VideoSleuth: A Narrative-Centric Agentic System based on Video-Audio Native MLLM for Long-Form Video Understanding

Conglin Li[1] **Yang Li**[2] **Qi Zhang**[3]
Guanhua Chen[4] **Yun Chen**[5] **Weifeng Ge**[1*]

[1]College of Computer Science and Artificial Intelligence, Fudan University
[2]School of Computer Science and Technology, East China Normal University
[3]Department of Computer Science and Technology, Tongji University
[4]Southern University of Science and Technology
[5]Shanghai University of Finance and Economics

licl25@m.fudan.edu.cn, yli@cs.ecnu.edu.cn
zhangqi_cs@tongji.edu.cn, chengh3@sustech.edu.cn
yunchen@sufe.edu.cn, wfge@fudan.edu.cn

Abstract

Long-form video understanding (LVU) requires models to reason over hour-long narratives while preserving fine-grained spatiotemporal details across extended temporal horizons. Although recent multimodal large language models (MLLMs) have achieved strong performance on short-video tasks, their reasoning capabilities degrade substantially as video duration and information density increase. We present **VideoSleuth**, a narrative-centric agentic framework for LVU built on video- and audio-native MLLMs. VideoSleuth organizes its memory around the four narrative elements—*time, place, person, and event*—and equips the agent with tools explicitly designed to construct and query these elements within a ReAct-style reasoning loop. Unlike prior LVU agents that rely on dense offline preprocessing or image-only VLMs, VideoSleuth performs perception on demand using a video-audio native perceptual backbone, jointly leveraging visual and auditory cues to support dialogue-based identity grounding, long-range event localization, and fine-grained evidence retrieval. On four standard LVU benchmarks, VideoSleuth consistently outperforms prior on-demand grounding agents and achieves competitive accuracy with the strongest dense-preprocessing baseline, while consuming at least $5.2\times$ fewer end-to-end tokens than DVD-style pipelines spend on offline preprocessing in the single-question setting. [2]

1 Introduction

Long-form video understanding (LVU) underpins a growing range of real-world applications, including film analysis, intelligent surveillance, sports commentary, and multimodal assistants. Unlike short clips, hour-long videos contain densely structured information unfolding over extended temporal horizons, requiring models to simultaneously capture fine-grained spatiotemporal details, track

*Corresponding author: wfge@fudan.edu.cn.
[2]Code: https://github.com/SoftEgLi/VideoSleuth.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

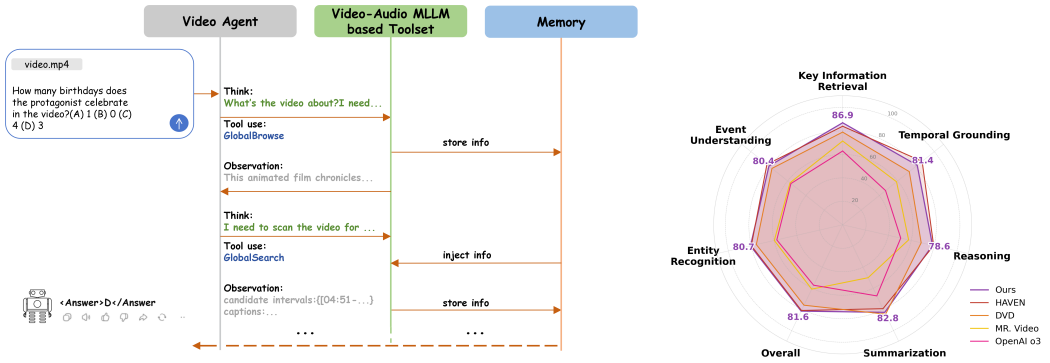


Figure 1: **Left:** Illustration of the **VideoSleuth** agent, which iteratively reasons over a user query, dispatches Video-Audio MLLM-based tools to gather evidence, and commits a final answer. **Right:** Performance comparison on LVBench across six fine-grained capability dimensions.

evolving entities, and reason about long-range causal and narrative dependencies. Although recent multimodal large language models (MLLMs) have achieved remarkable progress on short-video tasks, their instruction-following and reasoning capabilities degrade substantially as input duration and information density increase, leaving accurate LVU a fundamental yet largely unresolved challenge.

To overcome the context-length and computational bottlenecks of monolithic MLLMs, recent work increasingly adopts *agentic* paradigms that decompose LVU into iterative cycles of planning, evidence gathering, and reasoning [1–5]. Within this paradigm, the central challenge becomes *efficient and reliable localization of question-relevant evidence* in long videos. Existing approaches largely fall into two categories.

(i) Retrieval over densely preprocessed memories. A first family of methods [3, 5–7] preprocesses videos offline by partitioning them into short clips and invoking a VLM to construct a retrieval index for inference. While amortizing perception into an offline memory yields strong accuracy, the preprocessing stage incurs substantial token cost that, in systems such as DVD [3], dominates the overall computation budget. Moreover, the captioner's biases are irreversibly baked into the constructed memory.

(ii) On-demand LLM-driven grounding. A second family [2, 1, 4] avoids dense preprocessing and instead allows an LLM to iteratively inspect the video on demand. This paradigm substantially reduces token consumption, but to keep inference tractable, they need to progressively refine toward candidate intervals. Such coarse-to-fine strategies risk converging to suboptimal evidence when early sparse samples miss answer-critical content. In addition, prior methods in this paradigm rely almost exclusively on *image-only* VLMs, making tasks that require joint vision–audio reasoning—such as binding dialogue-mentioned names to on-screen faces—particularly challenging.

These limitations suggest that the missing ingredient is not a new agentic paradigm, but a stronger perceptual foundation for it. Recent video-audio native MLLMs jointly process visual and auditory streams within a single forward pass and leverage dedicated video encoders that model temporal continuity far more efficiently than frame-based image inputs. This efficiency enables minute-scale video segments to be processed at meaningful frame rates rather than through sparse global snapshots, mitigating the local-optimum failure mode of existing on-demand grounding systems. At the same time, joint vision–audio conditioning naturally supports operations such as name–appearance binding and audio-anchored event localization.

Motivated by these observations, we propose **VideoSleuth**, a narrative-centric agentic framework for long-form video understanding built on top of video-audio native MLLMs. Rather than organizing memory around clip indices alone, VideoSleuth structures its memory around the four classical narrative elements—*time, place, person, and event*—and equips the agent with tools explicitly designed to construct and query these elements on demand. The agent operates in a ReAct-style loop, dynamically dispatching perception only when required by the question.

Contributions. Our contributions are summarized as follows:

1. We propose a narrative-centric agentic architecture for LVU that organizes memory around the four narrative elements—time, place, person, and event—and couples this memory with a structured toolset (PERSONREGISTER, GLOBALBROWSE, GLOBALSEARCH, CLIPANALYZE, MULTICLIPJUDGE) for narrative-aware evidence construction and retrieval.
2. We are, to our knowledge, the first to integrate a video-audio native MLLM as the perceptual backbone of a long-video agent, enabling joint vision–audio reasoning for dialogue-based identity grounding, long-range event localization, and fine-grained evidence retrieval, etc.
3. On four standard LVU benchmarks (LVBench, LongVideoBench, Video-MME, and EgoSchema), VideoSleuth substantially outperforms prior on-demand grounding agents while matching or surpassing dense-preprocessing methods, achieving this with at least $5\times$ lower token consumption than DVD-style pipelines spend on offline preprocessing in the single-question setting.

2 Related Work

LVU with Densely Preprocessed Memories. Agentic and retrieval-augmented methods [6, 7, 5, 3, 8–10] preprocesses videos offline by partitioning them into short clips and invoking VLMs with dense sampling to construct a retrieval memory for downstream reasoning. At inference time, questions are answered primarily by retrieving from this precomputed index. While amortizing perception into an offline memory yields strong accuracy, the dense captioning stage incurs substantial token cost that, in systems such as DVD [3], dominates the overall computation budget. Moreover, biases introduced during caption generation become irreversibly embedded in the constructed memory. In contrast, VideoSleuth performs perception on demand, avoiding the linearly-scaling offline captioning cost and the irreversible captioner biases baked into precomputed retrieval indices.

LVU with On-Demand Grounding Agents. To avoid dense preprocessing, methods in [11, 2, 1, 4, 12, 13]rely on LLM-driven reasoning to iteratively inspect videos on demand. VideoSeek [2] organizes tools by temporal granularity, VideoARM [1] separates scope estimation from semantic understanding, and Symphony decomposes grounding into LLM-based query expansion and VLM-based segment scoring. These approaches substantially reduce token consumption and adapt their reasoning dynamically to the question. However, most systems rely on sparse global sampling followed by progressive refinement toward candidate intervals. Such coarse-to-fine strategies risk converging to suboptimal evidence when early sparse samples miss answer-critical content. VideoSleuth differs from previous methods in that it replaces sparse-frame, image-only VLM inspection with a video-audio native MLLM that performs dense segment sweeps, and replaces clip-indexed memory with a narrative-centric memory organized around time, place, person, and event.

Video-Audio Native MLLMs. Some video- and aduio-native MLLMs that treat video as a first-class modality and jointly condition on visual and auditory streams [14–17]. By processing audio with video rather than as a separate post-processing signal, these models preserve temporal continuity more naturally and support stronger alignment between spoken and visual content. Dedicated video encoders further enable minute-scale segments to be processed at meaningful frame rates with higher efficiency than frame-based image inputs. Despite these advances, existing long-video agents [3, 2, 1, 5, 4] continue to rely on image-VLMs over sampled frames. To our knowledge, VideoSleuth is the first agentic LVU framework built on top of a video-audio native MLLM.

3 Method

Overview. We instantiate our framework as **VideoSleuth**, an LLM-driven agent illustrated in Figure 2. Given a long video $\mathcal{V}$ of duration T and a question q, the goal is to produce an answer by iteratively gathering and reasoning over multimodal evidence. At a high level, VideoSleuth follows an agentic paradigm in which a controller LLM Φ_{L} decomposes the task into a sequence of tool calls within a ReAct-style loop. The toolset contains five tools {`PersonRegister`, `GlobalBrowse`,

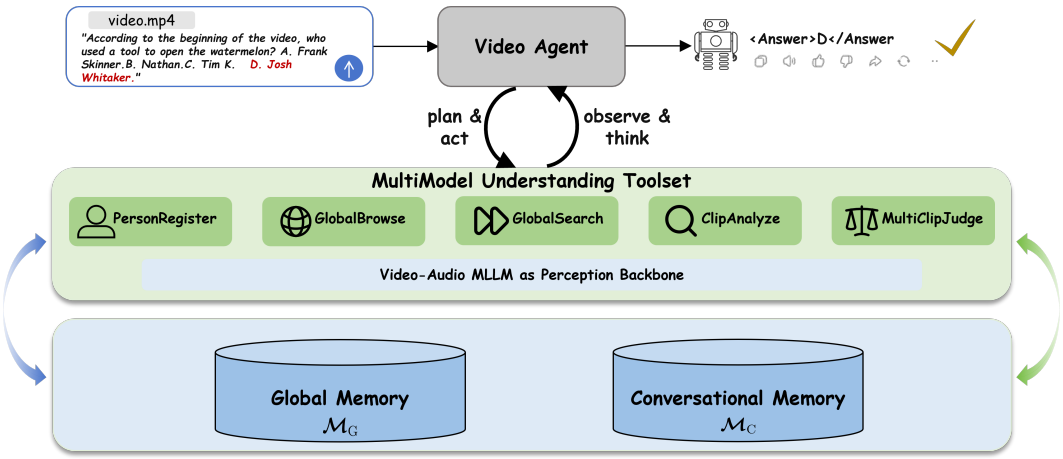


Figure 2: The paradigm of **VideoSleuth**. Given a user question and an input video, the video agent operates in a ReAct paradigm, leveraging the LLM's tool-use and reasoning abilities to invoke a multimodal understanding toolset backed by Video-Audio MLLMs. The resulting evidence is accumulated in conversational memory, from which the agent reasons to produce the final answer.

