

Deep Video Discovery : Agentic Search with Tool Use for Long-form Video Understanding

Xiaoyi Zhang^{*1} Zhaoyang Jia^{*2†} Zongyu Guo¹
 Jiahao Li¹ Bin Li¹ Houqiang Li² Yan Lu¹

¹Microsoft Research Asia

²University of Science and Technology of China

{xiaoyizhang, zongyuguo, jiahao li, binli, yanlu}@microsoft.com
 {jzy_ustc, lihq}@ustc.edu.cn

Abstract

Long-form video understanding presents significant challenges due to extensive temporal-spatial complexity and the difficulty of question answering under such extended contexts. While Large Language Models (LLMs) have demonstrated considerable advancements in video analysis capabilities and long context handling, they continue to exhibit limitations when processing information-dense hour-long videos. To overcome such limitations, we propose the **Deep Video Discovery (DVD)** agent to leverage an *agentic search* strategy over segmented video clips. Unlike previous video agents that rely on predefined workflows applied uniformly across different queries, our approach emphasizes the autonomous and adaptive nature of agents. By providing a set of search-centric tools on multi-granular video database, our DVD agent leverages the advanced reasoning capability of LLM to plan on its current observation state, strategically selects tools to orchestrate adaptive workflow for different queries in light of the gathered information. We perform comprehensive evaluation on multiple long video understanding benchmarks that demonstrates our advantage. Our DVD agent achieves state-of-the-art performance on the challenging LV-Bench dataset, reaching an accuracy of **74.2%**, which substantially surpasses all prior works, and further improves to **76.0%** with transcripts. The code has been released at <https://github.com/microsoft/DeepVideoDiscovery>.

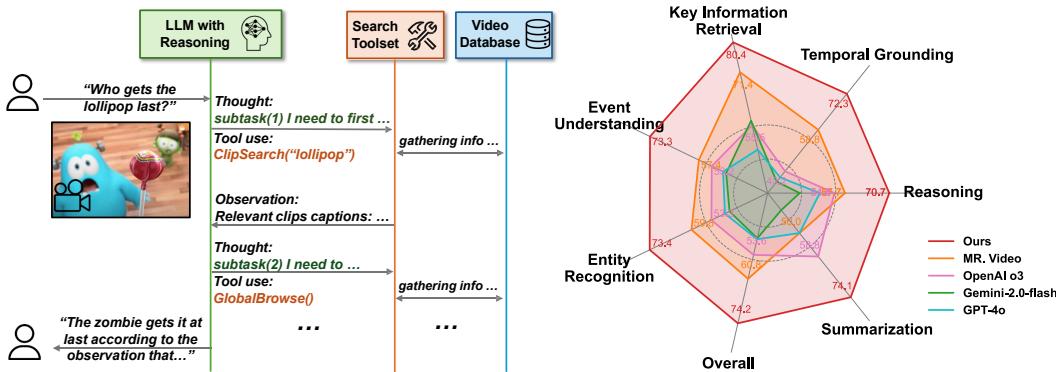


Figure 1: **Left:** Illustration of our Deep Video Discovery agent, which autonomously reasons on user query, iterative use tools to obtain the final answer. **Right:** Performance comparison on LV-Bench.

* Equal contribution.

† This work was done during the internship at Microsoft Research Asia as an open-source project.

1 Introduction

Long-form videos are ubiquitous in everyday life, spanning diverse domains such as movies, meeting recordings, sports games, and variety shows. Accurately comprehending and interpreting content within these extensive videos remains an intrinsically challenging task [8, 26, 31], demanding an ability to simultaneously integrate and reason about intricate spatiotemporal details across broad global contexts. Effective retrieval of relevant information from hour-long or even longer sequences not only necessitates attending to fine-grained local details but also simultaneously interpreting subtle semantic relations distributed throughout extended temporal intervals. Recent advancements in Large Language Models (LLMs) and Large Vision-language Models (VLMs) have notably improved capabilities in video understanding [17, 4, 28] and increased context length handling more than one million tokens [17, 25, 33]. However, even this extended context length remains insufficient for comprehending the information density typically found in long-form videos of hour-long duration. Empirical observations [17] also suggest a decline in the model’s effective instruction-following ability and reasoning clarity as the temporal dimension and information density increase.

Concurrently, recent breakthroughs [11, 18] on reasoning capability of LLMs have facilitated advances in agentic systems capable of complex information gathering tasks, such as Deep Research [16, 10, 20] or Deep Search [2, 3]. These agentic approaches demonstrate how decomposing difficult tasks into modular sub-tasks enables iterative reasoning, information searching, and content synthesis. Inspired by these successes, we conceptualize the challenge of understanding extremely long videos as a multi-step information search problem, where the video serves as the exploration environment and is segmented into multiple shorter video clips serving as manageable units of information. We name our approach as Deep Video Discovery (Fig. 1, left).

While existing video agent frameworks [34, 7, 19, 30] incorporate searching processes in their designs, they manually design the search process with their human prior. For instance, both VideoTree [30] and VCA [34] employ tree-based search strategies that navigate from root nodes to leaf nodes. This approach alleviates the context length limitations of VLMs but is inefficient for fine-grained queries which might benefit more from direct retrieving among leaf nodes. Additionally, semantically relevant entities may not exhibit temporal proximity, potentially diminishing the efficiency of backdate mechanism in tree-based search methods. Most critically, these existing frameworks prescribe fixed workflows that are uniformly applied across all query types, failing to adapt to the diverse information needs and optimal search strategies required by different questions.

In contrast to existing systems that typically rely on manually defined, rigid workflows, our approach is distinctly designed around an autonomous and adaptive agentic search paradigm for different queries. Instead of explicitly prescribing task workflows or search behaviors, we develop modular search tools that operate at multiple granularities, including (1) *Global Browse*, (2) *Clip Search*, and (3) *Frame Inspect*. Global Browse enables global summarization and indexing of subjects and global contexts across the entire video. Clip Search implements efficient semantic retrieval of relevant events within segmented clips. Specifically, Frame Inspection empowers the agent to extract fine-grained details directly from pixel-level information in a specified temporal range.

Equipped with this search-centric toolkit and multi-granular video database, our agent is inherently capable of autonomous reasoning, dynamic strategy orchestration, and iterative decision-making to proactively discover and extract crucial evidence. By leveraging the sophisticated reasoning capabilities intrinsic in the latest LLM, our agent does not merely use these tools independently, but adaptively combines their complementary strengths into a chain of thoughts and tool uses, effectively addressing diverse temporal-spatial and complex questions for long video. In the end, Deep Video Discovery can autonomously reason, plan, and retrieve pertinent information through adaptive search strategies tailored to specific video understanding queries.

We conduct comprehensive evaluations on long video benchmarks, demonstrating the efficiency and strong performance of our agent. In particular, on the challenging LVBench, we push forward the state-of-the-art performance by a large margin to 74.2% (as shown in Fig. 1, right), further achieving 76.0% with auxiliary transcripts. We also set a series of ablation studies that show the effectiveness of our tool design. In addition, we analyze the behavior patterns of different reasoning models in tool use sequences, providing future insight of developing agents for long video understanding tasks.

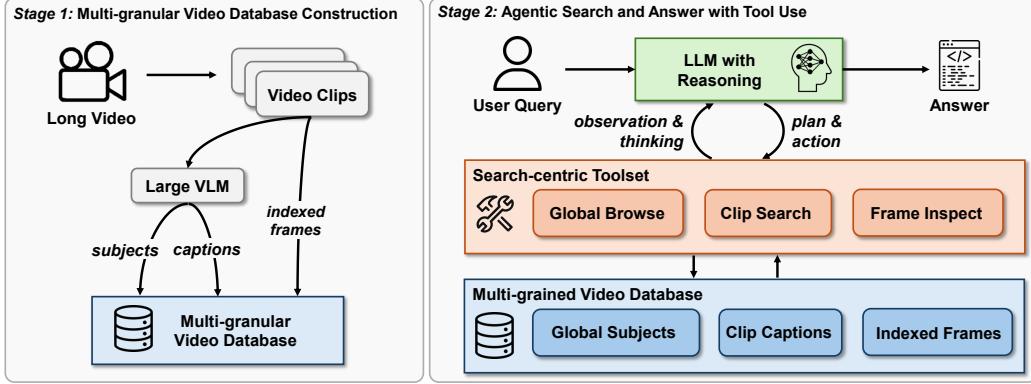


Figure 2: Deep Video Discovery consists of two stages: 1) Multi-granular Video Database Construction. We extract video information from different levels to enable comprehensive understanding, efficient retrieval, and preservation of original content. 2) Agentic Search and Answer. The agent iteratively reasons on user query and leverage the tailored toolset to gather information to answer.

