]: Chamber carvings → **vault-stone carvings**. [799.2, 932.4]: Terraced hillside → **Stairway to Heaven estate**. [932.4, 1065.6]: Carved fragments → **vault-stone carvings**. [1065.6, 1198.8]: Widespread temples → **pyramid complexes**. [1198.8, 1332]: Monumental stairway → **Stairway to Heaven estate**. [1864.8, 1998]: Pyramid-centered city → **pyramid complex**. [1998, 2131.2]: Courtyard + stepped buildings → **pyramid complex**. [2131.2, 2264.4]: Exterior masks/mosaics → **vault-stone carvings**. [2264.4, 2397.6]: Stone terraces + stairway → **Stairway to Heaven estate**. [2397.6, 2530.8]: Forest/interviews → no evidence. [2530.8, 2664]: Exterior geometric/figural carvings → **vault-stone carvings**.

0.0 133.2 266.4 ... 2397.6 2530.8 2664.0



The final answer should be "A". ❌

Figure 18. Example of Failure mode: **Premature Conclusion**.

## Failure Type: "Evidence Dilution"



The final answer should be "B". ❌

Figure 19. Example of Failure mode: **Evidence Dilution**. The notation "zoom in × n" indicates that several fine-grained but low-impact intermediate inspections are omitted from the visualization for brevity.

## Failure Type: “Persistent Ambiguity”



Figure 20. Example of Failure mode: **Persistent Ambiguity**.