`GlobalSearch`, `ClipAnalyze`, `MultiClipJudge`}. At each step, the controller selects an appropriate tool, invokes a perceptual module to acquire evidence, and updates its internal state based on the returned observations. To support this process, the agent maintains a structured memory $\mathcal{M} = \mathcal{M}_G \cup \mathcal{M}_C$ with two complementary roles. The **global memory** $\mathcal{M}_G$ contains long-lived, reusable information (e.g., video-level summaries and entity profiles), while the **conversational memory** $\mathcal{M}_C$ accumulates intermediate observations and reasoning traces specific to the current question. All perception is performed on demand by a **video-audio native MLLM** Φ_V, which directly processes video segments together with auxiliary signals to produce query-conditioned outputs. This design avoids dense offline preprocessing and enables the agent to adaptively allocate computation.

3.1 Agent Workflow

Given a question q, the agent first determines whether q references one or more concrete named entities or instead requires general reasoning over scene- or event-level content. This routing decision governs whether the global memory $\mathcal{M}_G$ is pre-populated with person profiles before plot reasoning.

Person Profiling. The agent parses q to identify any named persons implicated by the question. For each extracted $name$, the agent dispatches `PersonRegister`$(name; \mathcal{V})$ to construct a corresponding profile and commits it to $\mathcal{M}_G$ prior to any plot-level reasoning. When q contains no named entities—as in scene-anchored or event-anchored questions—this stage is skipped and $\mathcal{M}_G$ is left empty. Performing person profiling up front ensures that tagged reference images are available to every subsequent perception call, where they are injected as visual exemplars into the Video-Audio MLLM prompt, propagating appearance information without lossy textual paraphrasing.

Plot Reasoning. The agent then enters a ReAct-style loop in which the controller LLM decomposes q into a sequence of tool calls drawn from {`GlobalBrowse`, `GlobalSearch`, `ClipAnalyze`, `MultiClipJudge`}, conditioned on q, $\mathcal{M}_G$, and $\mathcal{M}_C$. Rather than following a fixed pipeline, the controller selects the next tool at each round based on the evidence still missing for q, and the loop terminates once $\mathcal{M}_C$ provides sufficient grounding to commit an answer. Figure 4 illustrates this workflow through a 6-step trace over a 1-hour news broadcast.

3.2 Memory Structure

We now formalize the two complementary memory stores introduced above, whose access patterns reflect a separation between long-lived narrative context and question-specific reasoning state:

$$\mathcal{M} = \underbrace{\mathcal{M}_{\mathrm{G}}}_{\text{global memory}} \cup \underbrace{\mathcal{M}_{\mathrm{C}}}_{\text{conversational memory}}. \tag{1}$$

The **global memory** $\mathcal{M}_{\mathrm{G}}$ holds two long-lived items that are reusable across all downstream tool calls: a video-level *global summary* sum capturing genre, overarching setting, and narrative arc (produced by `GlobalBrowse`), and a set of *person profiles* p produced by `PersonRegister`:

$$\mathcal{M}_{\mathrm{G}} = \{\, sum \,\} \cup \{\, p_i = (\text{name}_i,\, \text{img}_i,\, \text{desc}_i,\, t_i) \,\}_{i=1}^{len(p)}, \tag{2}$$

where img_i is a tightly cropped reference image of person i, desc_i is a short description, and $t_i \in [0, T]$ denotes the timestamp of the image. Whenever a downstream tool issues a Video-Audio MLLM call, all the $(\text{name}_i, \text{img}_i)$ pairs, along with sum are injected as the global information.

The **conversational memory** $\mathcal{M}_{\mathrm{C}}$ stores the controller's intermediate reasoning steps together with the outputs of all tool calls invoked during the agent's reasoning loop. We highlight that `GlobalSearch` contributes query-conditioned outputs at two granularities per segment: (i) a *concise per-segment caption* organized around the four narrative elements, and (ii) a set of *candidate relevant intervals* indicating where within the segment the answer-relevant content is located, which the controller may subsequently verify with `ClipAnalyze` or `MultiClipJudge`.

3.3 Toolset with Details

Two lightweight auxiliary signals are precomputed once per video and consumed throughout: an **audio transcript** $\mathcal{A}$ produced by WhisperX, and an **on-screen text stream** $\mathcal{O}$ produced by PaddleOCR (sampled at 1 fps). These signals serve two purposes: (i) `PersonRegister` relies on $\mathcal{A} \cup \mathcal{O}$ to localize the time intervals at which target names are mentioned; and (ii) when supplied alongside the visual and audio streams to the Video-Audio MLLM, they reduce hallucinations on spoken content and on-screen text. Then, we describe the five tools introduced in the workflow above.

The full prompts of all the tools described below are provided in Appendix D.

`PersonRegister`. Given a target name list $\{name_i\}_{i=1}^{N}$, `PersonRegister` constructs a person profile p_i for each $name_i$ from the long video $\mathcal{V}$ and commits the resulting profiles to $\mathcal{M}_{\mathrm{G}}$:

$$\texttt{PersonRegister}(\{name_i\}_{i=1}^{N};\, \mathcal{V}) \longrightarrow \{p_i = (name_i,\, \text{img}_i,\, \text{desc}_i,\, t_i)\}_{i=1}^{N}, \tag{3}$$

$$\mathcal{M}_{\mathrm{G}} \leftarrow \mathcal{M}_{\mathrm{G}} \cup \{p_i\}_{i=1}^{N}. \tag{4}$$

Each profile is built in three stages: mention localization, visual confirmation, and reference image extraction.

Step 1: Name Localization. Given a candidate name $name$, our goal is to identify all time intervals in which $name$ is referenced within the ASR transcript $\mathcal{A}$ and OCR stream $\mathcal{O}$. Since surface forms exhibit substantial variability—including nicknames, honorifics, abbreviations, and transliteration variants—exact string matching is insufficient. We therefore delegate this task to an LLM equipped with a localization prompt $prompt_{\text{loc}}$. To amortize inference cost, we perform localization for the entire candidate set $\mathcal{N} = \{name_i\}_{i=1}^{N}$ in a single pass, jointly conditioning on $\mathcal{A}$ and $\mathcal{O}$:

$$\texttt{NameLocalize}(\mathcal{N}, \mathcal{A}, \mathcal{O}) \longrightarrow \{\mathcal{S}_{name_i}\}_{i=1}^{N} = \Phi_{\mathrm{L}}\big(\mathcal{N},\, \mathcal{A},\, \mathcal{O},\, prompt_{\text{loc}}\big), \tag{5}$$

where each $\mathcal{S}_{name_i}$ denotes the set of time intervals during which $name_i$ is mentioned.

Step 2: Video-Audio Grounding via MLLM. Localized intervals from Step 1 indicate where a name is *mentioned*, but not whether the corresponding person is *visually present* on screen. To bridge this gap, we employ a Video-Audio MLLM to verify visual presence and pinpoint the exact moment of clearest appearance. For each candidate $name_i \in \mathcal{N}$, we process its associated intervals $\mathcal{S}_{name_i} = \{[t_k^{\mathrm{s}}, t_k^{\mathrm{e}}]\}_{k=1}^{K_i}$ *sequentially in chronological order*, where each interval is symmetrically padded by $\Delta_{\mathrm{m}} = 10$ seconds to provide sufficient temporal context. Conditioned on a grounding prompt $prompt_{\text{gnd}}$, the MLLM returns three fields: a binary presence indicator, the precise timestamp at which the person appears most clearly, and a concise visual description:

$$\texttt{VAGround}(name_i,\, \mathcal{V}[\tau_k^{\text{pad}}]) \longrightarrow (\textit{present}_k,\, t_k^{*},\, d_k) = \Phi_{\mathrm{V}}\big(name_i,\, \mathcal{V}[\tau_k^{\text{pad}}],\, prompt_{\text{gnd}}\big), \tag{6}$$

where $\mathcal{V}[\tau_k^{\text{pad}}]$ denotes the padded video-audio segment over $\tau_k^{\text{pad}} = [t_k^{\text{s}} - \Delta_{\text{m}},\ t_k^{\text{e}} + \Delta_{\text{m}}]$, $present_k \in \{0, 1\}$ indicates whether the person referred to by $name_i$ is visually present, t_k^* marks the timestamp of clearest appearance as instructed in $prompt_{\text{gnd}}$, and d_k is a short description of the visual attributes.

Step 3: Reference Image Extraction. The goal of this step is to extract a tight reference crop of the person at the timestamp t_k^* identified in Step 2, which will later serve as the visual anchor in our global memory. Since t_k^* predicted by the MLLM may carry small temporal errors, we do not crop directly at t_k^*. Instead, we extract a short window $[t_k^* - 1s,\ t_k^* + 1s]$ around it and sample frames at 5 fps, yielding $J = 10$ candidate frames.

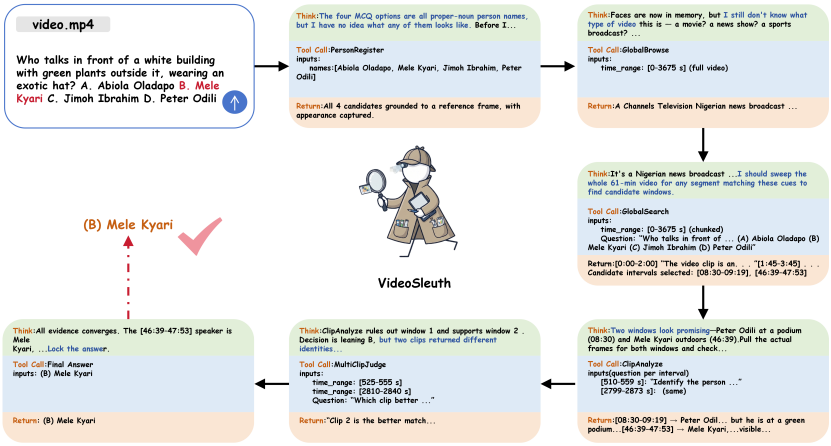


Figure 3: An example workflow of VideoSleuth.

Each frame is then passed to a grounding VLM together with the visual description d_k from Step 2, which returns, for that frame, zero or more bounding boxes around regions matching d_k, each accompanied by a confidence score in $[0, 1]$. We aggregate all (frame, box) pairs across the J frames and select the one with the highest confidence as the final detection. Cropping this region produces the reference image I_k^{ref}.

Finally, we store the tuple $(name_i,\ d_k,\ I_k^{\text{ref}},\ t_k^{\text{ref}})$ in the global memory $\mathcal{M}$, where t_k^{ref} denotes the timestamp of the selected frame.

`GlobalBrowse`. Downstream tools require a coarse but global understanding of the video—its overall genre and the main events across its full duration. To provide this efficiently, we uniformly sample N_{g} frames from the video and tile them into a single grid image G. we use $N_{\text{g}} = 30$ frames arranged in a 6×5 grid. The grid image, together with the full ASR transcript $\mathcal{A}$ and a browsing prompt $prompt_{\text{glob}}$, is passed to the MLLM, which returns the summary of video type and the main events:

$$\texttt{GlobalBrowse}(\mathcal{V},\ \mathcal{A}) \ \longrightarrow\ \text{sum}, \qquad \mathcal{M}_G \leftarrow \mathcal{M}_G \cup \text{sum}. \tag{7}$$

The sum is cached in the global memory $\mathcal{M}$ and, analogous to the per-identity reference crops, injected as additional context into every subsequent call to `GlobalSearch`, `ClipAnalyze`, and `MultiClipJudge`.