2 Related Work

Long Video Understanding. Long video understanding remains a formidable challenge due to the intricate demands of temporal and spatial reasoning over extended durations and the complexity of information retrieval [31, 26]. Recent efforts in VLM for long video understanding primarily tackle challenges of limited input frame number by extending the context length of models [25, 6] or minimizing video redundancy to reduce visual token numbers [12, 14, 28]. Approaches such as AdaRETAKE [28] dynamically compress visual token by allocating adaptive compression ratios across time and model layers, thus significantly expanding the effective input frame number. However, token compression inherently introduces uncertainty regarding information loss, and models continue to face difficulties when answering complex queries under elongated context windows. In parallel, given the sparsity of key information about the given query, some works [27, 7, 34, 30, 19, 32] propose to explore the video content by agentic system. But they usually manually guide the agent about the search workflow by their priors [34, 30] or only allow the agent at simplex frame granularity to search [27], which cannot make full use of the reasoning capability of LLMs, resulting in suboptimal search efficiency and a lack of holistic, global comprehension of the long video content.

Agent and tool use. Recent advancements in large language models (LLMs), particularly their enhanced reasoning and planning capabilities, have significantly accelerated the development of autonomous agents [35, 39, 38, 40]. The ability to leverage external tools [23, 22, 21] further narrows the gap between general-purpose LLMs and real-world applications, enabling LLMs to acquire information, perform planning, and execute actions in complex environments. Our work extends this line of research to long video understanding, contributing to the broader investigation of solving complex video understanding tasks by integrating the advanced reasoning capabilities of LLMs with tool use. We introduce a suite of search-centric tools that allow LLMs to autonomously gather information at varying levels of granularity. By dynamically composing these tools, the agent can construct multi-step tool-use chains to improve the ability to answer complex queries effectively.

3 Deep Video Discovery

Overview. To solve the long-form video understanding problem in an agentic search way, we first build a multi-grained structured database from the long-form video. The database then serves for search-centric tools that work at different granularities. Specifically, our Deep Video Discovery agent consists of three main components: the multi-grained video database \mathcal{D} , search-centric toolset \mathcal{T} , and the LLM M as the agent’s orchestrator. Given the user query Q , the agent reasons iteratively to choose an action $A_i \in \mathcal{T} \cup \{\text{ANSWER}\}$ with parameters P to gather information for the video database \mathcal{D} or make decision to answer the query by referring to the accumulated information in this

process. In the following subsections, we sequentially introduce the multi-grained video database construction and Agentic Search and Answer with Tool Use.

3.1 Multi-granular Video Database Construction

Given an ultra long input video V , our goal is to transform it into a database that can provide efficient fast retrieval and also provide the original pixels of video for detailed information when necessary. Hence, we design it in a multiple granularity style which can provide different levels of video information for corresponding search tools. Specifically, we first segment the video into clips as the basic information unit then make the database include global summarized information to cover the whole video, a clip-based caption corpus and indexed frames from the clip. Fig. 2 (left) provides an overview. We introduce these components sequentially.

Temporal segmentation. We start by uniformly partitioning the input video V into a temporal sequence of non-overlapping short clips $\{v_i\}_{i=1}^N$, where the total segments $N = \lceil \frac{\text{len}(V)}{t} \rceil$. Empirically, we set $t = 5$ seconds to provide an adequate balance between computing cost and semantic and action completeness. Then all the video clips are decoded into frames $\{f_i\}_{i=1}^N$ under 2 frames per second for further process.

Multi-granular information extraction. Our multi-granular video information is designed as three levels: global video level, clip level and frame level. Specifically, at the global level we summarize the video content into a compact, subject-centric representation. At the clip level, we leverage textual captions to facilitate efficient information retrieval, while at the frame level we preserve original decoded frames indexed according to their corresponding clips, enabling precise reference and detailed analysis when required.

To derive the subject-centric global representation while minimizing redundancy in caption generation, we maintain an progressive structured subject registry S throughout the clip captioning process. Specifically, given a video clip v_i and decoded frame f_i , we prompt a large VLM to generate the captioning c_i and evolve registry whenever new subjects appear. The process is denoted as $S_i, c_i = VLM(f_i, S_{i-1})$ where S_0 is initialized as empty, and at the conclusion of the captioning process, the final subject registry is denoted by $S = S_N$. Each subject within the registry is represented by a comprehensive set of attributes, including name, physical appearance, identity descriptors, associated actions, and corresponding temporal spans in the video. The obtained caption c_i is subsequently embedded into a dense semantic vector $e_i \in \mathbb{R}^d$ using a language embedding model, facilitating fast retrieval in downstream applications. Despite careful design choices, perceptual compression inherent in caption generation inevitably entails some information loss. To mitigate this when necessary, we explicitly retain the decoded frames f_i alongside their corresponding textual captions and embeddings.

Outcome. The finalized database therefore encapsulates the decoded frames, captions and corresponding embedding triples, thus forming a structured database $\mathcal{D} = \{S, \{f_i, c_i, e_i\}_{i=1}^N\}$. This offline construction procedure transforms a lengthy raw video into a structured set of textually searchable embeddings with associated clips, while simultaneously preserving the complete visual content at pixel resolution. The resulting database becomes the basis for adaptive tool usage, enabling global information browsing, efficient semantic retrieval at the video-clip scale, and comprehensive grounding of generated outputs back to their source frames.

3.2 Agentic Search and Answer with Tool Use

With the built multi-granular video database, we design a set of search-centric tools that can enable global information understanding, efficient clip retrieval by semantic query, and details exploration on original video content. By equipping a reasoning large language model with this toolset, we build our DVD that can address complex user query on long video through autonomous planning and strategical search tool combination, as shown in Fig. 2 (right). We refer to this stage as **Agentic Search and Answer with Tool Use (ASA)**. We introduce this stage through two subsections: Search-centric Tool Preparation and Agent Design.

Table 1: Action space overview of our DVD. The first three actions are from our toolset and the final ANSWER action is designed as stop criterion.

Action	Parameter
GLOBAL BROWSE	video database \mathcal{D} user query Q
CLIP SEARCH	video database \mathcal{D} agent synthesized query \hat{Q} return top- k captions
FRAME INSPECT	video database \mathcal{D} agent synthesized query \hat{Q} temporal range $[t_s, t_e]$
ANSWER	the answer to user query

Algorithm 1: Agentic Search and Answer.

```

Input : Initial query  $Q$ , max step  $N$ , LLM  $M$ , tool set  $\mathcal{T}$ , action space  $\mathcal{A} = \mathcal{T} \cup \{\text{ANSWER}\}$ 
Output : Answer to  $Q$ 
Initialize history  $H_0 \leftarrow \{Q, \mathcal{A}\}$ 
for  $i \leftarrow 1$  to  $N$  do
     $R_i \leftarrow M.\text{reason}(H_{i-1})$ 
     $A_i, P_i \leftarrow M.\text{call}(R_i, H_{i-1})$  where  $A_i \in \mathcal{A}$ 
    if  $A_i = \text{ANSWER}$  then
        break
    end
     $O_i \leftarrow A_i(P_i)$ 
     $H_i \leftarrow H_{i-1} \cup \{(R_i, A_i, O_i)\}$ 
    if  $t = N$  then
         $P_i \leftarrow M.\text{answer}(H_i)$ 
    end
end
return ANSWER( $P_i$ )

```

3.2.1 Search-centric Tool Preparation

Leveraging the established video database, we have developed a suite of tools designed to efficiently gather information from video data at varying levels of granularity. Specifically, we divide long videos into three distinct hierarchical levels and introduce corresponding specialized tools: (1) Global Browse, (2) Clip Search, and (3) Frame Inspect. Given the significant computational cost associated with processing lengthy videos using VLMs, our tool design carefully balances efficiency and performance. Central to our approach is an agentic search paradigm, wherein the agent decomposes the user query and strategically chains up tools with synthesized parameters, enabling iterative reasoning and information collection to resolve the task. Through the effective integration and coordinated use of these tools, the agent progressively enhances its understanding of user intent and precisely locates relevant information within extensive video content. We introduce the three tools sequentially in the following paragraphs.

Tool: Global Browse. The Global Browse tool takes the video database and the original user query as input, and returns global summaries capturing high-level contextual information. We construct two distinct types of global information: subject-centric and event-centric summaries. For subject-centric summarization, we pre-construct it when building the multi-granular video dataset as mentioned in Section 3.1 since it is query-irrelevant. For event-centric summarization, we uniformly sample frames across the entire video and feed these sampled frames into the VLM. We instruct the VLM to describe noteworthy events explicitly related to the original user query. Upon invocation by the agent, the Global Browse tool efficiently retrieves and returns these global representations, providing the agent immediate access to high-level global context information.

