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

Anonymous Author(s)

Affiliation

Address

email

1 In this document, we provide the supplementary material for the proposed Deep Video Discovery
2 (DVD) agent. This includes more details in evaluation and additional experimental results.

3 A Evaluation Details

4 A.1 Prompts

5 We describe the prompts used by different models within the proposed DVD, including: (1) Captioning
6 and subject extraction model M_{database} used in Multi-granular Video Database Construction (Tab. 2);
7 (2) Reasoning model $M_{\text{reasoning}}$ used in Agentic Search and Answer (Tab. 3); (3) Fine-grained detail
8 querying model M_{tool} used in the Frame Inspect tool (Tab. 4); (4) event-centric global information in
9 the Global Browse tool (Tab. 5). The JSON schema for tool use of OpenAI model are also provided in
10 Tab. 6 and Tab. 7.

11 A.2 Cost

12 We measure the average API cost of our agent when calling OpenAI o3 on LVBench. On average,
13 DVD incurs a cost of \$2.05 with 20803 tokens per question.

14 A.3 Azure OpenAI Service

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

19 A.3.1 Maximum Image Count Limitation

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

26 **For DVD**, we horizontally splice up to 5 frames into a single image. This allows us to input up to
27 250 frames (50 images \times 5 frames). Note that we only use the 250 frames setting in event-centric
28 global information gathering in the Global Browse tool while maximum 50 frames are used in the
29 Frame Inspect tool.

Table 1: Statistics for content filtering in Azure OpenAI API.

Methods	LVBench Overall	LongVideoBench (Val) Overall	Long	Video MME Long (w/o sub)	EgoSchema 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

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

33 A.3.2 API Content Filtering

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

37 **For OpenAI o3**, when a query is blocked, we randomly select an answer from the remaining
38 candidates. Across benchmarks, approximately 10% to 25% of queries are filtered, as shown in Tab. 1.
39 To better assess true performance, we also report scores on the subset of unfiltered responses, where
40 performance improves significantly.

41 **For DVD**, when VLM returns none due to the content filtering, we do not implement additional error
42 handling for the agent. Instead, we let the agent make its own decisions to handle error. It may invoke
43 alternative tools or attempt to answer based on available information. If a captioning query is blocked,
44 we leave the corresponding entry blank in the database. While this affects database completeness, the
45 agent remains operational due to its adaptive behavior. Tab. 1.

46 B More Results

47 B.1 Statistical Significance

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

51 B.2 Case Study

52 We provide illustrative examples of the tool-calling process for different agent behaviors described
53 in Section 4.5 of the paper. All examples are from LVBench [2]. Each example highlights how the
54 agent selects actions and invokes tools to gather information for answering the question, and how
55 each tool call retrieves relevant data from the database. In the tool responses, we highlight **useful and**
56 **correct information** in orange and **useless or incorrect information** in purple.

57 **Global Browse Only** is shown in Fig. 1. When asked about the type of video, the agent directly
58 invokes Global Browse to retrieve global context. The subject-level browsing reveals many technical
59 subjects, while the event-level browsing summarizes the key themes and categorizes the video as a
60 technology documentary. Based on this information, the agent correctly classifies the video into a
61 technology documentary.

62 **Simple Action** is shown in Fig. 2. In this case, Global Browse incorrectly indicates that the mirror
63 fragments are thrown away. To verify, the agent uses Clip Search to locate the time range of the
64 mirror breaking event and then calls Frame Inspect for finer-grained analysis. It discovers that the
65 fragments actually fly into the sky and thus selects the correct answer.

66 **Iterative Search** is shown in Fig. 3. This question involves multiple subjects (boy, dog, cat, bird,
67 and man), making it more complex. Global Browse initially suggests that the man is chasing the
68 dog. However, the first round of Clip Search and Frame Inspect indicates that he is chasing a bird.
69 To resolve the ambiguity, the agent issues two more Clip Search queries for events involving "man
70 chasing dog" and "man chasing bird". The search yields relevant and useful results only for the later,
71 suggesting that the man is actually chasing the bird. To confirm the subject's identity, the agent
72 checks for the presence of a mustache. Convinced that the mustached man is chasing the bird, the
73 agent selects the correct option. Notably, when a Frame Inspect call fails to find relevant results in
74 step 6, the agent can automatically ask another question to obtain an useful response.

75 **Frame Inspect Trap** is shown in Fig. 4. This question targets an event at a specific timestamp.
76 Around this time, the agent asks three consecutive questions. While Frame Inspect sometimes
77 captures accurate information, it occasionally fails to interpret frames correctly, providing misleading
78 responses. As a result, the agent cannot discern which information is reliable and ultimately answers
79 incorrectly.

80 **Clip Search Trap** is shown in Fig. 5. This example relies on identifying the correct time range using
81 key terms like "protagonist" and "utility room." However, the database contains few directly relevant
82 captions or subjects. Despite the agent's attempts to rephrase and refine its Clip Search queries, the
83 tool fails to return relevant results, leading to an incorrect answer.