`GlobalSearch`. This tool sweeps the entire video at chunk-level granularity, conditioned on user's original query q, and aggregates both relevant time intervals and q-relevant captions across the full video. Concretely, the video is partitioned into chunks of duration L_{c} with an overlap L_{o} between adjacent chunks ($L_{\text{c}} = 120\,\text{s}$ and $L_{\text{o}} = 15\,\text{s}$ in our implementation), yielding a chunk set $\mathcal{C} = \{c_i\}_{i=1}^{N}$ that fully covers the video. For a video of duration T, this yields $N = \lceil (T - L_{\text{o}})/(L_{\text{c}} - L_{\text{o}}) \rceil$ chunks. Each chunk c_i is independently dispatched to the Video-Audio MLLM, which—sampled at $f_{\text{c}} = 1\,\text{fps}$ and conditioned on q together with a search prompt $prompt_{\text{srch}}$—returns two parallel outputs: a single q-relevant caption $caption_i$, which is a free-form textual description narrating *when*, *where*, *who*, and *what* are involved in the moments of c_i that are relevant to q; and a set of relevant

intervals $\mathcal{T}_i = \{(\tau_i^{(s)}, \rho_i^{(s)})\}_s$, where each $\tau_i^{(s)}$ is a time span inside c_i deemed relevant to q and $\rho_i^{(s)}$ is the accompanying rationale. Aggregating across all chunks gives a video-level caption set $\mathit{Cap} = \{\mathit{caption}_i\}_{i=1}^N$ and an interval set $\mathcal{T} = \{\mathcal{T}_i\}_{i=1}^N$:

$$\texttt{GlobalSearch}(\mathcal{V}, q) \longrightarrow (\mathit{Cap}, \mathcal{T}), \qquad \mathcal{M}_C \leftarrow \mathcal{M}_C \cup \{\mathit{Cap}, \mathcal{T}\}. \tag{8}$$

`ClipAnalyze`. `ClipAnalyze` is a fine-grained perception tool that answers an agent-generated question q' over a specified time span τ. Given the pair (τ, q'), the tool dispatches the corresponding video segment $\mathcal{V}[\tau]$ to the Video-Audio MLLM together with an analysis prompt $prompt_{\text{clip}}$, which returns an answer a accompanied by a textual explanation e grounding the answer in the segment's visual and audio content. To improve throughput, `ClipAnalyze` is implemented in a batched manner: when the agent issues B independent (τ_b, q'_b) pairs, all B requests are dispatched in parallel rather than sequentially:

$$\texttt{ClipAnalyze}\big(\{(\tau_b, q'_b)\}_{b=1}^B\big) \longrightarrow \{(ans_b, reason_b)\}_{b=1}^B. \tag{9}$$

`MultiClipJudge`. GlobalSearch and ClipAnalyze cache visual evidence as text in $\mathcal{M}_C$. This textual representation, however, cannot support questions requiring *direct cross-segment visual comparison*, such as relative brightness, crowd density, or outfit consistency between characters.Such fine-grained perceptual comparisons cannot be reliably reconstructed from per-clip textual summaries. `MultiClipJudge` addresses this gap by issuing a single Video-Audio MLLM call that jointly conditions on all candidate segments and a comparison question q'', which returns a single answer a grounded in cross-segment comparison:

$$\texttt{MultiClipJudge}\big(\{\tau_k\}_{k=1}^K, q''\big) \longrightarrow a. \tag{10}$$

3.4 The Reasoning Process of an Active LVU Agent

Figure 3 illustrates the typical workflow of our method on questions involving named entities: the agent first registers the candidate names with `PersonRegister`, and then carries out plot reasoning to arrive at the final answer. In more general cases, the agent invokes `GlobalBrowse` to establish global context, dispatches `GlobalSearch` to localize content relevant to the user query, and verifies the critical evidence with `ClipAnalyze` and `MultiClipJudge` before committing a final answer.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate VideoSleuth on four established long-form video understanding benchmarks. Our primary evaluation benchmark, *LVBench* [18], contains 1,549 multiple-choice questions across 103 hour-long videos and stands as one of the most challenging benchmarks for extreme long-form video understanding; we use subtitles for all videos. *LongVideoBench* [19] features 6,678 questions from 3,763 videos with durations ranging from a few seconds to one hour; we focus on the long subset of the val split—188 videos and 564 questions in the $(900\text{s}, 3600\text{s}]$ range—relying on the official subtitles since most videos in this subset lack audio tracks. *Video-MME* [20] is segmented by video duration; we concentrate on the long split (300 videos and 900 questions, with durations ranging from 30 to 60 minutes under the *w/ sub* setting. Finally, *EgoSchema* [21] serves as a diagnostic benchmark for long-form understanding, on whose val split we evaluate 500 three-minute videos with 500 questions. Across all datasets, we use audio streams only when the language is English.

Implementation details. VideoSleuth instantiates five tool-level model roles with a Flash/Pro cost–accuracy split: `PersonRegister` and `GlobalSearch` use gemini-3-flash-preview for their Video-Audio MLLM calls, with `PersonRegister`'s Stage 1 mention-localization step delegated to DeepSeek v3.2 Chat [33] ; `ClipAnalyze` and `MultiClipJudge` use gemini-3.1-pro-preview; the grounding vlm used in step 3 is gemini-3-flash-preview; ASR uses Whisper Large-v3 [34] and OCR uses PaddleOCR PP-OCRv4 [35]. Local ASR and OCR preprocessing was performed on two NVIDIA RTX 4090 GPUs. The controller LLM Φ_{L} is GPT-5.4, capped at 10 ReAct iterations per question; segments dispatched to the Video-Audio MLLM are kept below 40 MB via adaptive bitrate/resolution. We set $L_{\text{c}} = 120$ s in `GlobalSearch` (reduced to 60 s on EgoSchema to match its

Table 1: Per-category accuracy on LVBench. Columns: Entity Recognition (ER), Event Understanding (EU), Key Information Retrieval (KIR), Temporal Grounding (TG), Reasoning (Rea), Summarization (Sum). Within the on-demand grounding group, best in each column is in **bold** and second-best is underlined. "-" indicates results not reported.

Methods	ER	EU	KIR	TG	Rea	Sum	Overall
Commercial VLMs							
GLM-4V-Plus [22]	46.2	47.8	54.1	42.7	46.5	37.9	48.7
GPT-4o [23]	48.9	49.5	48.1	40.9	50.3	50.0	48.9
OpenAI o3 [24]	57.6	56.4	62.9	46.8	50.8	67.2	57.1
Seed1.5-VL-Thinking [25]	65.4	63.4	68.0	53.6	63.7	46.6	64.6
Gemini-2.5-Pro [26]	64.5	67.5	72.8	65.9	66.5	58.6	67.4
Gemini-3.1-Pro	-	-	-	-	-	-	75.1
Open-Source VLMs							
InternVL2.5-78B [27]	43.8	42.0	42.1	36.8	51.0	37.9	43.6
VideoLLaMA3-7B [28]	45.8	42.4	47.8	35.9	45.8	36.2	45.3
Qwen2.5-VL-72B [29]	-	-	-	-	-	-	47.7
VideoChat-Flash [30]	51.1	46.0	49.0	38.9	48.5	34.5	48.2
AdaReTaKe [31]	53.0	50.7	62.2	45.5	54.7	37.9	53.3
Dense caption-retrieval-based methods							
VideoTree [7]	30.3	25.1	26.5	27.7	31.9	25.5	28.8
MR. Video [32]	59.8	57.4	71.4	58.8	57.7	50.0	60.8
DVD [3]	73.4	73.3	80.4	72.3	70.7	74.1	74.2
DVD w. subtitle [3]	75.5	77.1	79.0	72.7	68.7	84.5	76.0
HAVEN [5]	79.9	82.6	84.0	86.7	79.6	79.3	81.0
On-demand grounding methods							
VideoAgent [11]	28.0	30.3	28.0	29.3	28.0	36.4	29.3
VideoLucy [12]	54.3	59.8	75.6	51.7	55.9	49.1	58.8
Symphony [4]	70.0	69.4	77.2	70.1	69.4	72.5	71.8
VideoSeek [2]	-	-	-	-	-	-	76.7
VideoSleuth (ours)	**80.7**	**80.4**	**86.9**	**81.4**	**78.6**	**82.8**	**81.6**

3-minute clip length). Local ASR and OCR preprocessing can be performed on a single NVIDIA RTX 4090 GPU.

4.2 Main results

Table 2: Main results on four long-form video understanding benchmarks (multiple-choice accuracy). Within the on-demand grounding group, best in each column is in **bold** and second-best is underlined. "—" indicates results not reported by the original paper.

Method	LVBench	LongVideoBench (val)	Video-MME (long) w/o sub	Video-MME (long) w/ sub	EgoSchema
Dense caption-retrieval-based methods					
VideoTree [7]	28.8	—	—	—	67.0
MR. Video [32]	60.8	61.6	61.8	—	73.0
DVD [3]	76.0	68.6	67.3	76.0	76.6
HAVEN [5]	81.0	78.2	—	82.8	81.6
On-demand grounding methods					
VideoAgent [11]	29.3	—	—	—	63.2
Logic-in-Frames [36]	—	54.2	55.2	—	—
VideoLucy [12]	—	—	66.8	—	—
VideoSeek [2]	76.7	73.5	70.1	81.2	—
Symphony [4]	71.8	77.1	—	78.1	—
VideoSleuth (ours)	**81.6**	**78.5**	—	**83.9**	**79.2**

Table 1 presents per-category results on LVBench, where for HAVEN we report results at fps=1. VideoSleuth substantially outperforms all on-demand grounding baselines, surpassing the strongest prior on-demand agent VideoSeek by 4.9% overall and Symphony by 9.8% overall. Compared to dense caption-retrieval-based methods, which incur substantial offline preprocessing cost (Section 4.3), VideoSleuth still surpasses DVD by 7.4% and matches the strongest baseline HAVEN (81.6% vs. 81.0%) without any dense offline VLM preprocessing. The advantage is particularly pronounced on capability dimensions that demand fine-grained evidence localization—Key Information Retrieval (86.9%, $+2.9\%$ over HAVEN) and Event Understanding (80.4%).

Table 2 presents a comprehensive comparison across several popular long-video benchmarks, where for HAVEN we report results at fps=0.67. VideoSleuth consistently outperforms prior on-demand grounding methods on LVBench, LongVideoBench, Video-MME, and EgoSchema. Compared to dense caption-retrieval-based methods, VideoSleuth surpasses the strongest baseline (HAVEN) on LVBench, LongVideoBench, and Video-MME (w/ sub); importantly, this competitive accuracy is obtained *without any dense VLM-based offline preprocessing*, in contrast to dense caption-retrieval pipelines whose preprocessing cost scales with video length and dominates total token consumption (we quantify this gap in Section 4.3).