Tool: Clip Search. Clip Search provides a mid-level granularity retrieval capability, enabling fast and efficient exploration of video content via caption embedding. Given a query \hat{Q} synthesized based on the agent’s current internal reasoning context, this module retrieves a ranked list of top- k relevant video clips along with their captions. Specifically, the tool computes the cosine similarity between the embedding of the provided query and the pre-computed embeddings of all video clip captions, returning the clips corresponding to the highest-ranked caption matches. Each retrieved observation contains both the corresponding caption and the time ranges of the associated video clip. To achieve an accurate and detailed understanding, the agent can iteratively invoke this tool, progressively refining temporal constraints or reformulating its queries based on newly acquired contextual knowledge. This iterative chain-of-query approach effectively guides the agent toward precise temporal segments relevant to the original high-level query.

Tool: Frame Inspect. Frame Inspect receives a temporal range $[t_s, t_e]$ within the video and an sub-query freely defined by the agent as input, returning an open format visual-question-answering (VQA) response. The agent can invoke this tool whenever explicit frame-level details such as subtle attributes, object counting, or fine-grained spatial relationships, are required but not clearly depicted

in captions or global summaries. The open-ended query format allows significant freedom for the agent to leverage its reasoning capability, enabling highly adaptable visual inspection. Specifically, the Frame Inspect tool loads raw frames from the requested interval and prompts a VLM with these frames and agent-synthesized query. To ensure computational efficiency, we limit processing to a maximum of 50 frames, uniformly sampling from frames exceeding this limit. The resulting response thus equips the agent with accurate, visually-grounded evidence essential for detailed reasoning tasks.

3.2.2 Agentic Design

To maximally leverage the reasoning and planning capacity intrinsic to modern LLMs, we intentionally abstain from manually instructing explicit searching workflow or tool utilization patterns. Instead, we enable the agent to reason, plan, and take actions through a streamlined iterative *observe-reason-act* loop, similar to ReAct[35]. For a given query, the agent reasons about its current observation state, strategically selects search tools, formulates appropriate parameters for actions, and dynamically refines its internal reasoning in light of the gathered evidence. Within ASA, the LLM acts as a sophisticated cognitive driver, taking actions at each iteration based on cumulative knowledge and reasoned evidence, thereby reinforcing its pivotal role in adaptively navigating the discovery process.

Specifically, as illustrated in Algorithm 1, given an initial user query Q , a predefined action space $\mathcal{A} = \mathcal{T} \cup \{\text{ANSWER}\}$, and a maximum allowable step count N , our agent performs iterative reasoning to strategically navigate the available actions. The agent leverages an LLM M to reason upon the current dialogue history, plan its immediate action, interact with the toolset $\mathcal{T} = \{\text{GLOBALBROWSE}, \text{CLIPSEARCH}, \text{FRAMEINSPECT}\}$, and collect observations O_i . More concretely, at each step t , the agent maintains a historical context H_i , reflects to generate a reasoning step R_i , selects an action $A_i \in \mathcal{T} \cup \{\text{ANSWER}\}$ accompanied by relevant parameters P_i , and receives subsequent observation outcomes O_i from the environment. These components, reasoning, action, and obtained outcomes, are successively appended to the interaction history H_i , enriching the context for subsequent iterations of inference. The iterative process terminates either when the agent explicitly selects the `ANSWER` action, or upon reaching the step limit N , at which prompts the agent directly generates a final answer prediction. The agent then outputs the final answer to the original user query.

By positioning the LLM’s sophisticated reasoning at the core of this iterative loop, this approach endows the agent with an inherently autonomous, evidence-guided, and flexible action-taking mechanism. This autonomous and iterative paradigm fosters a strategic and context-sensitive inquiry cycle, thereby enabling the agent to effectively leverage the available tools to iteratively decompose the original query into progressively refined sub-queries, updating and improving the query representation as it receives new observations. Through iterative reasoning and interaction cycles, guided by deeper and increasingly comprehensive observations collected from prior tool usage, the agent systematically enhances its understanding and interpretation of the task context, ultimately leading to more accurate and informed answers to the given question.

4 Experiment

4.1 Benchmarks

We assess the long-form video understanding capabilities of Deep Video Discovery using several established long video benchmarks. Our primary evaluation benchmark, **LVbench** [26], includes 1,549 multiple-choice questions across 103 hour-long videos. It stands as one of the most comprehensive and challenging benchmarks for extreme long-form video understanding. **LongVideoBench** [31] features 6,678 questions from 3,763 videos, ranging in duration from a few seconds to an hour. We emphasize the longest subset with durations in $(900\text{s}, 3600\text{s}]$ (denoted as the *Long* subset), comprising 564 questions from 188 videos. **Video MME** [8] is segmented by video duration; we concentrate on the *Long* subset without subtitles to isolate long-video comprehension, covering 300 videos of 30 to 60 minutes with 900 questions. Finally, **EgoSchema** [15] serves as a diagnostic benchmark for long-video understanding, where we evaluate on its validation split of 500 videos with 500 questions.

4.2 Implementation Details

Baselines. We compare Deep Video Discovery with a range of long-video understanding systems, including both VLM-based [24, 1, 18, 9, 36, 29, 37, 4, 13, 28] and agent-based approaches [30, 7,

Table 2: Comparison on LVBench under different question categories. The metric shown is accuracy (%).

Methods	ER	EU	KIR	TG	Rea	Sum	Overall
<i>Commercial VLMs</i>							
Gemini-1.5-Pro [24]	32.1	30.9	39.3	31.8	27.0	32.8	33.1
Gemini-2.0-Flash [24]	47.4	48.5	56.8	39.3	44.4	41.4	48.6
GLM-4V-Plus [9]	46.2	47.8	54.1	42.7	46.5	37.9	48.7
GPT-4o [1]	48.9	49.5	48.1	40.9	50.3	50.0	48.9
OpenAI o3 [18]	57.6	56.4	62.9	46.8	50.8	67.2	57.1
<i>Open-Source VLMs</i>							
InternVL2.5-78B [29]	43.8	42.0	42.1	36.8	51.0	37.9	43.6
VideoLLaMA3-7B [37]	45.8	42.4	47.8	35.9	45.8	36.2	45.3
Qwen2.5-VL-72B [4]	-	-	-	-	-	-	47.7
VideoChat-Flash [13]	51.1	46.0	49.0	38.9	48.5	34.5	48.2
AdaRETAKE [28]	53.0	50.7	62.2	45.5	54.7	37.9	53.3
<i>Video Agents and Others</i>							
VideoTree [30]	30.3	25.1	26.5	27.7	31.9	25.5	28.8
VideoAgent [27]	28.0	30.3	28.0	29.3	28.0	36.4	29.3
VCA [34]	43.7	40.7	37.8	38.0	46.2	27.3	41.3
MR. Video [19]	59.8	57.4	71.4	58.8	57.7	50.0	60.8
Deep Video Discovery (Ours)	73.4	73.3	80.4	72.3	70.7	74.1	74.2
+ Auxiliary transcripts	75.5	77.1	79.0	72.7	68.7	84.5	76.0

34, 19]. Most baseline results are taken from official leaderboards or published reports, except for the recently released OpenAI o3 [18], which has not yet been evaluated on these benchmarks. Following [19], we uniformly sample 256 frames per video to evaluate OpenAI o3.

Deep Video Discovery flexibly integrates different models depending on the needs of each component. For the VLM in video database construction, we use GPT-4.1 [17] to produce high-quality captions on LVBench, and GPT-4.1-mini for other benchmarks to reduce cost. During agentic search and answering, we employ OpenAI o3 as LLM M for its strong reasoning ability, including in the Frame Inspect module for fine-grained VQA. All frames are resized to 720p to maintain visual details. In Clip Search, we set 16 as the default value of top- k while leaving the flexibility for LLM to change it. Maximum reasoning step is set to $N = 15$. To explore the upper bound of understanding ability, we additionally evaluate LVBench using *auxiliary transcripts*. Audio is transcribed with WhisperX[5], and transcripts are used to guide video segmentation and enrich captions. This audio-visual fusion enhances understanding of long, complex content, leading to stronger results.

API Content filtering. We use LLM API via Azure OpenAI Service. We observe that the safety content filtering mechanism of the service misjudges a small part of data from the benchmark as offensive and block the request, which leads to the reduced performance of both OpenAI o3 baseline and our DVD agent. We provide more details and mitigation strategies in Appendix A.3.2.