84 **C Broader Impacts**

85 **C.1 Positive Impacts**

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

89 **C.2 Negative Impacts**

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

```

// 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": <timestamp>
    },
    ...
  },
  "clip_description": {
    "subjects_present": [<subject_id_list>],
    "captions": <a smooth and very detailed narration about the whole video clip>
  }
}

```

Table 2: Prompts for captioning and subject extraction model M_{database} in video database construction.

```

// 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 as
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. 'unknown'
name or other cases, use 'FrameInspect' to ask a fine-grained question within a list of the specified
time ranges (list[tuple[h:mm:ss, h:mm:ss]]).

+ 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 progressively 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 3: 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 4: 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 5: 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:\n str: 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:\n str: 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 6: 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:\n str: The model's response to the question. If no relevant content is found within the time range,\n returns 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. If 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 7: Function JSON schema for OpenAI model's tool use. Part 2/2.

94

Question

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

95

Agent (Step 1)

Calling tool **Global Browse**.

96

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**.

97

Agent (Step 2)

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

Figure 1: 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

98

Agent (Step 1)

Calling tool **Global Browse**.

99

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.

00

00

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

01

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, ...

02

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?"

03

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**.

04

Agent (Step 4)

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

05

Figure 2: 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".

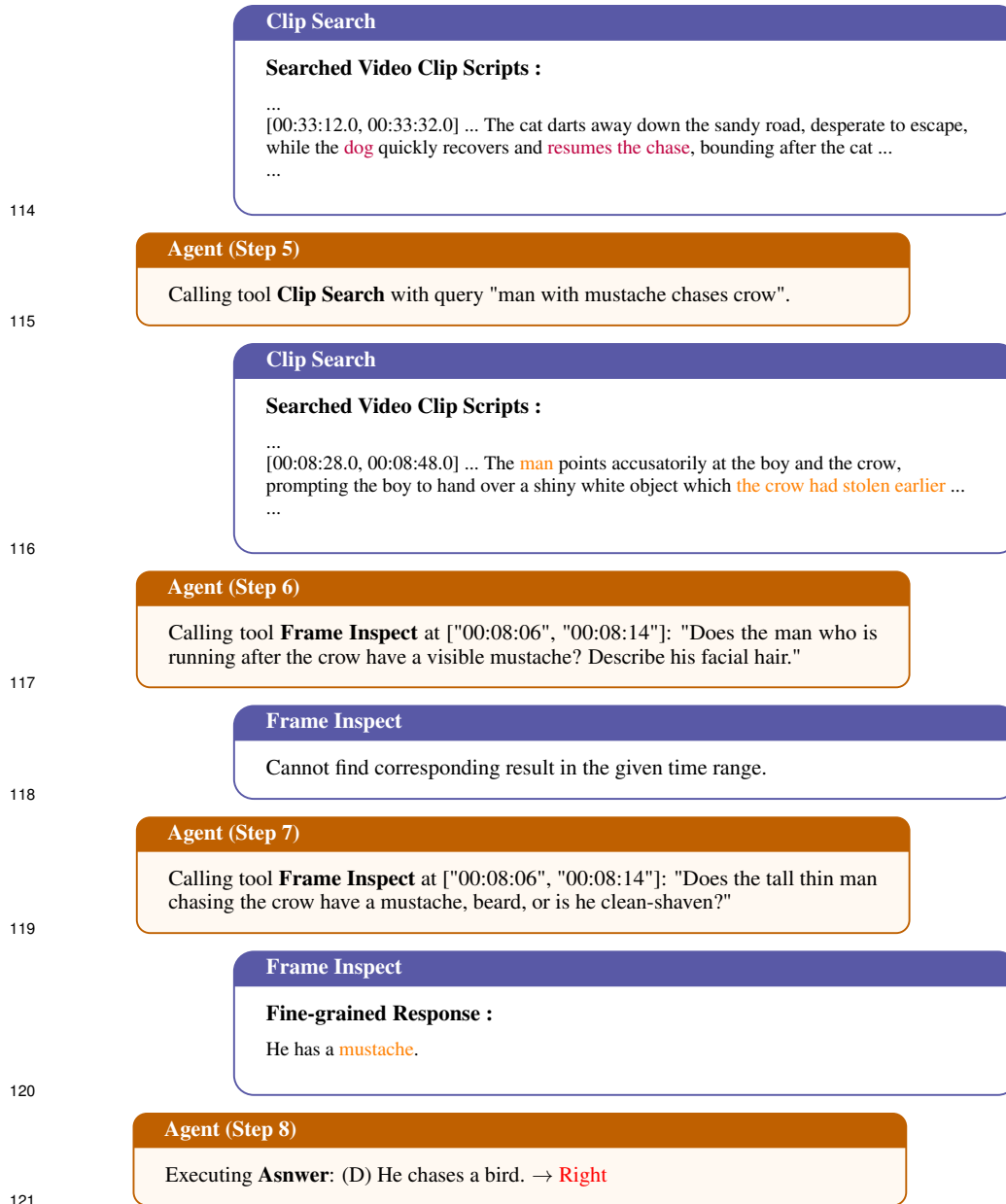


Figure 3: Case study of **Iterative Search** behavior.

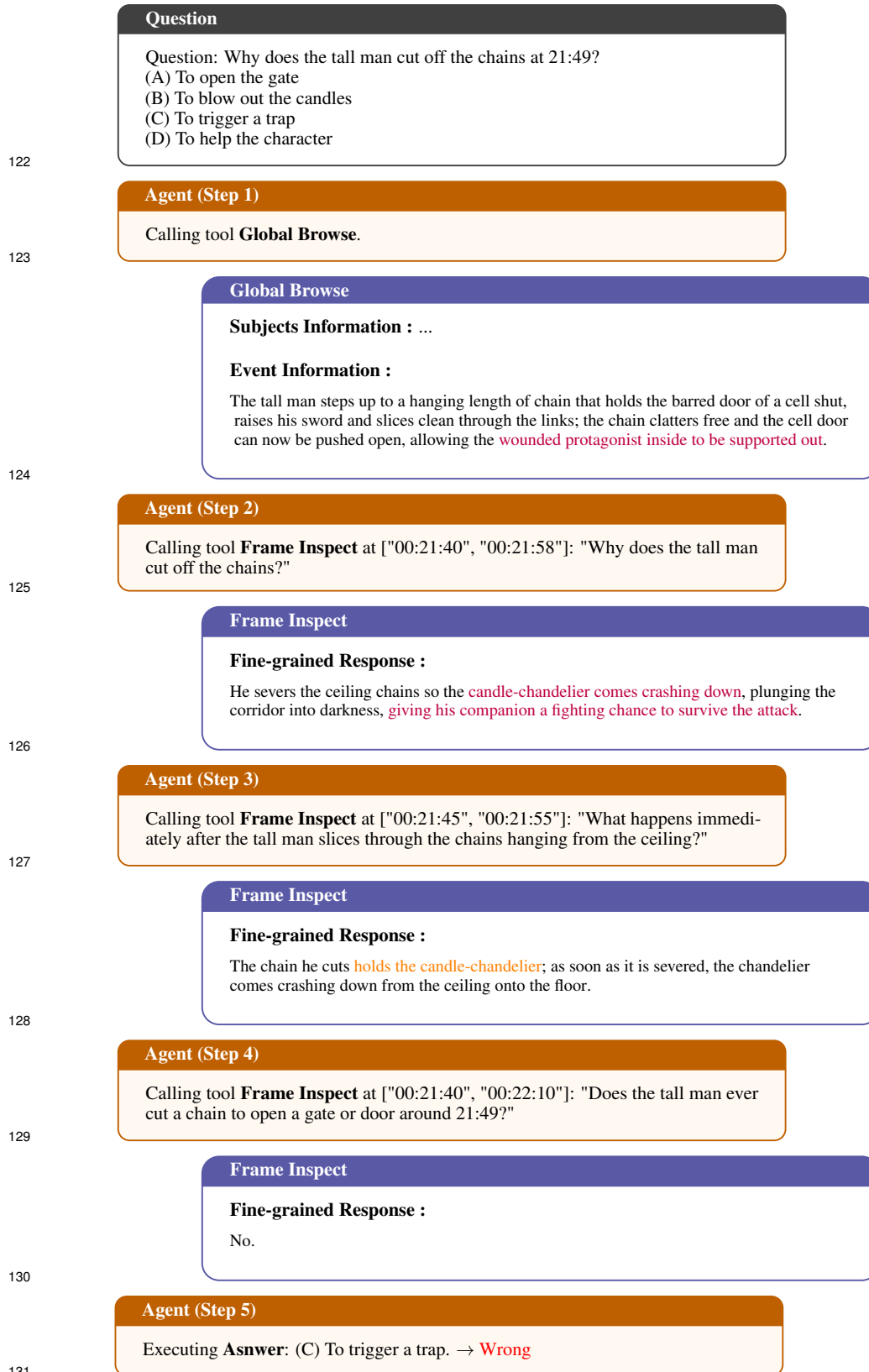


Figure 4: Case study of **Frame Inspect Trap** behavior.

132

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

133

Agent (Step 1)

Calling tool **Global Browse**.

134

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.

135

Agent (Step 2)

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

136

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 ...

137

Agent (Step 3)

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

138

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 ...
...

139

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"

140

Clip Search

Searched Video Clip Scripts :

...

141

Agent (Step 15)

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

142

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 ...
...

143

Agent (Step 16)

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

Figure 5: Case study of **Clip Search Trap** behavior.

144 **References**

- 145 [1] OpenAI. Introducing OpenAI o3 and o4-mini. [https://openai.com/index/](https://openai.com/index/introducing-o3-and-o4-mini/)
146 [introducing-o3-and-o4-mini/](https://openai.com/index/introducing-o3-and-o4-mini/), 2025. Accessed: 2025-05-15.
- 147 [2] W. Wang, Z. He, W. Hong, Y. Cheng, X. Zhang, J. Qi, X. Gu, S. Huang, B. Xu, Y. Dong, et al.
148 LVBench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*,
149 2024.