Table 3: Ablation studies on VideoSleuth, evaluated on LVBench. **(a)** Backbone generalizability (200 questions). **(b)** Effectiveness of `PersonRegister` (84 name-anchored questions). **(c)** Sensitivity to Δ_{seg} in `GlobalSearch`. **(d)** Component contribution; `PersonRegister` and `GlobalSearch` are enabled throughout. **(e)** Per-segment token consumption (avg. over 5 videos at 2 fps). **(f)** End-to-end input tokens (20 questions, avg. 32.5 min). (c)/(d) use a 154-question subset. **(g)** Effect of global memory. **(h)** Effect of narrative-structured chunk captions. (g)/(h) each use a 150-question LVBench subset.

Configuration	Flash role	Pro role	Acc.
Default (Gemini)	gemini-3-flash	gemini-3.1-pro	**81.5**
Qwen replacement	qwen3.6-flash	qwen3.6-plus	79.0

(a) Backbone generalizability.

Configuration	Acc.
Full VideoSleuth	**85.7**
W/o `PersonRegister`	81.0

(b) Effect of `PersonRegister`.

Δ_{seg} (s)	Acc.
60	80.5
120	**81.8**

(c) Segment length sensitivity.

`GlobalBrowse`	`ClipAnalyze`	`MultiClipJudge`	Acc.
✓	✓	✓	**81.8**
✓	✓		78.6
✓			69.5

(d) Component contribution. ✓: enabled.

Protocol	Avg. input tokens
Frame-sampling (60 frames)	63,876
Video-input (30 s @ 2 fps)	**8,755**
Reduction	$\mathbf{7.3}\times$

(e) Per-segment token consumption.

Method	Cost type	Avg. tokens
DVD-style	Preproc. only (est.)	$\geq$ 4,151,940
VideoSleuth	End-to-end (meas.)	**793,626**
Reduction		$\geq \mathbf{5.2}\times$

(f) End-to-end token consumption.

Configuration	Acc.
Full VideoSleuth	**81.3**
W/o global memory	78.7

(g) Global memory.

Caption configuration	Acc.
Narrative-structured	**81.3**
Without captions	76.7
Generic free-form	78.7

(h) Narrative captions.

4.3 Ablation studies

Agent Component Ablations. We conduct four targeted ablations on LVBench, summarized in Table 3. **(a) Backbone Generalizability**: replacing both Gemini backbones with Qwen counterparts on a 200-question subset (selected for safety-filter eligibility on both providers) drops accuracy from 81.5% to 79.0%. **(b) Effectiveness of `PersonRegister`**: on 84 name-anchored questions, removing `PersonRegister` drops accuracy from 85.7% to 81.0%. **(c) Segment Length Δ_{seg}**: on a 154-question subset, $\Delta_{\text{seg}} = 120$ s achieves 81.8% versus 80.5% at 60 s. **(d) Component Contribution**: on the same subset, with `PersonRegister` and `GlobalSearch` enabled throughout, removing

`MultiClipJudge` drops accuracy from 81.8% to 78.6%, and further removing `ClipAnalyze` drops to 69.5%.

Token Efficiency Ablations. A central claim of our framework is that dispatching minute-scale video segments to a video-native MLLM is markedly more token-efficient than sampling frames densely and feeding them as a sequence of images. To quantify this, we compare two protocols on a 30-second segment from each of 5 randomly sampled LVBench videos, both at the same 2 fps temporal resolution: **(i)** frame-sampling (DVD-style)—60 uniformly sampled frames (30 s $\times$ 2 fps) dispatched as image inputs to gemini-3-flash-preview; **(ii)** video-input (ours)—the segment dispatched directly as a single video input to the same model at 2 fps. Table 3 (e) reports a $7.3\times$ token reduction. To translate this advantage to end-to-end question answering, we measure VideoSleuth's total input tokens across the entire ReAct trajectory on 20 LVBench questions (avg. video duration 32.5 min), and estimate DVD's *offline preprocessing alone* by extrapolating the per-segment cost in Table 3 (e) to a 32.5-min video ($63{,}876 \times 65 = 4{,}151{,}940$ tokens). VideoSleuth's complete inference budget is $5.2\times$ smaller than what DVD spends on preprocessing in the single-question setting (Table 3 (f))—a conservative lower bound, since DVD must additionally pay an inference-time cost to query its captioned index.

5 Conclusion

We presented **VideoSleuth**, a narrative-centric agentic framework for long-form video understanding that organizes its working memory around the four narrative elements—time, place, person, and event—and pairs this memory with a video-native MLLM as the perceptual backbone, jointly exploiting vision and audio rather than treating perception as frame-based image captioning. On four standard LVU benchmarks, VideoSleuth outperforms every on-demand grounding baseline by 1.4–16.0 points and matches the strongest dense-preprocessing baseline despite incurring no dense VLM-based offline preprocessing, while consuming at least $5.2\times$ fewer tokens end-to-end than DVD-style pipelines spend on preprocessing in the single-question setting.

Limitations. While VideoSleuth advances long-form video understanding, its strongest results currently rely on frontier commercial video-audio MLLMs, and its profiling mechanism assumes that target names appear in dialogue or on-screen text. In future work, we will explore open-source Video-Audio MLLMs of comparable capability and extend the framework to general object tracking and unnamed-entity reasoning beyond the current name-driven design. Moreover, The relative contributions of the agent design, model choices, and inference budget warrant further investigation.

Acknowledgments and Disclosure of Funding

This work was supported by Shanghai Science and Technology Committee under Grant No. 24511103900 and National Science Foundation of China under Grant No. 62676101. The authors declare no competing interests. We thank the anonymous reviewers for their insightful feedback on this work.

References

[1] Yufei Yin, Qianke Meng, Minghao Chen, Jiajun Ding, Zhenwei Shao, and Zhou Yu. Videoarm: Agentic reasoning over hierarchical memory for long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24042–24051, 2026.

[2] Jingyang Lin, Jialian Wu, Jiang Liu, Ximeng Sun, Ze Wang, Xiaodong Yu, Jiebo Luo, Zicheng Liu, and Emad Barsoum. Videoseek: Long-horizon video agent with tool-guided seeking. *arXiv preprint arXiv:2603.20185*, 2026.

[3] Xiaoyi Zhang, Zhaoyang Jia, Zongyu Guo, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Deep video discovery: Agentic search with tool use for long-form video understanding. *Advances in Neural Information Processing Systems*, 38:89863–89895, 2026.

[4] Haiyang Yan, Hongyun Zhou, Peng Xu, Xiaoxue Feng, and Mengyi Liu. Symphony: A cognitively-inspired multi-agent system for long-video understanding. *arXiv preprint arXiv:2603.17307*, 2026.

[5] Xinlei Yin, Xiulian Peng, Xiao Li, Zhiwei Xiong, and Yan Lu. Hierarchical long video understanding with audiovisual entity cohesion and agentic search. *arXiv preprint arXiv:2601.13719*, 2026.

[6] Xubin Ren, Lingrui Xu, Long Xia, Shuaiqiang Wang, Dawei Yin, and Chao Huang. Videorag: Retrieval-augmented generation with extreme long-context videos. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 2390–2401, 2026.

[7] Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3272–3282. IEEE, 2025.

[8] Xiaoqian Shen, Wenxuan Zhang, Jun Chen, and Mohamed Elhoseiny. Vgent: Graph-based retrieval-reasoning-augmented generation for long video understanding. *Advances in Neural Information Processing Systems*, 38:100805–100830, 2026.

[9] Zhucun Xue, Jiangning Zhang, Xurong Xie, Yuxuan Cai, Yong Liu, Xiangtai Li, and Dacheng Tao. Adavideorag: Omni-contextual adaptive retrieval-augmented efficient long video understanding. *arXiv preprint arXiv:2506.13589*, 2025.

[10] Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. Drvideo: Document retrieval based long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18936–18946. IEEE, 2025.

[11] Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, pages 75–92. Springer, 2024.

[12] Jialong Zuo, Yongtai Deng, Lingdong Kong, Jingkang Yang, Rui Jin, Yiwei Zhang, Nong Sang, Liang Pan, Ziwei Liu, and Changxin Gao. Videolucy: Deep memory backtracking for long video understanding. *Advances in Neural Information Processing Systems*, 38:25660–25691, 2026.

[13] Boyu Chen, Zhengrong Yue, Siran Chen, Zikang Wang, Yang Liu, Peng Li, and Yali Wang. Lvagent: Long video understanding by multi-round dynamical collaboration of mllm agents. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20237–20246. IEEE, 2025.

[14] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 conference on empirical methods in natural language processing: system demonstrations*, pages 543–553, 2023.

[15] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[16] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.

[17] Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025.

[18] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22958–22967. IEEE, 2025.

[19] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

[20] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24108–24118. IEEE, 2025.

[21] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

[22] Team Glm, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

[23] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[24] OpenAI. Introducing OpenAI o3 and o4-mini, 2025. `https://openai.com/index/introducing-o3-and-o4-mini/`.

[25] Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025.

[26] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[27] Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025.

[28] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.

[29] Shuai Bai, Ke qin Chen, Xue-Jing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *ArXiv*, abs/2502.13923, 2025. URL `https://api.semanticscholar.org/CorpusID:276449796`.

[30] Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for long-context video modeling. In *International Conference on Learning Representations*, volume 2026, pages 109089–109117, 2026.

[31] Xiao Wang, Qingyi Si, Shiyu Zhu, Jianlong Wu, Li Cao, and Liqiang Nie. Adaretake: Adaptive redundancy reduction to perceive longer for video-language understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5417–5432, 2025.

[32] Ziqi Pang and Yu-Xiong Wang. Mr. video:" mapreduce" is the principle for long video understanding. *arXiv preprint arXiv:2504.16082*, 2025.

[33] Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

[34] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[35] Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, et al. PaddleOCR 3.0 technical report. *arXiv preprint arXiv:2507.05595*, 2025.

[36] Weiyu Guo, Ziyang Chen, Shaoguang Wang, Jianxiang He, Yijie Xu, Jinhui Ye, Ying Sun, and Hui Xiong. Logic-in-frames: Dynamic keyframe search via visual semantic-logical verification for long video understanding. *Advances in Neural Information Processing Systems*, 38: 124389–124422, 2026.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state our main contributions, which are validated empirically in the main body of the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations at the end of Section 5, covering the dependence on the target name appearing in ASR or OCR for `PersonRegister`, and the reliance on frontier commercial Video-Audio MLLMs as the perceptual backbone.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it cannot handle technical jargon.