4.3 Main Results

Table 2 presents the comparison results on LVBench. DVD significantly outperforms all baselines, surpassing the previous SOTA MR. Video by 13.4%. Compared to the prior leading video agent VCA, our method achieves a remarkable 32.9% gain. Against our base VLM, OpenAI o3, our full system delivers a substantial 17.1% gain, highlighting the importance of agentic reasoning. Incorporating transcript information provides an additional 1.8% boost. These results highlight the effectiveness of our search-centric agentic reasoning framework in handling ultra-long video understanding tasks.

Table 3 provides a comprehensive evaluation across several long-video benchmarks. On LongVideoBench, DVD outperforms the previous SOTA by 4.1% overall and 7.0% on the longest-

Table 3: Comparison on long video benchmarks.

Methods	LVBench	LongVideoBench (Val)	Video MME	EgoSchema	
	Overall	Overall	Long	Long (w/o sub)	Val
<i>Commercial VLMs</i>					
Gemini-1.5-Pro [24]	33.1	64.0	58.6	67.4	-
Gemini-2.0-Flash [24]	48.3	-	45.7	63.0	71.2
GPT-4o [1]	48.9	66.7	60.9	65.3	70.4
OpenAI o3 [18]	57.1	67.5	60.6	64.7	63.2
<i>Open-Source VLMs</i>					
mPLUG-Owl3 [36]	43.5	59.8	-	50.1	-
InternVL2.5-78B [29]	43.6	63.6	-	62.6	-
Qwen2.5-VL-72B [4]	47.7	60.7	-	63.9	-
AdaRETAKE [28]	53.3	67.0	-	65.0	-
<i>Video Agents and Others</i>					
VideoTree [30]	28.8	-	-	-	67.0
VideoAgent [27]	29.3	-	-	-	63.2
VCA [34]	41.3	-	-	-	73.6
MR. Video [19]	60.8	-	61.6	61.8	73.0
Deep Video Discovery (Ours)	74.2	71.6	68.6	67.3	76.6

Table 4: Ablation on used models. $M_{database}$ for captioning in database construction, $M_{reasoning}$ for reasoning in ASA, M_{tool} for Frame Inspect.

Adopted models			LVBench
$M_{database}$	$M_{reasoning}$	M_{tool}	w/ transcripts
4.1	o3	4.1-mini	72.3
4.1	o4-mini	o3	70.2
4.1	4o	o3	62.3
4.1-mini	o3	o3	71.9
4.1	o3	o3	76.0

Table 5: Ablation on the search-centric tools \mathcal{T} . Note that the anchor uses 4.1-mini for $M_{database}$, and o3 for both $M_{reasoning}$ and M_{tool} .

Search-centric Tools			LVBench
Global Browse	Clip Search	Frame Inspect	w/ transcripts
	✓	✓	69.0
	✓	✓	59.6
	✓	✓	63.5
	✓	✓	71.9

duration subset. On the Video MME Long subset, it beats the best open-source VLM, AdaRETAKE, by 2.3%, and MR. Video by 5.5%, approaching the performance of Gemini-1.5-Pro. On EgoSchema, our method exceeds the previous best by 3.0%. Notably, it exceeds reported human-level accuracy of $\sim 76\%$ on this benchmark. Across all datasets, our system consistently outperforms the base VLM OpenAI o3, confirming the effectiveness and generalizability of our agentic reasoning framework.

4.4 Ablation Study

We evaluate the impact of different model choices across system components. By default, GPT-4.1 is used for captioning and subject extraction during Multi-granular Video Database Construction and OpenAI o3 serves as the reasoning model in the Agentic search and Answer with tool use process while Frame Inspect tool also leverages OpenAI o3 to query the fine-grained details on the frame pixels. We denote the three models as $M_{database}$, $M_{reasoning}$ and M_{tool} in Table 4. Replacing GPT-4.1 with GPT-4.1-mini for database construction or Frame Inspect tool results in moderate drops of 4.1% and 3.7%, respectively, indicating relatively minor impact. For reasoning model in agentic search, switching to OpenAI o4-mini [18] leads to a 5.8% drop, while GPT-4o causes a substantial 13.7% decline. It highlights the reasoning model as the most critical component in our agentic system because our system is designed surrounding and to make full use of the reasoning capability of LLM. The lack of reasoning ability leads to the collapse of agent behavior, as analyzed further in the subsequent subsection.

Table 6: Comparison of DVD agent with different LLMs as reasoning models on **LVBench**. * denotes Qwen3-32B-Thinking model.

Category	Model	Accuracy (%)
DVD w/ closed-sourced	OpenAI o3	76.0
	GPT-4o	62.3
DVD w/ open-sourced	DeepSeek-R1	68.5
	DeepSeek-V3	57.5
	Qwen3-32B*	57.3

Table 7: Comparison DVD and its variation on different max step number.

Max Step Limit	8	12	15
Ours			
Avg. Actual Steps	6.7	7.2	7.3
Accuracy (%)	72.3	73.8	74.2
Ours w/ VideoAgent [27] workflow			
Avg. Actual Steps	5.0	8.3	11.1
Accuracy (%)	48.4	66.3	70.2

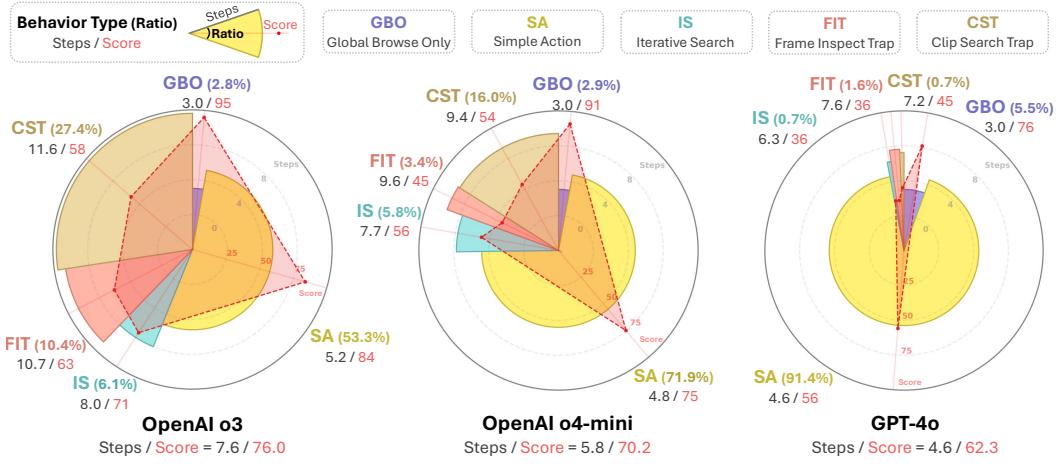


Figure 3: Analysis of the behavior of Deep Video Discovery using different *reasoning* models. We categorize tool-calling behavior into five types. For each type, we report its proportion (*Ratio*, sector angels), average reasoning steps (*Steps*, sector radius) and score (*Score*, dashed lines). A clear correlation emerges among behavior patterns, reasoning depth, and score (see Section 4.6 for details).

We next examine how the choice of reasoning model affects performance when pairing DVD with various open-source LLMs on LVBench. As reported in Table 6, combining DVD with the open-source DeepSeek-R1 reasoning model yields 68.5% accuracy, surpassing all prior methods. Notably, even when restricted to a 32B model Qwen3-32B, DVD attains 57.3% accuracy, outperforming existing VLM approaches, including GPT-4o and OpenAI o3.

Finally we assess the contribution of each tool in the agentic search and answer phase (Table 5). Removing Global Browse which is responsible for global summarization and long-range event linking leads to a 2.9% drop. Disabling Frame Inspect with the fine-grained VQA results in a 8.4% decline, highlighting its role in fine-grained understanding. Removing Clip Search causes the largest drop of 12.3%, as it breaks the searching ability for iteratively refine reasoning. These results underscore the importance of tool integration in our search-centric framework.

4.5 Analysis on Efficiency

To understand the efficiency of our designed agent, we compare DVD’s performance with different maximum step limits on LVBench without transcripts. Since the core strength of DVD lies in its adaptive workflow orchestrated by the reasoning model, we also evaluate a variant that replaces this adaptive workflow with the hand-crafted workflow from VideoAgent [27] while keeping all other settings identical. Specifically, the VideoAgent workflow follows a fixed sequence: first predicting answers and confidence scores, then identifying missing information through self-reflection, searching video clips for the missing information, merging found information into current observations, and finally re-predicting answers and confidence scores. As shown in Table 7, the hand-crafted approach

results in a significant 7.5% absolute accuracy drop while requiring more steps. Even when scaled to 11.1 average steps (52% more than our DVD), the hand-crafted workflow achieves only 70.2% accuracy, still 4.0% below our adaptive approach. This comparison highlights the superior search efficiency of our adaptive workflow design.

