Hierarchical Long Video Understanding with Audiovisual Entity Cohesion and Agentic Search

Supplementary Material

The supplemental material contains additional implementation details as well as more results and discussions.

A. Entity-Centric Re-Captioning

After consolidation, each canonical entity $\tilde{e}_j$ is associated with a global description and a set of linked segments Q_j. During retrieval, incorporating all linked segments of top-K_1 entities can be computationally expensive and may introduce query-irrelevant noise. Such noise can weaken embedding-based matching; while LLM-based re-ranking may mitigate this issue, it incurs higher computational cost. To address this, we further introduce an *entity-centric re-captioning* process during offline construction. For each linked segment i and entity $\tilde{e}_j$, we generate a focused description $\tilde{C}^t_{i,j}$ using an LLM that summarizes the entity's appearance, actions, and events within the segment, while excluding irrelevant context. The final entity database contains both canonical entities and fine-grained entity-segment descriptions: $\tilde{\mathcal{E}} = \{\tilde{\mathcal{E}}_g\,;\,\tilde{\mathcal{E}}_e\}$, where $\tilde{\mathcal{E}}_e = \{\tilde{C}^t_{i,j} | j = 1, 2, \ldots, J, i \in Q_j\}$. During retrieval, we first match entities in the embedding space of $\tilde{\mathcal{E}}_g$, then re-rank linked segments using similarity between the query and $\tilde{C}^t_{i,j}$, selecting the top-K_2 segments for precise grounding. This design balances retrieval precision and computational efficiency, avoiding excessive LLM overhead. In our implementation, K_1 and K_2 are set to 20 and 16, respectively.

B. Multi-Granularity Tools

Here we present details of the whole tool set denoted by

$$\mathcal{T} = \{T_{\text{scene}}, T_{\text{caption}}, T_{\text{visual}}, T_{\text{entity}}, T_{\text{inspect}}\}. \tag{8}$$

Global Scene Browse. This tool T_{scene} supports coarse-grained navigation and scene localization along the video timeline. Given a user query q and the scene collection $D = \tilde{\mathcal{S}}$, it identifies and summarizes the most relevant scenes with an LLM, returning their storyline and corresponding timestamps τ'. The agent tends to invoke this tool for complex or ambiguous queries involving multiple events or temporal dependencies.

Segment Caption Search. This tool T_{caption} performs fine-grained text-based retrieval within specified temporal ranges. Given the user query q and the segment database $D = \tilde{\mathcal{C}}$, the tool retrieves the most semantically relevant segment descriptions r along with their associated time spans τ'. This is achieved through cosine similarity matching between the query embedding and pre-computed caption embeddings for all video segments, ensuring efficient and accurate retrieval of localized content.

Segment Visual Search. To capture visual cues that may be overlooked in textual descriptions, the *Segment Visual Search* tool T_{visual} complements T_{caption}. While the latter relies on text embeddings, T_{visual} leverages cross-modal embeddings generated by the UNITE framework [11]. This design enables retrieval driven by rich visual semantics aligned with the query, ensuring that visually salient details are incorporated into the search process.

Entity Search. This tool T_{entity} supports high-level, entity-centric retrieval across large temporal ranges. Given an entity-related query q, the tool first retrieves the top-K_1 most relevant entities from database $\tilde{\mathcal{E}}_g$ based on their descriptions in the pre-computed embedding space. For all segments linked to these entities, it then performs a second-stage reranking to select the top-K_2 most relevant segments from the entity-centric database $\tilde{\mathcal{E}}_e$, using the same query. Finally, the tool applies entity-aware re-captioning to the retrieved segments, generating a coherent response r enriched with precise timestamps τ'.

Inspection Tool. This tool T_{inspect} provides fine-grained temporal inspection to support detailed reasoning. It consists of two complementary modules: *Clip Caption Inspect* ($T^{\text{tex}}_{\text{inspect}}$) and *Visual Inspect* ($T^{\text{vis}}_{\text{inspect}}$). The $T^{\text{tex}}_{\text{inspect}}$ examines coarse textual descriptions to determine what occurred during a specified time span. For example, for a query such as *"What does the protagonist do after he jumps down the stairs?"*, the tool inspects subsequent time ranges to identify the protagonist's next actions after locating the event *"he jumps down the stairs"* in previous iteration. The $T^{\text{vis}}_{\text{inspect}}$ tool leverages a VLM to perform precise visual verification within a given time range. Due to frame limits of the VLM, this inspection focuses on short intervals, ensuring accurate visual grounding for fine-grained queries.