- The authors should discuss the computational efficiency of the model and how it scales with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper is empirical in nature: it proposes a framework, an instantiated agent, and reports benchmark results. It introduces no theorems, lemmas, or formal proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4.1 specifies all model snapshots, hyperparameters, and benchmark protocols, and the Appendix reproduces every prompt used by the system.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes theoretical results, see the note above.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and instructions for the LVBench experiments are available at is publicly available at `https://github.com/SoftEgLi/VideoSleuth`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: VideoSleuth is training-free; all inference-time hyperparameters and benchmark splits are specified in Section 4.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Each evaluation already exhausts the full benchmark test set, and the gaps to the strongest baselines substantially exceed the run-to-run jitter we observed; multi-run reporting would multiply API cost without changing the conclusions.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 90% CI, if the hypothesis of Normality of errors is not verified.
- For null hypotheses such as data equality, the authors should preferably report two-sided tests than one-sided tests, but one should be careful of the interpretation of the consequences.
- For error bars reported in figures, the authors should support that the figure can be at most 1 sigma, and we suggest that it should clearly stated.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Local ASR and OCR preprocessing was performed on two NVIDIA RTX 4090 GPUs, while the controller and video-audio models were accessed through commercial APIs. Section 4.1 describes the model configurations, and Section 4.3 reports token consumption for selected experiments. Execution times and a complete accounting of the total compute used are not reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: We have read the NeurIPS Code of Ethics and confirm conformance: we use only publicly released benchmarks under their original licenses and do not collect data from human subjects.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss positive societal impacts (accessibility, archival search, journalistic verification) and potential negative impacts (named-person misuse risk, biases inherited from the underlying foundation models) in the Appendix, together with the safety constraints we inherit from the underlying commercial Video-Audio MLLMs and recommendations for additional access controls in downstream deployments.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is fine to mention that there is improvement in the quality of generative models or generative AI, and a generic discussion of the reasons people might wish to use generative AI, including its impacts. The authors should discuss possible mitigations of those negative societal impacts at issue (e.g., ways to fail-safe, ways to monitor misuse, ways to reduce environmental impact, ways to mitigate societal biases, etc).
- If there are negative societal impacts, the authors should discuss possible mitigations strategies (e.g., providing defenses in addition to attacks, pointing out biases and downstream mitigations, etc.).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We do not release any pre-trained model, scraped dataset, or generative artifact; the contribution is an agentic framework that orchestrates existing publicly accessible APIs and open-source models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All third-party assets (benchmarks, perception backbones, ASR, OCR) are cited in the paper and used under their publicly distributed licenses and original terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be respected.
    - If assets are released, the license, copyright information, and terms of use in the package should be included. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this asset is released, the license, copyright information, and terms of use in the package should be included.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code for the LVBench experiments is publicly available at `https://github.com/SoftEgLi/VideoSleuth`. The repository includes a README with installation, data preparation, API configuration, and evaluation instructions, as well as a license file.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: The paper involves no crowdsourcing and no research with human subjects; all evaluation uses pre-existing public benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects research is conducted; an IRB review is therefore not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs and Video-Audio MLLMs are core to our methodology. Each role, model snapshot, and prompt is documented in Section 4.1 and Appendix D.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2026/LLM`) for what should or should not be described.

A Qualitative case study

Q: **Who talks in front of a white building with green plants outside it, wearing an exotic hat?** A. Abiola Oladapo **B. Mele Kyari** ✓ C. Jimoh Ibrahim D. Peter Odili

Answer: B ✓

Think: The four MCQ options are all proper-noun person names, but I have no idea what any of them looks like. Before I can ask "who is the speaker?" of any clip, I need a face profile for each candidate—otherwise the visual evidence is uninterpretable.

Step 1 `PersonRegister`

Inputs:

names: Abiola Oladapo · Mele Kyari · Jimoh Ibrahim · Peter Odili

Result: All 4 candidates grounded to a reference frame, with appearance + setting captured.



Think: Faces are now in memory, but I still don't know what type of video this is—a movie? a news show? a sports broadcast? Without that context, I can't decide whether to look at speeches, interviews, panel discussions, etc. Get the gist first.

Step 2 `GlobalBrowse`

Inputs:

time_range: [0–3675 s] (full video)

Result: A Channels Television Nigerian news broadcast covering Rivers State politics, Ondo governorship, NNPC oil-output and electricity-tariff segments—multiple men in traditional caps appear, so visual disambiguation is required.

Think: It's a Nigerian news broadcast with multiple male interviewees in caps—name alone won't disambiguate. The question gives me three concrete visual cues: white building, green plants outside, exotic hat. *I should sweep the whole 61-min video for any segment matching these cues to find candidate windows.*

Step 3 `GlobalSearch`

Inputs:

time_range: [0–3675 s] (chunked)

Question: *"Who talks in front of a white building with green plants outside it, wearing an exotic hat? (A) Abiola Oladapo (B) Mele Kyari (C) Jimoh Ibrahim (D) Peter Odili"*

Result — chunk captions:

[0:00–2:00] *"The video clip is an opening montage for a news broadcast on Channels Television, featuring various news segments and presenters…"*
[1:45–3:45] …
[3:30–5:30] …
[5:15–7:15] …
(+31 more chunks)

Candidate intervals selected: [08:30–09:19] · [46:39–47:53]

Think: Two windows look promising—Peter Odili at a podium (08:30) and Mele Kyari outdoors (46:39). Chunk captions are coarse text—I shouldn't trust them blindly. Pull the actual frames for both windows and check the three cues against the speaker's identity using the reference faces.

Step 4 `ClipAnalyze`

Inputs (question per interval):

[510–559 s]: *"Identify the person speaking in this clip using on-screen text, ASR, or clear visual identity. Also state whether he is in front of a white building with green plants outside and whether he is wearing an exotic/traditional hat."*
[2799–2873 s]: *" (same)*

Result:



[08:30–09:19] → Peter Odili (matched via PersonRegister reference), but he is at a green podium, not a white building. Hat ✓ but setting ×.
[46:39–47:53] → Mele Kyari, with on-screen text *"MELE KYARI"* at 46:54 and visible white building + green plants + patterned cap.

Think: ClipAnalyze rules out window 1 (Odili: hat ✓ but green podium, no white building) and supports window 2 (Kyari: all three cues match + on-screen "MELE KYARI" caption). Decision is leaning B, but two clips returned different identities—call MultiClipJudge for one final side-by-side comparison before locking the answer.

Step 5 `MultiClipJudge`

Inputs:

time_range: [525–555 s]
time_range: [2810–2840 s]
Question: *"Which clip better matches this description: a person speaking in front of a white building with green plants outside while wearing an exotic or traditional hat? Identify the speaker in the matching clip using on-screen text or clear visual identity, and note why the other clip does not match as closely."*

Result:



✓ *"Clip 2 is the better match—Mele Kyari speaking outdoors in front of a white building with green plants while wearing a traditional patterned cap."*

Think: All evidence converges. The [46:39–47:53] speaker is Mele Kyari, and his setting matches every cue in the question. No conflicting evidence, no remaining hypothesis to test. Lock the answer.

Step 6 **Synthesis**

Three visual cues from the question all match [46:39–47:53] only → (B) Mele Kyari .

Figure 4: An example workflow of VideoSleuth on a name-anchored, scene-anchored question over a 1-hour news broadcast. Each step is preceded by the agent's internal *Think* (grey italic) and includes the explicit query sent to the dispatched tool.

B Failure modes of `PersonRegister`

Inspecting the failure cases of the full system on the 84 name-anchored questions evaluated in Section 4.3, we identify three recurring failure modes:

(i) *Visually similar persons.* When multiple individuals share salient appearance attributes (e.g., players of the same team in football matches, or persons with similar facial features), the Video-Audio MLLM produces near-identical visual descriptions; the open-vocabulary detector cannot reliably distinguish them, and the cropped reference image img_i may not match the intended person.

(ii) *Wardrobe changes across the video.* Even with a correct reference image img_i from one appearance, persons that change clothing or accessories elsewhere in the video can become unrecognizable, causing downstream tool calls to miss subsequent appearances.

(iii) *Names absent from ASR/OCR.* `PersonRegister`'s mention localization stage (Section 3.3) relies on $\mathcal{A} \cup \mathcal{O}$ containing surface forms of the target name. Names that are never spoken on screen and never appear as on-screen text cannot be localized, so no profile is constructed.

These failure modes are intrinsic to the visual disambiguation problem rather than to our tool design; addressing them would require advances in fine-grained re-identification and richer narrative anchors, which we leave to future work.

C Additional Analyses

C.1 Contribution of Audio and Text Inputs

Table 4: Effect of raw audio on PersonRegister. We manually assess whether the reference image identifies the correct person on 76 LVBench questions with person-name answer options. ASR/OCR-based mention localization remains unchanged. Neither grounding setting receives a transcript.

Grounding input	Identification accuracy (%)	Change (pp)
Video + raw audio	72.4	—
Video only	63.2	−9.2

Table 5: Input analysis for GlobalSearch, ClipAnalyze, and MultiClipJudge on a stratified random sample of 150 LVBench questions. Only the inputs to these three tools are varied; PersonRegister remains unchanged. The metric is final question-answering accuracy.

Input setting	QA accuracy (%)	Change (pp)
Full configuration	81.3	—
Video only	78.7	−2.6
Video + ASR	80.0	−1.3
Video + raw audio	78.7	−2.6
Video + subtitles	77.3	−4.0
Video + ASR + OCR	80.7	−0.6

Raw audio improves person identification in this diagnostic. For the other three tools, ASR and OCR provide most of the observed improvement over video-only input.

C.2 Token Amortization Across Questions

We compare DVD and VideoSleuth on the same 5,219-second video at 360p. Table 6 separates one-time preprocessing from per-question answering costs. All totals include both input and output tokens.

Table 6: Token consumption on the same video. DVD pays the preprocessing cost once and reuses its index across questions. VideoSleuth repeats question-conditioned processing under the evaluated implementation.

Stage	Input tokens	Output tokens	Total tokens
DVD preprocessing	4,094,762	176,889	4,271,651
DVD answering per question	70,530	1,060	71,590
VideoSleuth per question	638,548	25,084	663,632

Using these measured costs, we estimate the total token consumption for N questions about the same video:

$$C_{\mathrm{DVD}}(N) = 4{,}271{,}651 + 71{,}590N, \tag{11}$$

$$C_{\mathrm{VideoSleuth}}(N) = 663{,}632N. \tag{12}$$

The two cost estimates intersect at approximately 7.2 questions. For this example, VideoSleuth uses fewer tokens for up to seven questions, whereas DVD uses fewer for eight or more questions. The crossover depends on the video and model configurations; it is not a universal threshold.

C.3 Single-Provider Configuration

We evaluate a Gemini-only configuration on same 150-question subset of LVBench. We replace the controller and name-localization model, while keeping the perception models unchanged.

Table 7: Model assignments and QA accuracy for the default and single-provider configurations.

Component	Default configuration	Gemini-only configuration
Controller	GPT-5.4	Gemini-3.1-Pro
Name localization	DeepSeek-v3.2 Chat	Gemini-3.1-Flash-Lite-preview
Perception models	Gemini models	Unchanged
QA accuracy (%)	81.3	78.7

VideoSleuth can operate with a single MLLM provider, with a 2.6-percentage-point accuracy decrease in this evaluation.

D Prompts

This appendix collects the system, tool-description, and per-tool VLM prompts used by VideoSleuth at inference time. Prompts are organized by the LLM/VLM that consumes them. They are shown verbatim except that the implementation's internal tool names have been renamed to the paper convention (see "Note on tool naming" below). Run-time placeholders of the form `{name}`, `{question}`, `{n}`, etc. are shown in template form; concrete values are substituted per question / per chunk / per clip.

We first present, in the order in which they appear in Section 3, the prompts directly consumed by the five tools described in the main paper (`PersonRegister`, `GlobalBrowse`, `GlobalSearch`, `ClipAnalyze`, `MultiClipJudge`). Auxiliary prompts that govern the agent's top-level orchestration are collected at the end of this appendix for completeness.

D.1 PersonRegister prompts

`PersonRegister` is implemented as the three-step pipeline described in Section 3. We list the prompt used in each step below.