4.6 Analysis on Agentic Reasoning Behavior

The reasoning model is the most critical component in DVD. During the *observe-reason-act* loop, the agent autonomously integrates current context and flexibly decides the next tool to invoke. To better understand this, we analyze the tool-calling behavior during the agentic search and answer phase and category it into five types for analysis (see Fig. 3). At the same time, we provide more detailed qualitative case study in Appendix B.2.

Global Browse Only. The agent confidently answers after a single Global Browse call. This pattern, though rare, correlates with very high accuracy, suggesting the initial global context was sufficient.

Simple Action. The agent follows a direct *search-inspect-answer* path, typically involving one or two Clip Search and Frame Inspect calls. This is the most common and effective strategy, employed in over half of the queries and achieving high accuracy.

Iterative Search. The agent alternates between Clip Search and Frame Inspect multiple times. This indicates an initial difficulty in locating sufficient information, requiring a more exploratory approach. These trajectories are longer (*e.g.*, 8.0 vs. 5.2 steps for OpenAI o3) and yield slightly lower accuracy than a Simple Action.

Frame Inspect Trap. The agent becomes stuck in a loop of fine-grained analysis, invoking Frame Inspect more than three consecutive times without reaching a conclusion. This behavior leads to prolonged reasoning chains and significantly lower accuracy.

Clip Search Trap. The agent repeatedly calls Clip Search more than three times consecutively, often when key information is absent from the video database. The agent fails to adapt, resulting in unproductive loops. This pattern is a primary failure mode for the OpenAI o3 model.

This analysis yields two key insights into designing effective autonomous video agents:

Insight 1: The dual nature of reasoning length. For a single model, longer reasoning trajectories often signal uncertainty and correlate with lower accuracy (*e.g.*, Simple Action vs. Iterative Search). However, when comparing across models, superior performance is often achieved by the model capable of conducting more thorough, and thus longer, reasoning processes.

Insight 2: Overconfidence leads to behavioral collapse. The GPT-4o model significantly underperforms, collapsing into the Simple Action pattern for 91.4% of queries. With an average of only 4.6 reasoning steps, it frequently concludes prematurely, rarely exploring alternative strategies. This “overconfidence” and lack of flexibility appear to be the primary cause of its poor performance, highlighting the need for agents that can balance exploitation with exploration.

5 Conclusion

We introduce the proposed Deep Video Discovery agent for long-form video understanding, utilizing multi-granular search tools on constructed database for iterative search and reasoning over extensive video content. Our approach outperforms prior methods by adaptively integrating global browsing, clip search, and frame inspection, as demonstrated by state-of-the-art results on multiple benchmarks. Ablation studies confirm the effectiveness of our tool design, while analyses of reasoning model behavior provide insight into model reasoning patterns. Overall, our framework offers a scalable and flexible solution for comprehensive analysis of long videos.

Limitations. While our agent significantly improves long video understanding, the iterative reasoning introduces higher computational overhead. In future work, we will explore more effective database construction and searching to reduce reasoning difficulty and thereby lower computational costs.

Change Log

- **v1** (2025-05-23): Initial submission.
- **v2** (2025-05-28): Fixed the evaluation code to correctly account for answers enclosed in parentheses, resulting in consistently improved reported accuracy.
- **v3** (2025-07-24): GitHub repository link update.
- **v4** (2025-11-03): Camera-ready version. Update Tab. 6 and 7 to discuss efficiency of our DVD.

References

- [1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] J. AI. DeepSearch - Jina AI. <https://jina.ai/deepsearch/>, 2025.
- [3] X. AI. Grok 3 Beta — The Age of Reasoning Agents. <https://x.ai/news/grok-3>, 2025.
- [4] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- [5] M. Bain, J. Huh, T. Han, and A. Zisserman. Whisperx: Time-accurate speech transcription of long-form audio. *INTERSPEECH 2023*, 2023.
- [6] Y. Chen, F. Xue, D. Li, Q. Hu, L. Zhu, X. Li, Y. Fang, H. Tang, S. Yang, Z. Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024.
- [7] Y. Fan, X. Ma, R. Wu, Y. Du, J. Li, Z. Gao, and Q. Li. VideoAgent: A memory-augmented multimodal agent for video understanding. In *ECCV*, 2024.
- [8] C. Fu, Y. Dai, Y. Luo, L. Li, S. Ren, R. Zhang, Z. Wang, C. Zhou, Y. Shen, M. Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. *arXiv preprint arXiv:2405.21075*, 2024.
- [9] T. GLM, A. Zeng, B. Xu, B. Wang, C. Zhang, D. Yin, D. Zhang, D. Rojas, G. Feng, H. Zhao, et al. ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.
- [10] Google. Gemini Deep Research - your personal research assistant. <https://gemini.google/overview/deep-research>, 2025.
- [11] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [12] Y. Han, Q. Guo, L. Pan, L. Liu, Y. Guan, and M. Yang. Dynfocus: Dynamic cooperative network empowers llms with video understanding. *arXiv preprint arXiv:2411.12355*, 2024.
- [13] X. Li, Y. Wang, J. Yu, X. Zeng, Y. Zhu, H. Huang, J. Gao, K. Li, Y. He, C. Wang, Y. Qiao, Y. Wang, and L. Wang. VideoChat-Flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*, 2024.
- [14] X. Liu, Y. Shu, Z. Liu, A. Li, Y. Tian, and B. Zhao. Video-xl-pro: Reconstructive token compression for extremely long video understanding. *arXiv preprint arXiv:2503.18478*, 2025.
- [15] K. Mangalam, R. Akshulakov, and J. Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2023.
- [16] OpenAI. Introducing deep research. <https://openai.com/index/introducing-deep-research/>, 2025.
- [17] OpenAI. Introducing GPT-4.1 in the API. <https://openai.com/index/gpt-4-1/>, 2025. Accessed: 2025-04-14.

- [18] OpenAI. Introducing OpenAI o3 and o4-mini. <https://openai.com/index/introducing-o3-and-o4-mini/>, 2025. Accessed: 2025-05-15.
- [19] Z. Pang and Y.-X. Wang. Mr. video:" mapreduce" is the principle for long video understanding. *arXiv preprint arXiv:2504.16082*, 2025.
- [20] Perplexity. Introducing Perplexity Deep Research. <https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research>, 2025.
- [21] Y. Qin, S. Hu, Y. Lin, W. Chen, N. Ding, G. Cui, Z. Zeng, X. Zhou, Y. Huang, C. Xiao, et al. Tool learning with foundation models. *ACM Computing Surveys*, 57(4):1–40, 2024.
- [22] C. Qu, S. Dai, X. Wei, H. Cai, S. Wang, D. Yin, J. Xu, and J.-R. Wen. Tool learning with large language models: A survey. *Frontiers of Computer Science*, 19(8):198343, 2025.
- [23] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.
- [24] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- [25] G. Team, P. Georgiev, V. I. Lei, R. Burnell, L. Bai, A. Gulati, G. Tanzer, D. Vincent, Z. Pan, S. Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.
- [26] W. Wang, Z. He, W. Hong, Y. Cheng, X. Zhang, J. Qi, X. Gu, S. Huang, B. Xu, Y. Dong, et al. LVBench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*, 2024.
- [27] X. Wang, Y. Zhang, O. Zohar, and S. Yeung-Levy. VideoAgent: Long-form video understanding with large language model as agent. In *ECCV*, 2024.
- [28] X. Wang, Q. Si, J. Wu, S. Zhu, L. Cao, and L. Nie. Adaretake: Adaptive redundancy reduction to perceive longer for video-language understanding. *arXiv preprint arXiv:2503.12559*, 2025.
- [29] Y. Wang, X. Li, Z. Yan, Y. He, J. Yu, X. Zeng, C. Wang, C. Ma, H. Huang, J. Gao, et al. InternVideo2.5: Empowering video MLLMs with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025.
- [30] Z. Wang, S. Yu, E. Stengel-Eskin, J. Yoon, F. Cheng, G. Bertasius, and M. Bansal. VideoTree: Adaptive tree-based video representation for LLM reasoning on long videos. *arXiv preprint arXiv:2405.19209*, 2024.
- [31] H. Wu, D. Li, B. Chen, and J. Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. In *NeurIPS*, 2024.
- [32] Y. Yan, S. Jiang, T. Cao, Y. Yang, Q. Yang, Y. Shu, Y. Yang, and L. Qiu. Empowering agentic video analytics systems with video language models. *arXiv preprint arXiv:2505.00254*, 2025.
- [33] A. Yang, B. Yu, C. Li, D. Liu, F. Huang, H. Huang, J. Jiang, J. Tu, J. Zhang, J. Zhou, et al. Qwen2. 5-1m technical report. *arXiv preprint arXiv:2501.15383*, 2025.
- [34] Z. Yang, D. Chen, X. Yu, M. Shen, and C. Gan. VCA: Video curious agent for long video understanding. *arXiv preprint arXiv:2412.10471*, 2024.
- [35] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. ReAct: Synergizing reasoning and acting in language models. In *ICLR*, 2023.
- [36] J. Ye, H. Xu, H. Liu, A. Hu, M. Yan, Q. Qian, J. Zhang, F. Huang, and J. Zhou. mPLUG-OWL3: Towards long image-sequence understanding in multi-modal large language models. In *ICLR*, 2024.