C. Other Technical Details

We leverage agglomerative clustering with a threshold of 0.4 for entity clustering stage. A chunk size of 24 with an overlap of 3 is employed for scene segmentation. Due to the API frame limit, the *Ours (2 fps)* variant generates captions for two 15-second sub-segments, each sampled with 30 frames, and concatenates them into a single description.

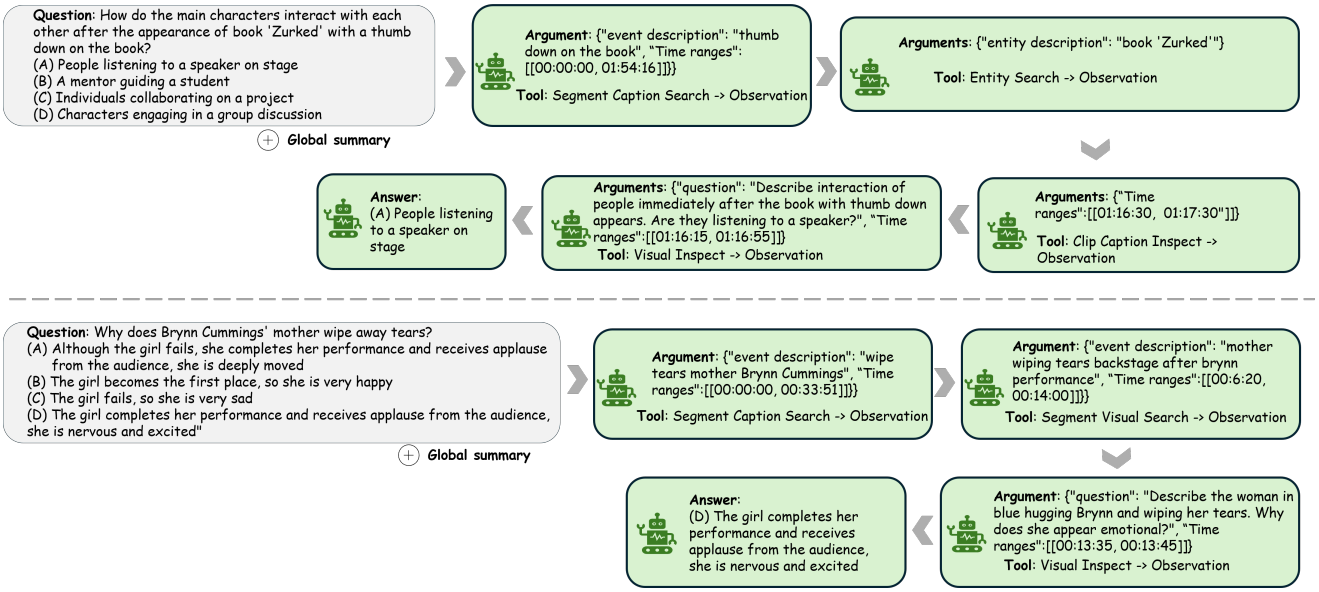

Figure 6. Cases with multiple reasoning steps.

D. Additional Case Study

As shown in Figure 6, we present more complex question-answer cases that involve multiple reasoning steps and tool calls, further demonstrating the effectiveness of our proposed pipeline. For the interaction question involving *the appearance of the book "Zurked" with a thumbs-down*, the agent first performs a coarse segment-level caption search, then progressively narrows the temporal window via entity search. After validating the candidate clips with caption inspection in a long time range, it applies fine-grained visual inspection to extract the correct interaction outcome. For the reasoning-oriented question *"Why does Brynn Cummings' mother wipe away tears?"*, the agent again initiates with segment caption search tool, but due to incomplete textual matches, it escalates to segment-level visual search to gather more reliable evidence. Once the event is localized, the agent conducts a targeted visual inspection to infer the emotional cause behind the mother's reaction.

E. Efficiency

Our semantic-consistent hierarchy enables more efficient navigation, achieving higher accuracy with fewer reasoning steps and less runtime compared with DVD, as shown in Table 4.