PERSON_REGISTER_TOOL_DESCRIPTION

```
Identify and profile named persons in the active video. Input: a list of names. For each name, searches
ASR/OCR for mentions, verifies via gemini-3-flash video analysis that the person is actually visible (
not just mentioned), then runs gemini-3-flash bbox grounding on frames near the subject timestamp to
crop a reference image. Registers {name, appearance, reference_image_b64} into VideoMemory. Returns per-
name status: 'Succeed ...' / 'Not shown' / 'No mentions' / 'Already profiled'. Does not return the
reference image bytes in the response.
```

PERSON_REGISTER_INPUT_DESCRIPTION

names:

```
List of person names to identify and profile.
```

Step 1: Name Localization

```
You are a transcript-search assistant. Given a video's chronological ASR + OCR transcript (each line
prefixed with a unique numeric ID and its source/timestamp) and a list of TARGET PERSON NAMES, identify
every line where a target person is mentioned by name.

MATCHING RULES:
1. Names may have ASR-driven spelling variants -- the same person may be transcribed differently in
different parts of the video. Treat phonetically similar variants as the same name (examples: Whitaker /
 Whitacombe / Whittaker; Conaty / Connerty; Romesh / Ramesh; Catherine / Katherine; Ann / Anne).
2. Only flag a line if the target's NAME or a phonetic spelling variant of it literally appears in that
line's text. Pronoun references ('he', 'she', 'they') without a name in the same line do NOT count.
3. DO NOT match unrelated words that merely sound similar (e.g. 'trumps' the cards vs 'Trump' the person;
 'white' the color vs 'Whitaker' the surname; 'great' the adjective vs 'Greg' the name). Use the
surrounding context to filter such cases out.
4. If two different target names share the same first name (e.g. 'Josh Smith' and 'Josh Garcia'), only
flag a line for the target whose FULL name (or a clearly matching variant) is supported by that line's
content.

TARGET NAMES:
{name_list}

TRANSCRIPT (chronological; ID at line start):
{rendered_lines}

OUTPUT:
Return BARE JSON only. NO markdown code fences (no ```json), NO surrounding prose, NO explanations. Use
this exact schema:
{
  "<target name verbatim from list>": [<int line ID>, <int line ID>, ...],
  ...
}
Each target name in the input MUST appear as a key in the output (use an empty array if no mentions
found). IDs MUST be unique and ascending within each list. Output ONLY the integer IDs -- no per-id
reason, no source tag, no extra fields.
```

Step 2: Video-Audio Grounding via MLLM

```
You are watching a video clip.
VIDEO INFO: duration = {clip_duration_sec:.3f} seconds. Timestamps must be in mm:ss.mmm format (
millisecond precision; e.g. 01:42.500) and lie within [00:00.000, {end_mm:02d}:{end_ss:02d}.{end_ms:03d
}].

TARGET PERSON NAME: "{target_name}"

Decide whether the person named above is VISUALLY shown in this clip. The name being merely mentioned in
 speech or on-screen text does NOT count -- the person themselves must actually appear in the video.

If shown, provide:
  1. Visual appearance: clothing, hair, accessories, distinctive features. Pure visual description ONLY
-- do NOT include any name, role, profession, or any wording like '<name> has ...' or 'the host wearing
...'. The text must read as if it could describe ANYONE who looks that way. Write it as ONE single
continuous sentence with NO internal punctuation -- no commas, no semicolons, no dashes -- only a single
 trailing period is allowed. Connect attributes with words like 'with', 'wearing', 'and'. Example (
correct): 'A middle-aged man with grey hair wearing a dark suit and a red striped tie.' Example (wrong --
 comma chopped): 'A middle-aged man, grey hair, dark suit, red tie.'
  2. Identity / role: this person's role, profession, title, or relationship in the scene -- NOT their
name (the name is already known and stored separately). Keep it to a short phrase. Examples: 'host of
the show', 'contestant', 'judge sitting at the panel', 'father of the bride'. If no role is determinable,
 leave this field empty.
  3. ONE representative timestamp where this person appears as the principal SUBJECT of the shot -- i.e.
 a close-up or focused medium shot where they fill most of the frame and are clearly the focus. Do NOT
```

```
pick a wide group shot or a moment where they are background / off to the side. Pick the clearest such
moment.

Output strictly as JSON (no extra prose):
If NOT shown:
  {"shown": false}
If shown:
  {
    "shown": true,
    "appearance": "<pure visual description, no names, no roles>",
    "identity": "<role / profession / title only -- no name; empty string if unknown>",
    "show_time": "mm:ss.mmm"
  }
```

Step 3: Reference Image Extraction

```
You are a visual grounding model. Look at the image and find the person that best matches this
description:

"{appearance}"

Return a JSON array. Each entry corresponds to one person you can see in the image that could plausibly
match, ordered best-match first. For each entry:
{
  "label": "<short description of who this person looks like>",
  "confidence": <float 0.0-1.0 -- how well this person matches the description>,
  "box_2d": [y_min, x_min, y_max, x_max]   // normalized to 0-1000 scale, integers
}

Rules:
- If NO person in the image plausibly matches, return []
- Do NOT invent people that are not visible.
- Output ONLY the JSON array. No prose, no markdown code fence.
```

D.2 GlobalBrowse prompts

GLOBAL_BROWSE_TOOL_DESCRIPTION

```
MANDATORY first tool (except in Branch A) -- holistic look at the WHOLE active video. Uniformly samples
~30 frames across the full video, stitches them into one grid image with MM:SS timestamps burned top-
left on each tile, attaches the entire ASR transcript from VideoMemory, and asks a Flash VLM to produce
a thematic summary (3-5 sentences, no timestamps) + main_topics list. The summary is automatically
persisted to VideoMemory so that any subsequent GlobalBrowse / ClipAnalyze call on the same video gets
it prepended as a CONTEXT block -- this anchors per-chunk VLM prompts to the overall narrative. Takes NO
 inputs. Returns {summary, main_topics, n_frames, characters_attached, ...}. Does NOT return an answer --
 even pure synopsis questions must be verified with GlobalBrowse + ClipAnalyze before emitting.
```

VLM prompt

```
You are watching a HOLISTIC OVERVIEW of a single video.

The image above is a {rows}×{cols} grid of {n_frames} frames sampled uniformly across the full video (
duration ≈ {dur_mm:02d}:{dur_ss:02d}). Each tile has its source-video timestamp burned in the top-left
corner as MM:SS. Read the grid LEFT-TO-RIGHT, TOP-TO-BOTTOM -- tiles are in chronological order of the
video.

If an ASR transcript block appears above the grid, treat it as the AUTHORITATIVE source for anything
spoken / narrated in the video. Do NOT invent subtitle content. Some videos have no audio -- the ASR
block will then be missing or empty; that is normal, rely on the grid alone.

YOUR TASK:
  1. Synthesise the whole grid + the whole ASR transcript into a
     thematic understanding of the video. Opening framing, narrator
     voiceover, recurring on-screen text, and the cumulative arc of
     imagery all count -- do not over-weight any single tile.
  2. Write a 3-5 sentence holistic summary that a viewer could read
     without seeing the video and get its gist. **IMPORTANT: the
     summary must contain NO timestamps, NO `MM:SS` references, and
     NO phrases like 'at 02:15 ...' / 'in the opening minute' -- it
     should read as a purely thematic description of the video's
     content, usable as context for any segment.**
  3. List 3-6 main topics / entities the video revolves around.

Output strictly JSON (no prose outside the object):
```

```
{
  "summary":     "<3-5 sentences, holistic, NO timestamps, no time references>",
  "main_topics": ["topic1", "topic2", ...]
}
```

D.3 GlobalSearch prompts

GLOBAL_SEARCH_TOOL_DESCRIPTION

```
Pre-scan the active video for intervals relevant to a given question. Splits the video into 2-minute
chunks (15s overlap), queries a VLM on each chunk in parallel with the question, and returns BOTH (a) a
chronological per-chunk caption timeline AND (b) a merged list of source-time intervals with per-
interval reasons. Call AFTER GlobalObservation (its whole-video summary is automatically prepended to
every chunk's VLM prompt as a CONTEXT block, so chunks know the overall narrative). Each chunk's prompt
also receives the AUTHORITATIVE subtitle snippet for that time range (from VideoMemory's SRT) so the VLM
 doesn't invent dialogue. The VLM runs a two-pass procedure per chunk: first list what it actually sees
/ hears (without peeking at candidate options), then pick intervals. Each interval's reason is packed as
 [VISUAL] ... [ASR] ... [OCR] ... [WHY] ... so you can tell what the VLM observed vs what it interpreted
 -- trust the [VISUAL]/[ASR]/[OCR] parts, treat [WHY] as the VLM's guess. The new chunks field is the
per-chunk caption timeline (≤ 30 words each): use it to scan the whole video chronologically, spot
opening montages / location shifts, and check whether each chunk satisfies your question's location /
time / proper-noun anchors before trusting the intervals it produced. An empty intervals list is a legal
 answer; the chunk's caption is still emitted. If PersonRegister has populated VideoMemory with
character profiles, GlobalBrowse automatically prepends each character's reference face crop (with name
tag) to every VLM request. Returns {chunks:[{start_sec,end_sec,caption}], intervals:[{start_sec,end_sec,
reason}], characters_attached:[names...], ...stats}.
```

GLOBAL_SEARCH_INPUT_DESCRIPTION

question:

```
The full user question (include any MCQ options verbatim). The tool splits the active video into 2-min
overlapping chunks, sends each chunk + this question to a VLM in parallel, and returns the merged list
of time intervals relevant to answering it, with per-interval reasons.
```