- [37] B. Zhang, K. Li, Z. Cheng, Z. Hu, Y. Yuan, G. Chen, S. Leng, Y. Jiang, H. Zhang, X. Li, P. Jin, W. Zhang, F. Wang, L. Bing, and D. Zhao. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.
- [38] K. Zhang, J. Li, G. Li, X. Shi, and Z. Jin. Codeagent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13643–13658, 2024.
- [39] Z. Zhang, X. Zhang, W. Xie, and Y. Lu. Responsible task automation: Empowering large language models as responsible task automators. *arXiv preprint arXiv:2306.01242*, 2023.
- [40] Z. Zhang, X. Liu, X. Zhang, J. Wang, G. Chen, and Y. Lu. Ui-evol: Automatic knowledge evolving for computer use agents. *arXiv preprint arXiv:2505.21964*, 2025.

A Evaluation Details

A.1 Prompts

We describe the prompts used by different models within the proposed DVD, including: (1) Captioning and subject extraction model M_{database} used in Multi-granular Video Database Construction (Tab. 9); (2) Reasoning model $M_{\text{reasoning}}$ used in Agentic Search and Answer (Tab. 10); (3) Fine-grained detail querying model M_{tool} used in the Frame Inspect tool (Tab. 11); (4) event-centric global information in the Global Browse tool (Tab. 12). The JSON schema for tool use of OpenAI model are also provided in Tab. 13 and Tab. 14.

A.2 Cost

We measure the average API cost of our agent when calling OpenAI o3 on LVbench. On average, DVD incurs a cost of \$0.213 with 0.15 million tokens per question with OpenAI API price at 2025-11-03.

A.3 Azure OpenAI Service

We utilize the LLM API through Azure OpenAI Service to evaluate both our proposed DVD agent and the OpenAI o3 model [18]. However, we note two key limitations in the service that may affect performance: (1) a restriction on the maximum number of input images, and (2) a safety content filtering mechanism. This section discusses their impact and our mitigation strategies.

A.3.1 Maximum Image Count Limitation

Azure OpenAI Service imposes a limit of 50 images per request, which is significantly lower than the token limit allowed by Large VLMs we used. This restriction poses notable challenges in tasks involving long-form video understanding, as the model’s comprehension capability strongly depends on the number of input frames provided. To overcome this limitation and deliver richer visual context within the imposed constraints, we adopt a strategy of combining multiple frames into a single composite image:

For DVD, we horizontally splice up to 5 frames into a single image. This allows us to input up to 250 frames (50 images \times 5 frames). Note that we only use the 250 frames setting in event-centric global information gathering in the Global Browse tool while maximum 50 frames are used in the Frame Inspect tool.

For OpenAI o3, we use a 2 \times 4 format—splicing up to 4 frames horizontally and stacking 2 such groups vertically per image. This enables our test condition of 256 frames by using 32 spliced images. We find that this configuration yields better performance than the 1 \times 5 format used in DVD.

A.3.2 API Content Filtering

Azure OpenAI’s safety filtering mechanism occasionally misclassifies benign inputs as offensive, blocking some requests. This affects the performance of both OpenAI o3 and our DVD agent. For such filtered questions, we adopt different mitigation strategies to reduce its influence.

For OpenAI o3, when a query is blocked, we randomly select an answer from the remaining candidates. Across benchmarks, approximately 10% to 25% of queries are filtered, as shown in Tab. 8. To better assess true performance, we also report scores on the subset of unfiltered responses, where performance improves significantly.

For DVD, when VLM returns none due to the content filtering, we do not implement additional error handling for the agent. Instead, we let the agent make its own decisions to handle error. It may invoke alternative tools or attempt to answer based on available information. If a captioning query is blocked, we leave the corresponding entry blank in the database. While this affects database completeness, the agent remains operational due to its adaptive behavior. Tab. 8.

Table 8: Statistics for content filtering in Azure OpenAI API.

Methods	LVBench	LongVideoBench (Val)	Video MME	EgoSchema	
	Overall	Overall	Long	Long (w/o sub)	Val
OpenAI o3	57.1	66.7	59.6	64.7	63.2
Unfiltered Ratio	83%	90%	85%	83%	75%
Unfiltered Subset Score	63.3	71.5	72.6	73.2	77.5
Deep Video Discovery (Ours)	71.9	70.5	68.4	66.8	76.6
Unfiltered Ratio	24%	61%	40%	24%	54%
Unfiltered Subset Score	70.5	71.6	70.5	69.7	77.0

B More Results

B.1 Statistical Significance

To assess the statistical significance of DVD, we evaluate it on LVBench (using auxiliary transcriptions) three times. It achieves an average score of 74.0 with a variance of 0.125, demonstrating the consistency and robustness of the proposed method.

B.2 Case Study

We provide illustrative examples of the tool-calling process for different agent behaviors described in Section 4.5 of the paper. All examples are from LVBench [26]. Each example highlights how the agent selects actions and invokes tools to gather information for answering the question, and how each tool call retrieves relevant data from the database. In the tool responses, we highlight **useful and correct information** in orange and **useless or incorrect information** in purple.

Global Browse Only is shown in Fig. 4. When asked about the type of video, the agent directly invokes Global Browse to retrieve global context. The subject-level browsing reveals many technical subjects, while the event-level browsing summarizes the key themes and categorizes the video as a technology documentary. Based on this information, the agent correctly classifies the video into a technology documentary.

Simple Action is shown in Fig. 5. In this case, Global Browse incorrectly indicates that the mirror fragments are thrown away. To verify, the agent uses Clip Search to locate the time range of the mirror breaking event and then calls Frame Inspect for finer-grained analysis. It discovers that the fragments actually fly into the sky and thus selects the correct answer.

Iterative Search is shown in Fig. 6. This question involves multiple subjects (boy, dog, cat, bird, and man), making it more complex. Global Browse initially suggests that the man is chasing the dog. However, the first round of Clip Search and Frame Inspect indicates that he is chasing a bird. To resolve the ambiguity, the agent issues two more Clip Search queries for events involving "man chasing dog" and "man chasing bird". The search yields relevant and useful results only for the later, suggesting that the man is actually chasing the bird. To confirm the subject's identity, the agent checks for the presence of a mustache. Convincing that the mustached man is chasing the bird, the agent selects the correct option. Notably, when a Frame Inspect call fails to find relevant results in step 6, the agent can automatically ask another question to obtain an useful response.

Frame Inspect Trap is shown in Fig. 7. This question targets an event at a specific timestamp. Around this time, the agent asks three consecutive questions. While Frame Inspect sometimes captures accurate information, it occasionally fails to interpret frames correctly, providing misleading responses. As a result, the agent cannot discern which information is reliable and ultimately answers incorrectly.

Clip Search Trap is shown in Fig. 8. This example relies on identifying the correct time range using key terms like "protagonist" and "utility room." However, the database contains few directly relevant captions or subjects. Despite the agent's attempts to rephrase and refine its Clip Search queries, the tool fails to return relevant results, leading to an incorrect answer.

```

// system prompt
You are a helpful assistant.

// user prompt
There are consecutive frames from a video. BENCHMARK_CUSTOM_PLACEHOLDER Please understand the video clip
with given corresponding transcript and historical subject information then output following JSON format
with filled content:

Transcript of the current video clip:

[TRANSCRIPTS]

Historical subject information:

[SUBJECTS]

Output template:
{
  "clip_start_time": CLIP_START_TIME,
  "clip_end_time": CLIP_END_TIME,
  "new_subject_registry": { # add new subjects if any not in the historical subject information else leave
    it empty
    <new_subject_id>: {
      "name": <new_subject_name if the name is unknown please use 'unknown'>,
      "appearance": <a list of subject appearance description phase, including hair, face, clothes>,
      "identity": <a list of subject identity description shown in current video clip>,
      "first_seen": <timestampl>
    },
    ...
  },
  "clip_description": {
    "subjects_present": [<subject_id_list>],
    "captions": <a smooth and very detailed narration about the whole video clip>
  }
}

```

Table 9: Prompts for captioning and subject extraction model M_{database} in video database construction.