F. Proprietary models

Our API versions are GPT-4.1 (2025-04-14) and o3 (2025-04-16). Variance of three runs on LVBench is 0.149, which demonstrates the robustness of our method. Furthermore, Our method achieves an competitive accuracy of 75.8% on LVBench using open-source models (DeepSeek-R1-0528 for reasoning + Qwen3-VL-32B-Instruct for visual inspection).

Table 4. Comparison of average number of iterations and runtime (second per query).

Methods	Iteration	Runtime (s)
DVD	7.6	151.0
Ours	4.2	98.7

G. Prompt for the Planner

We present the planner's prompt for agentic search in Table 5. This prompt guides the planner in selecting the most appropriate tools to search from the hierarchical database and determining what information to request at each reasoning step, thereby enabling systematic information gathering and progressively moving toward the final answer.

Table 5. The agentic search prompt structure. The prompt is divided into four distinct sections: goal, tools, tool preferences and hints.

Module	Prompt Content
GOAL	You will be given a set of tools to assist you exploring the video, understanding it and reasoning the answer. Please follow the THINK → ACT → OBSERVE loop: • THOUGHT: Reason step-by-step about what question to ask and which tool to call next. • ACTION: Call exactly one tool that moves you closer to the final answer. • OBSERVATION: Summarize the tool call's output. Continue the loop until the user's query is fully resolved, then end your turn with the final answer.
TOOLS	Here are tools you can use to reveal your reasoning process whenever the provided information is insufficient: • **global_scene_browse_tool**: for scene-related query to explore scenarios, temporal orders, and contextual structure in a rough manner without precise details (e.g., first appearance, second song, third collision). • **entity_search_tool**: for entity-related information retrieval, finding important subjects involved in events. • **clip_caption_search_wtime_tool**: to search from rough captions and audio transcriptions of local clips within a list of time ranges related to a query. If you want to search from the whole video, use the whole video time range. • **clip_visual_search_wtime_tool**: to search from visual features of local clips within a list of time ranges (list[tuple[HH:MM:SS, HH:MM:SS]]) related to a query. If you want to search from the whole video, use the whole video time range. This tool may provide more detailed information as a supplement to **clip_caption_search_wtime_tool**. • **clip_caption_inspect_tool**: to extract rough captions and audio transcriptions of local clips within any list of time ranges (list[tuple[HH:MM:SS, HH:MM:SS]]). This tool is suitable for further inspecting what happened in a time range before or after some events happened. • **visual_inspect_tool**: to extract details from local visual clips within a narrow list of time ranges that covers less than 50 seconds to answer a question or retrieve query-related details. When the time ranges cover over 50 seconds, first use **clip_caption_inspect_tool** to get a rough context. • **finish**: Once you believe you have found the answer, you can call the **finish** tool with an answer.
TOOL PREFERENCES	• When no context is given, call **global_scene_browse_tool** for scene-related queries to get an overview of related context and timelines, or call **clip_caption_search_wtime_tool** for event-related queries to get a rough context and timelines, or call **entity_search_tool** for entity-related queries. • When you cannot locate the needed context from scene or entity tools, use **clip_caption_search_wtime_tool** or **clip_visual_search_wtime_tool** to expand your search. • If the retrieved material by **clip_caption_search_wtime_tool** lacks relevant contexts, further call **clip_visual_search_wtime_tool** for more fine-grained search. • If the retrieved material lacks precise, question-relevant detail (e.g., an unknown name, count) or you are uncertain of an answer after searching, call **clip_caption_inspect_tool** or inspect frames with **visual_inspect_tool** with a list of time ranges to take a closer look. • After locating an answer in the script, always make a **CONFIRM** with **visual_inspect_tool** query.
HINTS	• Before giving the final answer, confirm critical visual or numeric facts with **visual_inspect_tool**. • If you call **clip_caption_search_wtime_tool** in three consecutive times but still cannot find useful information to answer the question, please try **clip_visual_search_wtime_tool** to get more detailed information. • You have at most 10 iterations of THINK→ACT→OBSERVE; plan strategically. Avoid redundant information retrieval steps. • To make a good plan to questions that need complex reasoning, sometimes you need to first ask some other related contexts instead of directly asking the target question, • For questions that need counting the number of times an event occurs over time, call **global_scene_browse_tool** first. If you are uncertain about its answer, please search for related events or subjects without counting first to find all related information, then do the counting based on observations. • Your final answer must be concise and directly address the question.