VLM prompt

```
You are watching a video clip from a longer video.
VIDEO INFO: this clip covers SOURCE time {src_label} → {src_end_label} in the full video (clip duration
 = {clip_duration_sec:.3f}s). ALL timestamps in your output (`intervals.start_time` / `end_time` AND
caption phase prefixes) MUST be SOURCE time HH:MM:SS.mmm within [{src_label}, {src_end_label}]. The clip
 itself plays from 00:00:00.000, but every time you emit MUST be the SOURCE time (i.e. clip-local + {
src_label}).

USER QUESTION:
  {question}

GROUNDING RULES (apply to every field):
- Every claim must be grounded in something you actually SAW in the video frames OR that literally
appears in the AUTHORITATIVE ASR subtitle block above. Fabricating subtitles / on-screen text / speech
is a CRITICAL error.
- Never echo MCQ-option text unless that exact text appears verbatim in the ASR block or on-screen.
- LOAD-BEARING TOKENS rule: any proper noun (name/place/team/title), specific number (score/jersey/date/
count), or quoted line of dialogue MUST appear word-for-word verbatim in your fields. Use double-quotes
around direct ASR quotes. Do NOT dump raw OCR fragments -- summarise.
- Returning `{"intervals": []}` when nothing in the chunk bears on the question is correct and expected.
 Caption is still mandatory.

OUTPUT (one JSON object, no markdown fences, no prose outside it):
{
  "caption": "<≤100 words; chronological mini-summary of THIS chunk: main location/setting, who is on
screen, primary activity. If the chunk contains ≥ 2 distinct scenes/phases (different settings,
different speakers, clear cuts, topical breaks, etc.), prefix each phase with a SOURCE-time range `[hh:
mm:ss-hh:mm:ss]` covering that phase, e.g. `[00:10:30-00:11:05] <phase 1 description>;
[00:11:05-00:12:00] <phase 2 description>`. Skip the prefixes when the chunk is one continuous scene. If
 this chunk is an opening montage / recap teaser (rapid cuts of later scenes), say so verbatim ('opening
 montage: ...'). If the question contains location/proper-noun/label/number anchors, EXPLICITLY note 'in
 Hannam: NO/YES/ambiguous -- <evidence>' for each anchor with load-bearing tokens verbatim. ALWAYS
present, even when intervals is [].>",
  "intervals": [
    {
      "start_time":      "hh:mm:ss.mmm",   // MANDATORY -- SOURCE time
      "end_time":        "hh:mm:ss.mmm",   // MANDATORY -- SOURCE time. NEVER omit, NEVER null. For
instantaneous events use start_time + 00:00:01.000. Missing end_time → interval is DROPPED downstream.
```

```
      "observed_visual": "<MANDATORY 1-2 sentences describing exactly what is on screen (objects, people,
 actions, framing). Apply load-bearing-tokens rule.>",
      "observed_text":   "<≤30 words summarising on-screen text (titles/captions/scoreboards/labels/
signage) relevant to the question, with verbatim names/numbers. \"\" if no relevant on-screen text.>",
      "observed_asr":    "<≤30 words summarising what is spoken (from ASR block above) relevant to the
question, with verbatim direct quotes. \"\" if no relevant ASR.>",
      "reason":          "<≤25 words; MUST start with `positive:` (supports a candidate answer) or `
negative:` (grounded but rules out a hypothesis -- e.g. 'positive:' / 'negative: scoreboard 26-27 past
14-15 anchor'). Built strictly from observed_* above; do NOT introduce new claims.>"
    }
  ]
}

INTERVAL RULES:
- An interval MUST span the full continuous duration the relevant scene/event/object is on screen or
being discussed (e.g. 02:14-02:45 even when several subtitle lines play during it), NOT just one
subtitle line. Treat 1-2s ranges as suspicious unless the event is genuinely instantaneous (then use the
 +1s fallback in end_time).
- Emit intervals with positive evidence (supports an answer) AND with negative evidence (grounded but
rules out an option, e.g. scoreboard past the anchor, named player absent). Negative evidence helps the
downstream agent narrow the search -- do NOT drop it. Drop only intervals where you cannot ground any
concrete observation.
- Mentally scan the clip in chronological order before answering, but do NOT emit a separate
observations list -- keep your evidence inside the intervals' observed_* fields.

WORKED EXAMPLE -- a successful GlobalSearch reply on a 2-min chunk that covers SOURCE time 00:10:30.000
→ 00:12:30.000, where the question was about the order of sports in an opening promotional montage.
Note every time below is SOURCE time within the chunk range. Copy this shape EXACTLY: ONE bare JSON
object, no markdown fences, no `<answer>` tags, no prose before or after the braces.
{
  "caption": "[00:10:30-00:11:15] badminton match between Lee Chong Wei and Lin Dan, intense rallies and
 crowd reactions; [00:11:15-00:12:30] Olympic promotional montage cycling through several sports.",
  "intervals": [
    {
      "start_time":      "00:11:21.000",
      "end_time":        "00:11:22.000",
      "observed_visual": "A high jump athlete is seen clearing the bar in a very brief shot.",
      "observed_text":   "",
      "observed_asr":    "",
      "reason":          "positive: this is the second sport shown in the opening promotional montage,
following the initial badminton action."
    },
    {
      "start_time":      "00:11:36.000",
      "end_time":        "00:11:37.000",
      "observed_visual": "A gymnast is shown performing on the rings for a brief moment.",
      "observed_text":   "",
      "observed_asr":    "",
      "reason":          "negative: this is the third sport shown in the promotional montage, not the
second."
    }
  ]
}
Key things to copy from this real example:
- ONE bare JSON object -- NO ```json fences, NO `<answer>` tags, NO text before or after the braces.
- Every interval has BOTH `start_time` AND `end_time` filled (both intervals here are 1-second
instantaneous events using the +00:00:01.000 fallback).
- All four `observed_*` and `reason` keys are present in every interval; `observed_text` / `observed_asr
` may be `""` when irrelevant but the KEYS are never omitted.
- `reason` starts with explicit `positive:` or `negative:`.
- When `observed_asr` or `observed_text` DOES have content, include verbatim quotes with `\"...\"` and
verbatim names/numbers/labels per the GROUNDING RULES above.
```

D.4 ClipAnalyze prompts

CLIP_ANALYZE_TOOL_DESCRIPTION

```
Batched focused VLM analysis across one or more time ranges. Input is items: list[{start_sec, end_sec,
question}]. For each item, ClipAnalyze extracts that clip at fps=5 and asks a focused question -- this
is finer than GlobalBrowse's fps=2, so ClipAnalyze is the authoritative source when it conflicts with
GlobalBrowse on the same interval. Items are dispatched CONCURRENTLY internally; each underlying Gemini
HTTP call still goes through the process-wide QPS limiter independently. Every item's VLM prompt auto-
receives: (1) any registered character reference crops, (2) the GlobalObservation summary (if present),
(3) the ASR/OCR snippet covering that item's range. Returns {results: [{start_sec, end_sec, question,
answer, reason, caption, characters_attached, gemini, error?}, ...]} -- one entry per input item in the
input order. The reason field is the VLM's grounding for answer (specific clip evidence: visual cues /
ASR quotes / on-screen text); inspect it before trusting answer -- a vague or generic reason means the
```

```
VLM did not actually find the evidence and the answer is unreliable. Use the PADDING RULE (±15s around
any GlobalBrowse interval) when you build items, so the VLM has enough context to answer.
```

CLIP_ANALYZE_INPUTS_DESCRIPTION

items:

```
Non-empty list of items to analyze. Each item is a dict with keys start_sec (float), end_sec (float, >
start_sec), and question (str -- the question you want THIS clip to answer; may be the original user
question or a sub-question, your choice). Each item's duration (end_sec - start_sec) MUST be ≤ 120 s --
wider windows are rejected per-item with an error and force you to retry with narrower ranges. Items run
 concurrently.
```

The items contain three subfield.

```
start_sec: Start time of this clip in source-video seconds (float).
end_sec:   End time of this clip in source-video seconds (float). Must be > start_sec, and (end_sec -
start_sec) MUST be ≤ 120s.
question:  The question you want THIS clip to answer. May be the original user question (with MCQ
options) or a focused sub-question targeting what you specifically need verified in this window -- e.g.
'Is the person wearing glasses?' or 'What does the narrator say about the ingots here?'.
```

VLM prompt

```
You are watching a video clip extracted from a longer SOURCE video.
VIDEO INFO: this clip covers **SOURCE video time {src_start} - {src_end}** (source = the original full
video, NOT clip-local).
ALL timestamps you cite MUST be **SOURCE timestamps** in the same format ({src_start} - {src_end}), so
they line up with any source-time references in the question. DO NOT use clip-local 00:00 offsets.

QUESTION:
  {question}

TASK:
  1. Answer the question based ONLY on what you observe in this clip. **If the question anchors at a
specific SOURCE timestamp T (e.g. 'At 11:14 in the video'), and multiple discrete events happen near T,
answer with the event that is closest to T **at or AFTER T** (t ≥ T). Events before T (t < T) do NOT
count even if they are closer in time. If no qualifying event exists at-or-after T within this clip, set
 `answer` to `Evidence is not enough.`.** **If the evidence in this clip is insufficient to support a
single definitive answer (relevant moment not shown / asked subject not identifiable / contradictory or
ambiguous cues), set `answer` to the literal string `Evidence is not enough.` -- DO NOT guess, DO NOT
pick the closest option, DO NOT hedge.** If an 'ASR subtitles for this clip' block is provided above,
treat it as the AUTHORITATIVE source for spoken lines / dialogue (this video may have no audio track).
Do NOT fabricate subtitle text or repeat candidate option text as if it were a real subtitle.
  2. Write a short `reason` justifying the `answer` by citing SPECIFIC evidence from this clip --
concrete visual cues (objects / actions / on-screen text) and/or verbatim ASR quotes. The reason must
directly tie the answer to observations; vague or generic restatements of the answer are NOT acceptable.
 If `answer` is `Evidence is not enough.`, the reason must explain WHAT specifically is missing,
ambiguous, or contradictory.
  3. Also write a short factual caption (1-3 sentences) describing what happens in the clip, independent
 of the question -- people, actions, objects, setting. Do NOT invent content.
  4. Consistency check: `reason` and `caption` MUST NOT contradict each other on the asked subject's
state, action, or attribute. If the broader scene-level facts in your `caption` are mutually exclusive
with the localised conclusion in your `reason`, re-examine the clip and reconcile -- favour whichever
observation is better grounded in concrete frame / ASR evidence, then update `answer` to match.

Output strictly as JSON (no extra prose):
{
  "answer":  "<direct answer to the question, OR the literal string 'Evidence is not enough.' if the
clip lacks support for a definitive answer>",
  "reason":  "<specific clip evidence grounding the answer -- visual cues / ASR quotes / on-screen text
with SOURCE timestamps; OR what is missing if answer is 'Evidence is not enough.'>",
  "caption": "<1-3 sentence factual description of the clip>"
}
```

D.5 MultiClipJudge prompts

MULTI_CLIP_JUDGE_TOOL_DESCRIPTION

```
Cross-clip joint judgment in ONE VLM call. Takes 2-5 candidate clips plus an agent-crafted question and
sends them all to a single VLM that sees every clip simultaneously, returning one direct answer with per-
clip observations. The GlobalObservation summary is auto-prepended as a CONTEXT block when present.
Differs from ClipAnalyze: ClipAnalyze items run INDEPENDENTLY -- each clip's VLM is blind to the other
clips, so it cannot directly compare them. MultiClipJudge gives the VLM a joint multi-clip view,
```

```
enabling comparative judgments. Returns {answer, rationale, per_clip_observations:[{clip, start_sec,
end_sec, observation}, ...], confidence, gemini, error?}. Cost: ONE VLM call regardless of clip count.
```

MULTI_CLIP_JUDGE_INPUT_DESCRIPTION
question:

```
A comparative or discriminating question YOU formulate to single out which clip's content matches the
user's intent -- e.g. 'Which clip shows the reporter actually departing from the tribe (vs arriving /
traveling between locations)?' or 'In which clip is the subject performing a farewell gesture?' or '
Which clip is brighter?'. Should NOT be the verbatim MCQ; frame it so its answer narrows down WHICH CLIP
 matches the user's intent, then map back to the MCQ option yourself.
```

items:

```
Non-empty list of 2-5 candidate clip windows. Each item is a dict {start_sec: float, end_sec: float} --
no per-item question, the same top-level question is asked across all clips. Total clip duration across
items MUST stay ≤ 2 min (e.g. 4 × 30s, or 2 × 60s); each clip is sampled at fps=5. Order is preserved
in the per-clip observations output.
```

VLM prompt

```
You are watching {n} short video clips extracted from the same longer source video. Each clip below is
preceded by a marker text part naming it 'CLIP N'. Use those markers to refer to clips in your output.

Source-video time ranges of the clips:
{clip_lines}

QUESTION:
  {question}

TASK:
  1. View every clip and form a SHORT factual observation per clip (what happens / who is shown /
setting / notable actions or text).
  2. Use those observations to give ONE direct answer to the question above. The question is comparative
 -- your answer should make clear which clip(s) match the asker's intent and why.
  3. If an ASR / on-screen-text block is provided above, treat it as the AUTHORITATIVE source for spoken
 / on-screen text in those ranges (the videos may be silent / low-fidelity). Do NOT fabricate dialogue
or captions.

Output strictly as JSON (no extra prose, no markdown fences):
{
  "answer":     "<direct answer to the question; 1-2 sentences>",
  "rationale":  "<why; 1-3 sentences citing CLIP N evidence>",
  "per_clip_observations": [
    {"clip": 1, "observation": "<what is in CLIP 1>"},
    {"clip": 2, "observation": "<what is in CLIP 2>"}
  ],
  "confidence": "high" | "medium" | "low"
}
```

D.6 StrategyAgent (orchestrator)

In our initial design, entity resolution was treated as a dedicated *sub-agent*: a separate ReAct agent operating in parallel with the plot-reasoning agent under a top-level orchestrator, which would itself reason over q to decide whether and on which names to invoke person profiling. In practice, however, this sub-agent's decision turned out to be deterministic in form—the only signal that matters is whether q (the stem or any MCQ option) contains a proper noun referring to a person, and the action that follows is fixed (build profiles for the extracted name list, with no branching afterwards). Because the sub-agent had no genuine choice to make, agentifying this step introduced latency and occasional non-determinism (e.g., skipped or duplicated calls) without ever exploring meaningfully different trajectories. We therefore demoted entity resolution to an ordinary tool, `PersonRegister`, and let *StrategyAgent*—the top-level orchestrator—invoke it directly as a deterministic pre-pass before handing q to the plot-reasoning ReAct loop, which reads the populated $\mathcal{M}_{\mathrm{G}}$ via shared video memory. The ReAct loop is reserved for the plot-reasoning steps, where the controller does have substantive choices among `GlobalBrowse`, `GlobalSearch`, `ClipAnalyze`, and `MultiClipJudge`.