C Broader Impacts

C.1 Positive Impacts

DVD facilitates natural language interaction with long-form video content, improving accessibility for users with visual or cognitive impairments. It can also enhance efficiency in media summarization and analysis, significantly reducing the need for manual review.

C.2 Negative Impacts

As DVD leverages large models (e.g., LLMs and VLMs) for captioning and reasoning, it inherits common concerns such as potential bias from training data, which may result in inaccurate or unfair interpretations of video content. Additionally, the agentic search process demands substantial computational resources, raising sustainability and accessibility concerns.

```

// system prompt
You are a helpful assistant who can answer multistep questions by sequentially calling functions. Follow a pattern of THOUGHT (reason step-by-step about which function to call next), ACTION (call a function to take a next step towards the final answer), OBSERVATION (output of the function). Reason step by step which actions to take to get to the answer.

Only call functions with arguments coming verbatim from the user or the output of other functions. Please keep going until the user's query is completely resolved, before ending your turn and yielding back to the user. Only terminate your turn when you are sure that the problem is solved.

If you are not sure about file content or codebase structure pertaining to the user's request, use your tools to read files and gather the relevant information: do NOT guess or make up an answer.

You MUST plan extensively before each function call, and reflect extensively on the outcomes of the previous function calls. DO NOT do this entire process by making function calls only, as this can impair your ability to solve the problem and think insightfully.

The timestamp can be formatted as 'HH:MM:SS' or 'MM:SS'.

// user prompt
Carefully read timestamp and narration in the following script. Pay attention to the cause and sequence of events, the detail and movement of objects and the action and pose of persons.

You will be given a subject registry (which could be inaccurate) in this video and a video-level coarse understanding of the question. Please output your think process and use corresponding functions if you can not directly find answer from the given information.

+ CLIP_SEARCH_WITH_TIME_PLACEHOLDER If you need to find details without knowing the specific timespot, please use 'ClipSearch'.

+ If the observed / searched information does not include accurate question-relevant content, e.g. 'unkown' name or other cases, use 'FrameInspect' to ask a fine-grained question within a list of the specified time ranges (list[tuple[hhmmss, hhmmss]]).

+ If you can not make sure the answer by searched content, you should use 'FrameInspect' to query related content in specified time ranges.

+ If you have found answer in a specific script, you should use 'FrameInspect' to ask a fine-grained question to **confirm** it.

You can use multiple times of these functions to prgressively find the answer.

Based on your observations and returned result from functions, select the best option that accurately addresses the question.

Question: [question]

Answer with the option's letter from the given choices directly and only give the best option.

// history messages in the reasoning chain
[history messages]

```

Table 10: Prompts for reasoning model $M_{\text{reasoning}}$.

```

// system prompt
You are a helpful assistant to answer questions.

// user prompt
Carefully watch the video frames. Pay attention to the cause and sequence of events, the detail and movement of objects and the action and pose of persons.

Based on your observations, if you find content that can answer the question, select the best option that accurately answers the question. If no relevant content is found within the given time range, return: 'Error: Cannot find corresponding result in the given time range.'.

Question: [question]

```

Table 11: Prompts for fine-grained detail query M_{tool} in Frame Inspect tool.

```

// system prompt
You are a helpful assistant to answer questions.

// user prompt
Carefully watch the video frames. Pay attention to the cause and sequence of events, the detail and movement of objects and the action and pose of persons.

Based on your observations, provide detailed insights to help interpret the question and video. If you find content that answers the question, suggest a possible answer. If no relevant content is found in the sampled frames, returns 'Error: Cannot find corresponding result in sampled frames.'. Format the answer into a single paragraph.

Question: [question]

```

Table 12: Prompts for query in Global Browse tool.

```

[
  {
    "function": {
      "name": "global_browse",
      "description": "Obtain a subject registry and relevant information about the query from the whole video.\n\nReturns:\nstr: A formatted subject registry includes multiple identical attributes of subjects and a narration about relevant information about the query.\n",
      "parameters": {
        "properties": {
          "database": {
            "$ref": "#/$defs/NanoVectorDB",
            "description": "The database object that supports querying with embeddings."
          },
          "query": {
            "description": "A textual description of the question to search for.",
            "type": "string"
          }
        },
        "required": [
          "database",
          "query"
        ],
        "type": "object"
      }
    },
    "type": "function"
  },
  {
    "function": {
      "name": "clip_search",
      "description": "Searches for events in a video clip database based on a given event description and retrieves the top-k most relevant video clip captions.\n\nReturns:\nstr: A formatted string containing the concatenated captions of the searched video clip scripts.\n\nNotes:\n- This function utilizes the OpenAI Embedding Service to generate embeddings for the input text.\n- Use default values for 'top_k' to limit the number of results returned.",
      "parameters": {
        "properties": {
          "database": {
            "$ref": "#/$defs/NanoVectorDB",
            "description": "The database object that supports querying with embeddings."
          },
          "event_description": {
            "description": "A textual description of the event to search for.",
            "type": "string"
          },
          "top_k": {
            "default": 16,
            "description": "The maximum number of top results to retrieve. Just use the default value.",
            "type": "integer"
          }
        },
        "required": [
          "database",
          "event_description"
        ],
        "type": "object"
      }
    },
    "type": "function"
  }
]

```

Table 13: Function JSON schema for OpenAI model's tool use. Part 1/2.

```

[
  {
    "function": {
      "name": "frame_inspect",
      "description": "Crop the video frames based on the time ranges and ask the model a detailed question about the cropped video clips.\nReturns:\nstr: The model's response to the question.\nIf no relevant content is found within the time range,\nreturns an error message: \"Error: Cannot find corresponding result in the given time range.\"",
      "parameters": {
        "properties": {
          "database": {
            "$ref": "#/$defs/NanoVectorDB",
            "description": "The database containing video metadata. Must be an instance of NanoVectorDB."
          },
          "question": {
            "description": "The specific detailed question to ask about the video content during the specified time ranges. No need to add time ranges in the question.",
            "type": "string"
          },
          "time_ranges_hhmmss": {
            "description": "A list of tuples containing start and end times in HH:MM:SS format.\nIf the time range is longer than 50 seconds, the function samples 50 evenly distributed frames. Otherwise, it uses all frames within the specified range.",
            "items": {
              "items": {},
              "type": "array"
            },
            "type": "array"
          }
        },
        "required": [
          "database",
          "question",
          "time_ranges_hhmmss"
        ],
        "type": "object"
      }
    },
    "type": "function"
  },
  {
    "function": {
      "name": "answer",
      "description": "Call this function after confirming the answer of the user's question, and finish the conversation.",
      "parameters": {
        "properties": {
          "answer": {
            "description": "Answer to the user's question.",
            "type": "string"
          }
        },
        "required": [
          "answer"
        ],
        "type": "object"
      }
    },
    "type": "function"
  }
]

```

Table 14: Function JSON schema for OpenAI model's tool use. Part 2/2.

Question

Question: What type of video is this?
(A) A technology documentary
(B) A news report
(C) A travel documentary
(D) A travel vlog

Agent (Step 1)

Calling tool **Global Browse**.

Global Browse**Subjects Information :**

```
...
"subject_2": {
    "name": "industrial_robot",
    "appearance": ...,
    "identity": [
        "Industrial automation robot, performing repetitive manufacturing or assembly
        line tasks"
    ],
},
...
"subject_10": {
    "name": "digital_face_scan_display",
    "appearance": ...,
    "identity": [
        "Computer interface for facial recognition and identification, used for matching
        and analyzing faces in security or surveillance contexts"
    ],
},
...
"subject_21": {
    "name": "Pedro Domingos",
    "appearance": ...,
    "identity": [
        "Academic expert and commentator, providing context on the Go match and
        public expectations"
    ],
},
...
```

Event Information :

The montage shows extended, polished coverage of artificial-intelligence themes: the AlphaGo challenge match and Google DeepMind branding, autonomous trucks on desert highways, . . . Therefore, the video is best classified as a **technology documentary**.