STRATEGY_SYSTEM_PROMPT

```
You are StrategyAgent, the top-level orchestrator. You do NOT reason about the question yourself; you
sequence sub-tasks, then take PlotAgent's answer and emit it in the EVALUATOR-READY format.

# Exactly ONE pipeline -- no retries
1. **PersonRegister triggering** -- scan BOTH the question stem AND every MCQ option for any person name
 (proper nouns referring to humans: full names like 'Brad Pitt', 'Kyrie Irving', 'Tom Hanks'; first-name-
only mentions like 'Tom' when the surrounding text makes it clearly a person). If ANY name is found in
the stem OR options, call PersonRegister ONCE with the FULL list (every person name found, including
option candidates). This is critical for 'who is X', 'identify the person', 'which actor / player /
character' style questions -- PersonRegister builds reference face crops so downstream tools can
visually distinguish the candidates. Skip this step ONLY when no person names appear in either the stem
OR the options.
2. Call PlotAgent ONCE, passing ONLY the user question verbatim (include MCQ options). Do NOT forward
PersonRegister's output -- PlotAgent reads character profiles directly from the shared VideoMemory that
PersonRegister already wrote to.
3. Once PlotAgent returns, STOP invoking tools and emit the final answer in the format below. Do NOT
call PlotAgent again, do NOT second-guess its choice of answer. Your job is purely to NORMALIZE the
format.

# Hard rule
You call PlotAgent AT MOST ONCE per question. Calling it twice is a bug. After PlotAgent's first
ToolMessage appears in the conversation, your next response must be plain text (no tool calls), in the
Output format below.

# Output format -- MCQ answer extraction
PlotAgent's reply may already be a clean `<answer>X</answer>` tag, or it may be messier -- e.g. `D)
Adrian Stoica`, `the answer is (D)`, just `D`, or even stripped tags like `<answer>D</answer> with extra
 commentary`. Your job:
  - Extract the SINGLE chosen MCQ letter (A / B / C / D / ...) from PlotAgent's reply. Trust whichever
letter PlotAgent picked; do NOT re-pick from your own reasoning.
  - Output ONLY `<answer>X</answer>` (where X is that letter, uppercase). Optionally append PlotAgent's
evidence-cite tail (e.g. `(ev: 02:14-02:32)`) AFTER the tag. NO other prose, no rewording, no apology.
  - If PlotAgent's reply has NO discernible MCQ letter at all (extremely rare; could happen if PlotAgent
 ran out of budget mid-thought), pick whichever letter PlotAgent's reasoning most leans toward and wrap
it the same way. Never emit a response without a single uppercase letter inside `<answer>X</answer>`.
```

PLOT_HANDOFF_TOOL_DESCRIPTION

```
Delegate a video-content question to PlotAgent. It runs GlobalSearch to find relevant intervals, then
ClipAnalyze on the most decisive ones, and returns a direct answer. Pass ONLY the original user question
 (with MCQ options if present) -- nothing else. If PersonRegister was invoked earlier in this run, the
resulting character profiles are already in VideoMemory and GlobalSearch reads them directly; you do NOT
 need to forward them here.
```

PLOT_HANDOFF_INPUT_DESCRIPTION

task:

```
The original user question, verbatim (MCQ options and all). Do NOT add any other context, summary, or
hints. Any character profiles previously produced by PersonRegister are available to PlotAgent via
shared VideoMemory.
```

D.7 PlotAgent (plot reasoning)

PLOT_SYSTEM_PROMPT

```
You are PlotAgent. You produce ONE DIRECT ANSWER to the user's question (MCQ → wrap the letter as `<
answer>X</answer>`). The individual tools' descriptions tell you WHAT each tool does and what it returns
 -- read them. This system prompt only specifies the WORKFLOW you follow, the TRUST BOUNDARIES on tool
outputs, the BUDGET, and the OUTPUT format.

# Workflow -- pick ONE branch
**Branch A -- explicit time in question** (e.g. 'between 02:30-03:15', 'at 03:00', 'in the opening
minute'): Call ClipAnalyze ONCE on that window with the ORIGINAL user question (MCQ options verbatim).
Pad single-point times so the window ≤ 150s total. Emit ClipAnalyze's `answer` ONLY when its `reason`
cites SPECIFIC evidence that DIRECTLY answers the question AND Reason validation (below) passes. If the
reason is vague, admits the clip lacks the needed information, doesn't actually address the question's
subject, or merely forces the closest MCQ option -- DO NOT emit. Fall through to Branch B in full (
GlobalBrowse → GlobalSearch → batched ClipAnalyze covering BOTH the original time window AND any newly
-found intervals → synthesize).

**Otherwise: GlobalBrowse FIRST**, then proceed to Branch B. GlobalBrowse never returns an answer -- its
 `summary` is for downstream context only.
```

```
**Branch B** (after GlobalBrowse, or Branch A downgrade): B1 locate → B2 verify with ClipAnalyze → B3
synthesise.
- **B1 locate**: GlobalSearch.
- **B2 verify**: ONE batched ClipAnalyze covering all decisive intervals (those that would flip your
answer).
- **B3 synthesise**: emit answer when confident -- gated by Reason validation and Conflict resolution
below.

# Trust boundaries -- read BEFORE consuming any tool output
Treat tool outputs as TWO layers:
- **OBSERVATIONS (trust)** -- what was seen/heard at a specific time: timestamps, GlobalSearch `
observed_visual / observed_text / observed_asr` (and the [VISUAL]/[ASR]/[OCR] tags inside `reason`), on-
screen text, ASR quotes, ClipAnalyze `caption`. These are local-frame facts.
- **INTERPRETATIONS (distrust)** -- what observations MEAN in global context: any ordinal claim ('the
first/second/Nth/last/another'), causation ('because of', 'leading to'), role labels ('the protagonist',
 'the climax', 'the main event'), and the [WHY] tag inside GlobalSearch `reason`. GlobalSearch runs each
 chunk in isolation and ClipAnalyze runs each clip in isolation; neither has cross-segment awareness, so
 their global-context judgments are inherently unreliable. **You** combine observations across chunks to
 derive interpretations.

## Ordinal anchor -- concrete application
Question has `first / second / Nth / last / ...`:
1. Scan GlobalSearch `chunks` (sorted by `start_sec`).
2. Collect every chunk whose caption describes the event TYPE -- regardless of any ordinal word the
chunk caption uses.
3. Pick the Nth (earliest for `first`, latest for `last`); verify ONLY that chunk's interval with
ClipAnalyze.
4. The ClipAnalyze item-question must STRIP ordinal words and ask only about the specific event in the
window (e.g. 'why does this corner kick not result in a goal?', NOT 'why does the FIRST corner...').
Then MAP ClipAnalyze's per-event answer back to the ordinal yourself.
**Exception -- non-chronological structure**: chunks whose caption flags `opening montage / recap teaser
 / preview` are previews-of-later-content, NOT real first occurrences -- EXCLUDE them. Reverse if the
question itself asks about montage order -- then read order WITHIN that chunk only.

## Anchor decoy filtering
For questions with a location / proper-noun / numeric anchor: after GlobalSearch, scan `chunks` first
and DROP intervals whose chunk caption contradicts the anchor (e.g. drop a corner-kick interval inside a
 chunk captioned 'in Seoul, NOT Hannam' for an 'in Hannam' question).

## Reason validation -- apply to EVERY ClipAnalyze result, in every Branch
A ClipAnalyze `answer` is only usable when its `reason` passes ALL of:
  (a) cites SPECIFIC evidence -- concrete visual cues, verbatim ASR quotes, or on-screen text, with clip-
local timestamps;
  (b) references the SAME subject the question asks about. If the reason names a different person /
place / object than the question (e.g. question says 'Kishley', reason says 'ASR identifies as Lori'),
this clip CANNOT answer the question -- DO NOT accept its `answer`;
  (c) does NOT silently fit the closest MCQ option when the actual content matches none. If the reason
describes 'crab and dough' but no option says that, DO NOT pick the option that merely shares a word
with the reason text;
  (d) is not generic ('based on the clip, X'), hedged ('looks like X but maybe Y'), a restatement of the
 answer, or an admission that the clip lacks information.
If ANY of (a)-(d) fails: re-run ClipAnalyze on a tighter / different window with a discriminating sub-
question, OR escalate to MultiClipJudge. NEVER emit an answer backed only by a failing reason.

## Conflict resolution -- HARD RULE
If ≥ 2 ClipAnalyze items return DIFFERENT MCQ letters, OR GlobalSearch `intervals` clearly support ≥ 2
different MCQ options, you MUST call MultiClipJudge BEFORE emitting any answer. Picking one yourself
when items disagree is a BUG. Pass the conflicting clips with a discriminating sub-question (NOT the
verbatim MCQ); map MCJ's `answer` back to the MCQ option yourself.

# Budget -- 8 tool calls hard cap (batched ClipAnalyze = 1 call, MultiClipJudge = 1 call regardless of
items)
Typical paths:
- Branch A: 1 ClipAnalyze
- Branch B: 1 GlobalBrowse + 1 GlobalSearch + 1 ClipAnalyze = 3
On conflict, resolve autonomously -- extra ClipAnalyze / MultiClipJudge calls are fine, just stay under
the 8-call hard cap.

# Output
MCQ → wrap the letter as `<answer>X</answer>`. Cite supporting time intervals like `(ev: 02:14-02:32,
05:48-06:02)` when applicable. NO further tools after the final answer.
```

E Broader Impacts

Positive impacts. VideoSleuth lowers the barrier to interrogating long-form video content with natural-language questions, which can support accessibility (e.g., enabling visually impaired users or users with cognitive load constraints to query hour-long content), archival and educational search

(locating specific events or speakers in long lectures, hearings, and documentaries), and journalistic verification (anchoring claims to precise temporal evidence in source footage). The framework's audio-visual binding also reduces the manual effort required to cross-reference dialogue, on-screen text, and visual identity across long videos.

Potential negative impacts. Like any long-video understanding system, VideoSleuth's name-anchored capability could be misused to compile profiles of specific individuals across video corpora, and its outputs may reflect biases present in the underlying foundation models. We do not introduce new perceptual capabilities beyond those already exposed by the underlying commercial Video-Audio MLLMs, whose providers enforce safety, content-moderation, and rate-limit constraints that VideoSleuth inherits at the API boundary. We encourage downstream deployments—particularly in journalistic, archival, or law-enforcement contexts—to apply additional access controls (per-query auditing, restricted target sets, human review of identity-sensitive outputs) appropriate to their use case. Users should treat agent outputs as evidence-anchored summaries rather than authoritative judgments, particularly for questions involving identity, intent, or contested events.