Agent (Step 2)

Executing **Answer**: (A) A technology documentary. → **Right**

Figure 4: Case study of **Global Browse Only** behavior.

Question

- Question: What happens when the villain's mirror breaks?
- (A) The mirror fragments automatically fuse on the ground into a whole mirror again
 - (B) The mirror fragments cut the villain's hand
 - (C) The mirror fragments are thrown away by the villains
 - (D) The mirror fragments fly into the sky

Agent (Step 1)

Calling tool **Global Browse**.

Global Browse

Subjects Information :

```
...
"subject_18": {
  "name": "Baron Badblood",
  "appearance": ...,
  "identity": [
    "Villainous character, shown invoking his magic mirror to reveal ..."
  ],
  ...
}
```

Event Information :

The scene shows the villain ...; the next shots focus on the villain angrily flinging the pieces aside so that the multicolored shards clatter to the ground and lie scattered around his lair; ... ; therefore the **mirror fragments are thrown away by the villains**.

Agent (Step 2)

Calling tool **Clip Search** with query "villain mirror breaks shards thrown away."

Clip Search

Searched Video Clip Scripts :

```
[00:03:7.0, 00:03:20.0] ... Baron Badblood stands before a large, ornate mirror ...
...
[00:09:46.0, 00:10:6.0] ... Baron strikes the mirror with a sword, causing it to shatter. The shards explode outward, tumbling and spinning through the air as they scatter ...
...
[00:23:52.0, 00:24:4.0] ... The mirror shimmers with swirling pink light, ...
```

Agent (Step 3)

Calling tool **Frame Inspect** at ["00:09:46", "00:10:06"]: "What do the broken pieces of the mirror do after Baron Badblood smashes it?"

Frame Inspect

Fine-grained Response :

The fragments come to life: they quiver, gather into a swirling flock and fly out of the room, **sailing away into the sky**.

Agent (Step 4)

Executing **Answer**: (D) The mirror fragments fly into the sky. → **Right**

Figure 5: Case study of **Simple Action** behavior.

Question

Question: What does the man with the mustache do when boy, cat, and dog carrying treasure chest, shovel and pickaxe back?

- (A) He chases a cat
- (B) He chases a cow
- (C) He chases a dog
- (D) He chases a bird

Agent (Step 1)

Calling tool **Global Browse**.

Global Browse

Subjects Information : ...

Event Information :

Among the night-time forest shots the boy, cat and dog are shown tramping back along the path laden. The black-moustached postman suddenly rushes out and runs after Sharik the dog, trying to **grab the dog**, while the cat and boy keep hold of the chest and tools.

Agent (Step 2)

Calling tool **Clip Search** with query "man with mustache chases dog while boy cat carrying treasure chest".

Clip Search

Searched Video Clip Scripts :

...
[00:08:0.0, 00:08:20.0] ... the **boy** pushes a cart carrying a large, padlocked chest, followed closely by the **cat and dog**, each carrying a pickaxe and shovel ... Overhead, a **black crow** soars across the sky ... Meanwhile, the tall thin **man rushes after the group**, ...
...

Agent (Step 3)

Calling tool **Frame Inspect** at ["00:08:00", "00:08:24"]: "Describe exactly what the tall thin adult man does as the boy, cat, and dog carry the treasure chest, shovel, and pickaxe back along the path. Is he chasing anyone specifically?"

Frame Inspect

Fine-grained Response :

As the boy, cat, and dog come back, the man appears on the same path.

1. He is sprinting, coat and scarf flying, eyes fixed on the sky.
2. His arms reach up as he runs, grabbing at the air.
3. He **never even looks at the boy, the animals or the chest**; instead he is clearly trying to **reach the black crow**.
4. The man leaps and claws at the air, following the crow's flight.

So he is not chasing the boy, the cat, the dog or the treasure chest at all; he is chasing only the crow that is carrying the white object.

Agent (Step 4)

Calling tool **Clip Search** with query "mustached man chases dog".

Clip Search

Searched Video Clip Scripts :

...
[00:33:12.0, 00:33:32.0] ... The cat darts away down the sandy road, desperate to escape, while the **dog** quickly recovers and resumes the **chase**, bounding after the cat ...
...

Agent (Step 5)

Calling tool **Clip Search** with query "man with mustache chases crow".

Clip Search

Searched Video Clip Scripts :

...
[00:08:28.0, 00:08:48.0] ... The **man** points accusatorily at the boy and the crow, prompting the boy to hand over a shiny white object which **the crow had stolen earlier** ...
...

Agent (Step 6)

Calling tool **Frame Inspect** at ["00:08:06", "00:08:14"]: "Does the man who is running after the crow have a visible mustache? Describe his facial hair."

Frame Inspect

Cannot find corresponding result in the given time range.

Agent (Step 7)

Calling tool **Frame Inspect** at ["00:08:06", "00:08:14"]: "Does the tall thin man chasing the crow have a mustache, beard, or is he clean-shaven?"

Frame Inspect

Fine-grained Response :

He has a **mustache**.

Agent (Step 8)

Executing **Answer**: (D) He chases a bird. → **Right**

Figure 6: Case study of **Iterative Search** behavior.

Question

Question: Why does the tall man cut off the chains at 21:49?
(A) To open the gate
(B) To blow out the candles
(C) To trigger a trap
(D) To help the character

Agent (Step 1)

Calling tool **Global Browse**.

Global Browse

Subjects Information : ...

Event Information :

The tall man steps up to a hanging length of chain that holds the barred door of a cell shut, raises his sword and slices clean through the links; the chain clatters free and the cell door can now be pushed open, allowing the **wounded protagonist inside to be supported out**.

Agent (Step 2)

Calling tool **Frame Inspect** at ["00:21:40", "00:21:58"]: "Why does the tall man cut off the chains?"

Frame Inspect

Fine-grained Response :

He severs the ceiling chains so the **candle-chandelier comes crashing down**, plunging the corridor into darkness, **giving his companion a fighting chance to survive the attack**.

Agent (Step 3)

Calling tool **Frame Inspect** at ["00:21:45", "00:21:55"]: "What happens immediately after the tall man slices through the chains hanging from the ceiling?"

Frame Inspect

Fine-grained Response :

The chain he cuts **holds the candle-chandelier**; as soon as it is severed, the chandelier comes crashing down from the ceiling onto the floor.

Agent (Step 4)

Calling tool **Frame Inspect** at ["00:21:40", "00:22:10"]: "Does the tall man ever cut a chain to open a gate or door around 21:49?"

Frame Inspect

Fine-grained Response :

No.

Agent (Step 5)

Executing **Answer**: (C) To trigger a trap. → **Wrong**

Figure 7: Case study of **Frame Inspect Trap** behavior.

Question

Question: What does the protagonist see through the window after she is taken to the utility room?

- (A) A group of monks sitting cross-legged in the snow
- (B) A group of citizens chatting together
- (C) A group of warriors practicing swords
- (D) A group of samurais eating

Agent (Step 1)

Calling tool **Global Browse**.

Global Browse

Subjects Information : ...

Event Information :

Inside the cramped utility room the heroine peers out through the lattice window and her eyes settle on the snow-covered courtyard where **several monks sit cross-legged** and bare-chested despite the freezing weather, engaged in silent meditation.

Agent (Step 2)

Calling tool **Clip Search** with query "utility room window sees group monks cross-legged snow".

Clip Search

Searched Video Clip Scripts :

...
[00:19:31.0, 00:19:42.0] ... As **soft snowflakes** drift through the pale morning air, the solitary traveler appears on a frosted hill overlooking a vast, traditional Japanese town ...

Agent (Step 3)

Calling tool **Clip Search** with query "utility room protagonist looks through window sees".

Clip Search

Searched Video Clip Scripts :

...
[00:02:2.0, 00:02:7.0] ... Dust motes drift lazily through beams of sunlight streaming from the paper **window**, highlighting the wooden shelves lined with bowls ...
...

Agent (Step 4 - Step 14)

Calling tool **Clip Search** with queries :
"utility room small storage taken escorted feed him utility",
"utility room window look out",
...
"storeroom small room snow see out window"

Clip Search

Searched Video Clip Scripts :

...

Agent (Step 15)

Calling tool **Clip Search** with query "utility room".

Clip Search

Searched Video Clip Scripts :

...
[00:17:32.0, 00:17:49.0] ... The setting is dim and somber, likely a **shelter or workshop** with stone flooring and faint light filtering in from the outside ...
...

Agent (Step 16)

Executing **Answer**: (A) A group of monks sitting cross-legged in the snow. →
Wrong

Figure 8: Case study of **Clip Search Trap** behavior.